Méthodes d'apprentissage profond pour l'estimation de paramètres météorologiques à partir d'images webcam. Applications au suivi des épisodes de neige en plaine

Pierre Lepetit

▶ **To cite this version:**

Pierre Lepetit. Méthodes d'apprentissage profond pour l'estimation de paramètres météorologiques à partir d'images webcam. Applications au suivi des épisodes de neige en plaine. Océan, Atmosphère. Université Paris-Saclay, 2021. Français. NNT : 2021UPASJ023 . tel-03625313

HAL Id: tel-03625313
https://theses.hal.science/tel-03625313

Submitted on 30 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

université
PARIS-SACLAY

Méthodes d'apprentissage profond pour l'estimation de paramètres météorologiques à partir d'images webcam. Applications au suivi des épisodes de neige en plaine.

Deep learning methods for the estimation of meteorological parameters from webcam images. Application for the characterization of snow events in lowlands.

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 579 : sciences mécaniques et énergétiques, matériaux et géosciences (SMEMaG)
Spécialité de doctorat : Météorologie
Graduate School : Géosciences, climat, environnement et planètes
Référent : Université de Versailles-Saint-Quentin-en-Yvelines

Thèse préparée dans l'unité de recherche LATMOS (Université Paris-Saclay, UVSQ, CNRS, LATMOS) sous la direction de Laurent BARTHES, enseignant-chercheur, la co-direction de Cécile Mallet, professeur des universités, et le co-encadrement de Lucie ROTTNER, ingénieur des travaux de la météorologie.

Thèse soutenue à Paris-Saclay, le 15 décembre 2021, par

Pierre LEPETIT

Composition du jury

Valérie CIARLETTI Professeur des universités, UVSQ	Présidente
Patrick GALLINARI Professeur des universités, Sorbonne Université	Rapporteur & Examinateur
Nicolas HAUTIÈRE IPEF - Ingénieur des Ponts, des Eaux et des Forêts, Université Gustave Eiffel	Rapporteur & Examinateur
Mounim A. EL YACOUBI Professeur des universités, Télécom SudParis	Examinateur
Laure RAYNAUD Ingénieur des Travaux, CNRM	Examinatrice
Laurent BARTHES Enseignant-chercheur, UVSQ	Directeur de thèse
Cécile MALLET Professeur des universités, UVSQ	Co-Directrice de thèse

THESE DE DOCTORAT
NNT : 2021UPASJ023

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMAG)

Titre :Méthodes d'apprentissage profond pour l'estimation de paramètres météorologiques à partir d'images webcam. Applications au suivi des épisodes de neige en plaine.

Mots clés : images webcam, apprentissage profond, apprentissage semi-supervisé, learning-to-rank, observation de la neige, visibilité

Résumé : La surveillance des conditions météorologiques répond à des enjeux importants en matière de sécurité civile. Aujourd'hui, le réseau de capteurs spécifiques est insuffisant pour bien observer des phénomènes à enjeu difficiles à prévoir et souvent très localisés comme la neige en plaine et le brouillard. D'un autre côté, les webcams, très nombreuses sur le territoire, enregistrent ces phénomènes. Leurs images contiennent une information complémentaire valorisable et qualifiée d' « opportuniste ». L'extraction automatique de ces informations opportunistes présente donc un fort intérêt. Mais tirer de l'information pertinente à partir de scènes et de caméras très variées représente un défi au plan méthodologique. L'objectif principal de cette thèse était d'explorer des méthodes d'extraction par apprentissage qui soient applicables au suivi d'épisodes de neige en plaine. Trois paramètres ont été ciblés en particulier : la surface couverte par la neige, la profondeur de la couche de neige et la visibilité, qui varie avec l'intensité des précipitations. La construction de jeux de données, éléments clefs des approches par apprentissage, a constitué une part importante du travail. Nous avons commencé par rassembler des séquences d'images webcam très diverses qui contiennent des épisodes de neige. Faute de pouvoir associer ces séquences avec des mesures météorologiques fiables, nous avons développé une méthode d'annotation semi-automatique pour construire nos jeux d'apprentissage. La méthode comporte plusieurs étapes. La première étape, manuelle, a permis de définir des problèmes de classification du temps sensible. La seconde étape repose sur un algorithme de tri fusion adapté à la comparaison d'images par paires. Avec cet algorithme, un grand nombre de paires d'images ont été comparées par rapport à chacun des trois paramètres d'intérêt. Ces paires comparées nous ont permis de définir des problèmes d'estimation relative.

Nous avons évalué plusieurs méthodes de classification et d'estimation relative. Les réseaux de neurones profonds se sont montrés plus efficaces que les autres méthodes testées. Nous avons par la suite cherché à améliorer nos scores sur le problème d'estimation relative de façon à caractériser les phénomènes d'intérêt avec le plus de précision possible. Cela nous a conduit a développer des stratégies d'apprentissage spécifiques à des fonctions de rang implémentées sur des réseaux de neurones profonds. Nous avons d'abord montré qu'il était possible d'exploiter les images non-annotées pour améliorer les performances des fonctions de rang. Ces dernières ont aussi été adaptées à la prise en compte des paires d'images jugées incomparables pendant la deuxième étape de l'annotation. Nous introduisons des fonctions de rang "bivaluées" qui associent un intervalle à chaque image. Elles sont entraînées à restituer la relation d'incomparabilité à travers le chevauchement des intervalles prédits. Nous abordons ensuite le problème de l'étalonnage des fonctions de rang de façon à produire un intervalle de valeurs de visibilité plausibles pour une image donnée.

Un travail analogue a été réalisé sur les paramètres relatifs à la neige. Pour évaluer l'intérêt de ces méthodes au plan opérationnel, elles ont été portées sur une cinquantaine de webcams françaises au cours des mois d'hiver 2020-2021.

Title : Deep learning methods for the estimation of meteorological parameters from webcam images. Application for the characterization of snow events in lowlands.

Keywords : webcam images, deep learning, semi-supervised learning, learning to rank, observation of snow, visibility

Abstract : Civil security services highly depend on weather monitoring. Nowadays, high-stack events as settling snow in lowlands still partially escape to the dedicated sensors of our observation networks. On the other hand, these phenomena may be captured by simple webcams, which are ubiquitous. Webcam images may actually contain valuable meteorological data.
The automatic extraction of such opportunistic data would be of great interest. However, the extraction of fine-grained information about the weather conditions is still a challenging task. The main goal of this work was to explore deep learning approaches in order to characterize snow events in lowlands. In particular, three parameters are targeted : the extent of the snow layer, its depth and the horizontal visibility, which varies with the snowfall rate.

A first part of the work was devoted to the building of suitable data sets, which are keystones of the deep learning approaches. More than 5,000 webcam sequences around snow events were gathered. As it was not possible to associate these images with reliable measurements, a subset of ca. 500 sequences were labeled through an original semi-automatic approach. The first labeling step consisted in a handcrafted classification. The second step involves a specific sorting algorithm wich was developed on purpose. This algorithm allowed to compare a large number of paired images with respect to each of the three parameters of interest.
On these data sets, we compared several methods of classification and relative estimation. Off-the-shelf deep neural networks performed better than the other tested methods. We sought to improve performance on the relative estimation problem in order to characterize the weather with the finest possible resolution. It led us to develop specific deep learning strategies for ranking functions.
We first showed that unlabeled webcam images could be used to improve generalization performance of ranking functions. The latter were also adapted to take into account paired images that were considered as incomparable during the second labeling step.
Regarding the visibility, ranking functions were calibrated in order to provide image-wise intervals of possible values. A similar work was conducted on to characterize the snow settling. An observation campaign was realized over fifty french webcams during several months of the 2020-2021 winter to assess these approaches in an operational context.

Table des matières

1 Introduction **31**

1.1 Un rapide historique . . . 33

1.1.1 La caméra comme capteur de substitution . . . 33

1.1.2 La classification du temps sensible . . . 35

1.2 Perspectives offertes par le machine learning . . . 38

1.2.1 Problèmes de la vision par ordinateur abordés par machine learning . . . 38

1.2.2 Le deep learning . . . 43

1.2.3 Stratégies d'apprentissage transversales . . . 52

1.2.4 Conclusion . . . 55

1.3 Cadre de la thèse et organisation du manuscrit . . . 56

1.3.1 Cadre de la thèse . . . 56

1.3.2 Organisation du manuscrit . . . 57

1.4 Utilisation des annexes . . . 59

2 Collection des images et annotation **61**

2.1 Collection des images . . . 63

2.1.1 Exploitation de la source AMOS . . . 63

2.1.2 Les webcams des DIRs et du réseau Infoclimat . . . 67

2.1.3 Les archives du réseau TENEBRE . . . 68

2.2 Annotation . . . 70

2.3 Annotation des images . . . 70

2.3.1 Première étape d'annotation . . . 71

2.3.2 Deuxième étape d'annotation . . . 74

2.3.3 Troisième étape d'annotation . . . 80

2.3.4 Annotation des images TENEBRE. Correspondance avec l'annotation à la main. . . . 81

3 Résultats de base : classification et apprentissage par paires. Cas de la visibilité. **85**

3.1 Calssification de l'état du sol et de l'atmosphère . . . 86

3.1.1 Définition des problèmes de classification . . . 86

3.1.2 Méthodes d'apprentissage . . . 86

3.1.3 Résultats . . . 87
3.1.4 Conclusion et épilogue . . . 92
3.2 Méthodes d'apprentissage par paires . . . 93
3.2.1 Définition du problème . . . 93
3.2.2 Paramétrisation de la descente de gradient stochastique. Augmentation de données 95
3.2.3 Prédiction par paire . . . 97
3.2.4 Prédiction par image (fonction de rang) . . . 98
3.3 Apprentissage par paires sur AMOSvv . . . 100
3.3.1 Intérêt des paires non consécutives pour la prédiction par paire . . . 100
3.3.2 Apprendre la relation d'incomparabilité ? . . . 101
3.3.3 Comparaison entre prédiction par paire et prédiction par image . . . 104
3.3.4 Comparaison avec d'autres méthodes . . . 106
3.3.5 Une première application . . . 107
3.4 Conclusion . . . 109

4 Amélioration et étalonnage des fonctions de rang 111
4.1 Une méthode semi-supervisée . . . 113
4.1.1 Construction du jeu AMOSvvExt . . . 113
4.1.2 Discussion sur l'approche semi-supervisée . . . 115
4.2 Prédiction d'un ordre d'intervalle . . . 117
4.2.1 Notion d'ordre d'intervalle et limites d'utilisation. . . . 117
4.2.2 Fonctions de rang bivaluées . . . 119
4.2.3 Fonction de coût . . . 119
4.2.4 Utilisation des images équivalentes et moyennage . . . 120
4.2.5 Entraînement des modèles . . . 122
4.2.6 Evaluation des fonctions de rang bivaluées . . . 122
4.2.7 Cas d'erreur : . . . 127
4.2.8 Discussion sur les fonctions d'ordre bivaluées . . . 128
4.2.9 Conclusion sur les fonctions de rang bivaluées . . . 128
4.3 Etalonnage des fonctions de rang . . . 130
4.3.1 Etalonnage d'une fonction de rang bivaluée . . . 130
4.4 Intercalibration par apprentissage . . . 143
4.4.1 Méthode . . . 143

4.4.2 Conclusion sur l'approche par intercalibration 150
4.5 Conclusion et perspectives 150

5 Application à la caractérisation de la neige en plaine 153
5.1 Jeux d'apprentissage et problèmes d'apprentissage associés 155
5.2 Résultats de base 158
5.2.1 Procédures d'apprentissage et nomenclature 158
5.3 Une approche semi-supervisée pour l'étendue du manteau neigeux 164
5.3.1 Construction d'AMOSssExt 164
5.3.2 Résultats 165
5.3.3 Discussion 167
5.4 Classification neige/non neige sur les séquences TENEBRE 168
5.4.1 Modèles et données utilisées 168
5.5 Résultats et discussion 169
5.6 Suivi d'épisodes de neige en plaine pendant l'hiver 2020-2021 171
5.6.1 Données et méthode 171
5.6.2 Résultats et discussion 172
5.6.3 Discussion 173
5.7 Conclusion et perspectives 175

6 Conclusion 179
6.1 Rappel des principales étapes 179
6.2 Bilan global sur les objectifs 182
6.3 Limites et perspectives 184

A Compléments sur l'annotation 197
A.1 Critères d'annotation 197
A.1.1 Labels relatifs à une image 197
A.1.2 Annotation relative à une paire d'images (consécutives et non consécutives) 203
A.2 Annotation par sous-séquences (détails de la troisième étape d'annotation) 211
A.2.1 Préparation des sous-séquences 211

B Description détaillée des principaux jeux d'apprentissage 215
B.1 Description des images annotées à la main. 215

B.2 Description des séquences d'AMOSssExt. 219
B.3 Description du jeu TENEBRE. 219

C Nomenclature des modèles **223**
C.1 Nomenclature 223
C.2 Modèles mis à disposition 224

D Compléments sur les chapitres 3 et 4 **227**
D.1 Choix des modèles 227
D.2 Construction d'AMOSvvExt 229
D.2.1 Extraction des séquences 229
D.2.2 Tri automatique des sous-séquences 230
D.3 Approches multi-tâches 233
D.3.1 Apprentissage des préférences multi-paramètre 233
D.3.2 Apprentissage des préférences + classification 234
D.3.3 Conclusion 234
D.4 Prédictions d'un ordre d'intervalle sur un jeu de synthèse 235
D.4.1 Définition du problème 235
D.4.2 L'approche indirecte : 236
D.4.3 L'approche directe 238
D.4.4 Variantes 238
D.5 Compléments sur les résultats des fonctions de rang bivaluées 241
D.5.1 Prudence sur les scènes difficiles 241
D.5.2 Prédiction des inclusions 242
D.5.3 Démonstration de la propriété 2 244
D.5.4 Visualisation des résultats après étalonnage 245
D.6 Jeux de données pour l'intercalibration 251

E Compléments sur le chapitre 5 **253**
E.1 Compléments sur AMOSssExt 253

Table des figures

1.1 le problème de la classification binaire « ensoleillé »/« nuageux » sur des images singulières est difficile : l'humaine peut tirer parti de plusieurs caractéristiques de l'image, comme la présence ou l'absence d'ombres portées. 35

1.2 Deux exemples de réseaux de neurones profonds à couches de convolutions : VGG16 [56] et ResNet18 [57]. 46

1.3 Exemple de réseau complètement convolutif (full convolutional network). Ce U-net est utilisé sur des problèmes de segmentation sémantique, et de régression par pixel et de débruitage. 47

1.4 a.noyaux de convolution (premier canal) de la première couche d'un ResNet50 pré-entraîné sur Imagenet. Les couleurs rouges (resp. bleues) correspondent aux poids positifs (resp. négatifs). b. image placée en entrée du même réseau. c.sortie d'un des neurones de la couche L_{10} avant application de la fonction ReLU. Les pixels rouges correspondent à des valeurs positives. 48

1.5 Différentes stratégies d'apprentissages applicables dans le cas où des données supplémentaires sont disponibles (transfer learning), ou lorsque les données à disposition sont défectueuses (supervision faible). Les items surlignés en orange couvrent des techniques testées pendant la thèse. 53

2.1 Schéma synthétique des étapes d'extraction et de sélection à partir des quatre sources d'images. En bout de chaîne, en bleu, le premier nombre indique le nombre de séquences et le nombre d'images obtenus. 64

2.2 Exemples de scènes routières tirées de la base AMOS pendant des épisodes de chutes de neige. Les scènes et les caméras (modèle, orientation) sont très variées. 65

2.3 a-b. Images ne comprenant aucune information météorologique, rejetées pendant l'élagage manuel. c-d. Pendant les épisodes de précipitations (neige ou pluie), les images de mauvaises qualité sont fréquentes. Ici, la neige collée à la vitre de protection masquent la moitié de l'image. Mais ces images contiennent encore une information relative à la météorologie locale. Elles sont donc conservées. 67

2.4 Les dix scènes du jeu TENEBRE_1213. Seules les deux premières scènes peuvent être considérées comme des scènes routière. La dernière scène (Markstein) a généralement été exclue des analyses du fait de nombreuses mesures erronées. 69

2.5 Interface de LabelGO, logiciel d'annotation développé début 2019 pour l'annotation d'images. Après annotation complète, l'image est remplacée par la suivante dans l'ordre chronologique. Les attributs et les labels sont détaillés à l'annexe A.1. 72

2.6 Annotation de l'attribut « état du sol ». Cinq classes sont distingués : a. Par beau temps, la route est sèche (**dry_road**). b. Sous la pluie, ou au début des chutes de neige, la route s'assombrit et des reflets apparaissent (**wet_road**). c. La neige tient sur le sol, mais pas sur la route (**snow_ground**). La neige tient sur la route, mais ne la couvre pas entièrement (**snow_road**), la neige couvre entièrement la route (**white_road**). 72

2.7 Le contraste entre les parties claire et sombre de la chaussée sont dus à l'humidité. b. la route est exposée au soleil après une averse de neige. Les reflets du soleil peuvent donner l'impression que la route est couverte de neige. c. un cas de gelée blanche. d. sel fraîchement déposé sur la voie. 73

2.8 les images consécutives a – f ont été parcourues dans l'ordre chronologique. Le contraste apparu entre l'arbre au premier plan et la végétation au second plan, dans le rectangle blanc, conduit à décider que la visibilité est plus grande d'après l'image b que d'après a (visi : lower). 74

2.9 définition des niveaux utilisés pour la règle 2 à partir des labels qualitatifs. 76

2.10 Les relations obtenues pendant la première étape (paires consécutives) sont représentées en rouge. La règle 2 permet de tirer des relations des labels relatifs à l'image (en vert). 77

2.11 Graphes associés aux relations de la figure 2.10. Comme il n'y a pas d'équivalences dans cette séquence, $\mathcal{E}^v$ n'est pas représenté. . 77

2.12 Réduction du nombre de chaînes. Les images a, b, c, d, e, f sont celles des figure 4 et 5. A chaque étape, on compare les sommets de deux chaînes prises au hasard parmi les n chaînes de l'entrée. L'annotateur est sollicité (1,3,4) dès que la comparaison n'est pas contenue dans un deux graphes. Dès qu'une comparaison stricte est trouvée, le plus grand élément est mis de côté (croix noires). Lorsque toutes les images d'une des chaîne ont été mises de côté, un réarrangement en $n-1$ chaînes peut être effectué. 78

2.13 Relation d'ordre partiel entre douze images (noeuds du graphe) triées suivant le critère de la visibilité. Les nombres associés aux images sont composés de trois numéros : le premier indique l'indice de la chaîne dans la décomposition, le second indique la situe dans la chaîne, le dernier la situe dans la classe d'équivalence de l'image (seule une classe comporte deux images : 120,121). Les arêtes noires figurent la décomposition en chaînes obtenue par *poset-mergesort*. 79

3.1 Evolution de la fonction de coût (entropie croisée - équation 1.1) pendant l'apprentissage d'un ResNet50 et d'un ResNet18 sur le problème de la classification de l'état du sol (problème complet). . 89

3.2 Métriques pour l'évaluation d'un modèle qui prédit les trois classes de comparaison. 95

3.3 Prédiction par paires d'une relation d'ordre entre les deux images par un classifieur. 98

3.4 Prédiction par image avec une fonction d'ordre. Lors de l'entraînement, les réseaux siamois sont pénalisés lorsque les sorties sont rangées dans le mauvais ordre (ici, par une Hinge Loss). 98

3.5 a.b.c. Courbes d'apprentissage dans les configurations A,B,C définies dans la table (en haut). Dans les configurations A et B, le nombre d'arêtes est restreint. d. : Evolution des justesses (équation (5.1)) dans les trois configurations. 102

3.6 Comparaison entre deux classifieurs VGG11 entraînés à la prédiction par paire. Le premier est entraîné sur $\mathcal{G}_0^v$ à classer les paires d'images parmi $\{\dot{\succ}, \dot{\prec}\}$. Le second est entraîné sur $\mathcal{G}_0^v$ et $\mathcal{U}_0^v$ à classer parmi $\{\dot{\succ}, \dot{\prec}, \dot{\perp}\}$. En a., on a désactivé la prédiction d'incomparabilité pour comparer les justesses sur VAL_{same} et VAL_{indep}. En c. et d., on compare sur le jeu de test les classifieurs après épaississement (voir encadré 3.1). Les courbes sont obtenues en faisant varier le paramètre λ. 103

3.7 Comparaison entre les meilleurs classifieurs (prédiction par paire) et les meilleures fonctions de rang (prédiction par image). a.Courbes d'apprentissages lissées par moyenne mobile (5 époques consécutives). b-c. les fonctions de rang sont épaissies pour obtenir des relations d'incomparabilité (encadré 3.2) et comparées aux classifieurs épaissis. Les meilleures performances sont obtenues par symétrisation des classifieurs (equation (3.13)). 106

4.1 Comparaison entre des fonctions de rang implémentées sur des ResNet50 et apprises soit sur AMOSvv (méthode supervisée) soit sur AMOSvvExt (méthode semi-supervisée). a. Courbes d'apprentissage représentatives pour chacune des deux méthodes. b. Comparaison sur le jeu de test à partir des métriques définies en section 3.2.1.2. Pour chaque méthode, deux fonctions de rang ont été apprises sur les paires strictement ordonnées. L'apprentissage de la troisième fonction de rang fait intervenir les équivalences (voir l'annexe C). 115

4.2 a. L'ordre partiel défini sur l'ensemble des $\mathcal{I}_k$ est un ordre d'intervalles. L'incomparabilité correspond au chevauchement. b.Les éléments $x_1; x_2; x_3; x_4$ ne peuvent pas être représentés sous la forme d'intervalles. C'est un ensemble ordonné de type "2+2". 117

4.3 Entraînement d'une fonction de rang bivaluée. Les sorties (z_i^-, z_i^+) du modèle induisent un ordre d'intervalle. La relation d'ordre contenue dans les graphes (a.) peut être partiellement représentée sous la forme d'un ordre d'intervalle (b.). La position relative des bornes supérieures des intervalles associés à x_i, x_k (segments rouges) ne peut pas être précisée à cause d'un sous-ensemble de type 2+2 ($\{x_a; x_b; x_i; x_k\}$). Les positions relatives des bornes sont encodées dans les $\delta_{i,j}$ (c.). Dans la fonction de coût (d.), les bornes mal positionnées sont pénalisées par la RankNet Loss (e.). 119

4.4 Entraînement avec $\mathcal{L}^{du}$ sur AMOSvv ($\mathcal{G}_0^v$ et $\mathcal{U}_0^v$) et AMOSvvExt ($\mathcal{G}_1^v$ et $\mathcal{U}_1^v$). La justesse est calculée sur les trois classes. Les courbes d'apprentissage sont légèrement lissées (moyenne mobile sur trois époques consécutives). 123

4.5 Diagrammes Correctness-Completness et Prudence-Completeness sur le jeu de test d'AMOSvv. Les croix resprésentent les compromis Correctness-Completeness (ou Prudence-Completeness) atteints par les fonctions de rang bivaluées. Les courbes représentent les compromis obtenus par épaississement des fonctions de rang à valeur réelle (tiretés) et des fonctions de rang bivaluées (en traits pleins, voir encadré 4.9). a.Effet des fréquences de présentation et du moyennage. La fréquence de présentation des paires incomparables est de 2/3 pour le groupe 2 contre 1/3 pour le groupe 1. La courbe noire représente la moyenne des fonctions de rang bivaluées apprises sur AMOSvvExt. b-c. Comparaison entre fonctions d'ordre à valeurs réelles (groupe 0) et bivaluées (groupe 1). La légende vaut pour les trois graphiques b,c,d. d. Même comparaison qu'en c., mais en limitant le calcul des scores aux paires où au moins une des images est de mauvaise qualité. 125

4.6 a-e. Images du jeu de test d'AMOSvv associées à un intervalle de taille maximum (comparé aux images de même séquence) par le modèle vv_sl_due111.0. L'image a. est floutée par la condensation et l'image b., à cause de la focalisation sur des débris collés à la vitre. f. La dernière image est issue de la séquence entzheim3 (TENEBRE). 126

4.7 Cas d'une inclusion correctement prédite. Les cinq relations prédites sont correctes : entre l'image x_1 et l'image x_3, la visibilité a décru, mais un flocon masque complètement la scène sur l'image x_4. Pour prédire correctement ces relations, il faut que l'intervalle associé à x_4 inclue l'intervalle associé à x_2. Les images viennent de la séquence AMOS n°713 du jeu de test. La dernière image est complètement masquée par un flocon. 127

4.8 Comparaison des performances en généralisation sur le problème à deux classes. Les points verts représentent les centres d'intervalles de mean_bivalued_gr1gr2. Les autres points représentent les valeurs prédites par les fonctions d'ordre entraînées sur AMOSvv (lignes 2-4 de la table 4.2). Lorsqu'une colonne est vide, c'est que les images de la séquence sont toutes incomparables deux à deux. . 129

4.9 Etalonnage des prédiction sur quatre caméras de TENEBRE. Les fonctions utilisées pour l'étalonnage des fonctions bivaluées sont estimées à partir de diagrammes quantile-quantile. Les coordonnées des points sont les centiles des séries de prédictions (abscisses) et les centiles des séries de visibilités (ordonnées) mesurées sur site pendant l'hiver 2012-2013 (octobre 2012 - mars 2013), de jour, à un pas de temps de dix minutes. Les prédictions ont été obtenues avec vv_sl_mean_gr1_gr2. 133

4.10 Comparaison entre prédictions et mesures sur quatre scènes du réseau TENEBRE. La visibilité (en mètres) est en ordonnée, l'heure UTC en abscisse. Les graphiques couvrent des périodes de trois jours, entre 8 h UTC et 17 h UTC (alors que l'étalonnage ne fait intervenir que les images prises entre 9 h et 16 h) ; ils couvrent au moins une situation de brouillard (visibilité $<$ 1000 m) par scène. Les estimations ponctuelles (croix vertes) sont obtenues par l'équation (4.15), les bornes des intervalles par les équations (4.14 - 4.13). Les grands intervalles sont représentés par les couleurs orange et rouge (voir encadrés 4.2). Les croix bleues représentent les mesures instrumentales 136

4.11 a.Dispersion des visibilités mesurées à Entzheim et Nancy.b.CDF estimées sur la même période (oct 2012 - mars 2013), pour Enzheim et Nancy.c.Prédictions du 19/11 sur la scène entzheim3 après calage sur les quantiles de visibilité mesurées à Entzheim.d. Série des prédictions sur la scène nancy1, le même jour, après calage sur les quantiles mesurés à d'Entzheim. 139

4.12 Distribution des erreurs relatives commises sur la prédiction en utilisant une station distante. La distribution est représentée par la donnée de la médiane (courbes claires, en bas), du troisième quartile et du neuvième décile (courbes foncées, en haut). Ces trois paramètres sont donnés pour des visibilités de l'ordre de 300 m (courbes noires) à 3000 m (courbes bleues). Par exemple, l'erreur relative médiane est la plus faible (20 %) pour une visibilité de l'ordre de 3000 m et une station à moins de 50 km. 141

4.13 Variations de $y^*(v)$, solution du problème d'optimisation défini par l'équation (4.19) dans le cas où la fonction de coût est la MSE (courbe bleue) ou la MAE (courbe rouge). à partir des mesures de RADOMEvv1213. Les courbes grises de la figure correspondent aux distributions de la visibilité associées à dix séries prises au hasard. . 145

4.14 Variations de $y^{*,-}(v)$ (courbe rouge) et $y^{*,+}(v)$ (courbe bleue), solutions du problème d'optimisation définies par l'équation (4.23) défini à partir des 112 séries de mesures de RADOMEvv1213. . . . 146

4.15 Les courbes rouges sont relatives au modèle entraîné sur les paires intra-séquence (vv_sl_due111.0), les vertes sont associées au relatives au modèle entraîné par intercalibration (cal0103). Ligne 1 : distribution des écarts relatifs à la mesure dans le cas où le modèle est étalonné sur nancy2 - 0%. Ligne 2 : distributions des écarts relatifs entre les prédictions selon la modalité d'étalonnage (étalonnage local ou étalonnage sur nancy1). 149

5.1 Résultats des apprentissages sur les jeux relatifs au paramètre étendue du manteau. Les fonctions de rang bivaluées (prédiction par image) sont en orange et rouge, les classifieurs (prédiction par paire) en bleu. L'épaississement est obtenu selon les règles des encadrés 3.1, 3.2 et 4.1. 160

5.2 Résultats des apprentissages sur les jeux relatifs au paramètre épaisseur du manteau (AMOSsd.0). Les fonctions de rang bivaluées (prédiction par image) sont en orange et rouge. Les classifieurs (prédiction par paire) sont en rose et bleu. Le classifieur ss_pl_du11.1 a aussi été entraîné sur les paires supplémentaires issues de la troisième étape d'annotation. 160

5.3 Exemple d'une paire contradictoire sur une caméra du jeu AMOS. Sur l'image de gauche, l'étendue est plus grande et l'épaisseur moins grande que sur l'image de droite. 161

5.4 a. Le meilleur classifieur sur le paramètre épaisseur est testé sur le jeu relatif à l'étendue (courbe rouge). La courbe bleue correspond au meilleur classifieur sur le paramètre étendue. b. Les mêmes courbes sont tracées après restrictions aux paires contradictoires du jeu de test AMOSss.1. Sur ces paires, le score parfait pour helvetica ss_pl_du11.0 est de +1 tandis qu'il est de -1 pour le réseau sd_pl_du00.0, entraîné à ordonner suivant l'épaissseur. Pour ce graphe, les prédictions ont été symmétrisées (voir chapitre 3). c.Les mêmes modèles sont évalués sur le jeu de test d'AMOSsd.0. . . . 162

5.5 Performances des fonctions de rang semi-supervisées pour l'estimation relative de l'étendue. Les Pso, prudence, completeness et correctness ont été calculées sur les paires du jeu de test d'AMOSss.1. La courbe bleue marque les performances du classifieur utilisé pour construire le jeu AMOSssExt. Les courbes vertes représentent les fonctions de rang bivaluées apprises sur AMOSssExt. Le moyennage de ces cinq fonctions donne la courbe noire. Les courbes pointillées correspondent à des fonctions de rang à valeurs réelles apprises sur AMOSssExt tandis que les courbes oranges continues représentent les meilleures fonctions de rang bivaluées apprises sur AMOSss.0. . 165

5.6 Détail des performances. Les modèles comparés sont les mêmes que pour la figure 5.5. a.Diagramme Pso-Correctness après restriction aux paires contenant au moins une image sans neige au sol. b.Diagramme prudence-correctness après restriction aux paires comprenant au moins une image de mauvaise qualité. 166

5.7 Comparaison des classifieurs et des fonctions de rang bivaluées sur la classification « neige au sol/non neige au sol ». La vérité terrain est définie à partir des labels instrumentaux de la base TENEBRE_1218. Les 9 diagrammes Spécificités-Sensibilité correspondent aux caméras TENEBRE (figure 2.4) 170

5.8 a.Prédictions de mean_bivalued sur les séquences d'images du 16/01/2021. Les intervalles prédits ($[z_i^-, z_i^+]$) sont représentés en rouge foncé lorsque le seuil est dépassé ($z_i^- > z_{1000}^+$). La première image de la journée sur laquelle le seuil est dépassé donne le début « prédit » de l'événement. Les flèches vertes représentent l'heure UTC à laquelle la neige commence à apparaître sur le sol (début observé). Les heures des débuts observés (resp. prédits) sont affichées en vert (resp. en rouge). Le cas particulier de la webcam LFFD est décrit dans le texte. b.Localisation des webcams (sauf l'aérodrome EBAV, située en Belgique). 172

5.9 Enneigement (ligne a.) et augmentation progressive de la visibilité (ligne b.) par « traduction d'image ». avec un UNet. Les deux images prises (une pour chaque ligne) en entrée appartiennent au jeu de test. La ligne a. a été générée à partir d'une image sans neige. La ligne b., à partir d'une image prise par beau temps. . . . 177

A-1 Le contraste entre les parties claire et sombre de la chaussée sont dus à l'humidité. b. la route est exposée au soleil après une averse de neige. Les reflets du soleil peuvent donner l'impression que la route est couverte de neige. c. un cas de gelée blanche. d. sel fraîchement déposé sur la voie. Contrairement au routes enneigées, le sel s'étale depuis le centre. 198

A-2 Première ligne : flocons en vol, de jour. a. trainées pixelisées visibles sur toute l'image. b. les flocons ne sont visibles que par contraste, devant les sapins. c. quelques flocons au premier plan. Des traces de saleté sont aussi visibles sur la vitre de protection. Seconde ligne. De nuit. d. la source de lumière est proche de la caméra, la trainée laissée par les flocons est plus grande que sur l'image suivante (e.), où les flocons n'aparaissent que sous les réverbères. f. sur cette image, il ne s'agit probablement pas de flocons en vol, mais de neige soufflée par le vent. 199

A-3 Quatre masque physiques. Les flocons sur la lentille, et les gouttes sur la vitre de protection sont très fréquents pendant les épisodes de précipitation. Les stalactites, les araignées sont plus anecdotiques, mais susceptibles de biaiser les prédictions sur de longues séquences d'images. 200

A-4 a. saturation du capteur de jour, b. saturation de nuit, pendant un épisode de neige. c. image floue (condensation). d. une partie de l'image manque. 201

A-5 Les deux premières (AMOS 1550 et 1002) sont des exemples de scènes urbaines, les deux autres sont des exemples de scènes spéciales. La troisième scène est une vue en plongée sur une route de campagne enneigée. Les créneaux en bas permettent d'estimer la hauteur de neige. 203

A-6 Sur a, le sommet de la colline est masqué par un plafond nuageux plus bas : on ne peut pas appliquer la règle 1. Par contre, le contraste à l'horizon est plus faible que sur b (chutes de neige en cours). On décide donc $b \prec_v a$. 205

A-7 Règle des contrastes entre objets de même luminance. Un constraste apparaît dans le carré vert sur l'image b. On décide alors $a \prec_v b$. . 206

A-8 Halo amplifié par les mauvaises conditions météorologiques. 207

A-9 Un banc de brouillard réduit la portée optique sur l'image a. Elle est moins grande que sur l'image b prise lors d'un épisode de neige. Mais les critères 3-4 indiquent $b \prec_v a$. On décide donc l'incomparabilité ($a \perp_v b$). 207

A-10 De l'application des critères résultent des relations d'ordre pouvant être représentées sous forme de graphes. Ces représentations permettent de différencier les causes d'incomparabilité qui sont intrinsèque à l'image de causes d'incomparabilité qui sont dues à la différence d'aspect entre les deux images. Par exemple, à gauche, l'image x_0 est plus souvent trouvée incomparable parce que sa qualité est affectée par des parasites. A droite, les incomparabilités sont liées à des différence d'aspect qui compliquent la comparaison des contrastes (par exemple lorsque l'éclairage global est d'intensité très différente, ou lorsque les surfaces sont enneigées sur l'une des deux images et pas sur l'autre). 210

A-1 Apprentissage des préférences sur AMOSvv. Performances en validation de la prédiction par paire. Les modèles VGG7 et VGG8 contiennent respectivement 4 et 5 couches de convolution. Les courbes ont été lissées par moyenne mobile (largeur : 5 époques). . 228

A-2 Apprentissage des préférences sur MAMOSvv. Performances en validation de la prédiction par image. a. Courbes d'apprentissage (après lissage) sur AMOSvv. b.meilleures performances sur le jeu de validation. c-d. Séparation des différentes architectures sur le jeu de test, après apprentissage sur AMOSvvExt. 228

A-3 Construction d'un ordre d'intervalle sur les séquences d'AMOSvvExt (voir section D.2.2). Les arcs rouges correspondent aux comparaisons faites à la première passe, les bleus, à la seconde. Les incomparabilités sont représentées par des arcs en pointillé. Sur le graphique de droite, on ne représente que les relations d'incomparabilité (segments verticaux gris). 230

A-4 Exemple d'images du jeu de synthèse. En général, les disques contiennent assez d'information pour estimer les bornes d'un intervalle contenant la valeur cible. 235

A-5 Ordre d'intervalle associé aux images de la figure A-4. 236

A-6 Correspondance entre les bornes supérieures et inférieures prédites et observées. Les points rouges (resp. bleus) sont de coordonnées $(\mu_i + 2 \times \sigma_i, z_i{}^+)$ (resp. $(\mu_i - 2 \times \sigma_i, z_i{}^-)$) où les $z_i{}^\pm$ sont les sorties de la fonction d'ordre bivaluée. Le diagramme est effectué sur les 500 premières images de SYN2000. La prédictions sont obtenues avec la fonction d'ordre bivaluée SYN_slbiv_VGG11.0. Une erreur est commise lorsqu'entre un point rouge et un point bleu, l'ordre des ordonnées est contraire à celui des abscisses (par exemple, entre les deux points cerclés de noir. 239

A-7 Chaque point correspond à une séquence du jeu de test. Les points rouges (resp bleus), aux scènes à horizon bas (resp. à horizon haut). Les coordonnées correspondent aux largeurs moyennes des intervalles après rognage bas (abscisses) ou haut (ordonnées). 242

A-8 Comparaison entre prédictions et mesures sur quatre scènes du réseau TENEBRE. La visibilité (en mètres) est en ordonnée, l'heure UTC en abscisse. Les graphiques couvrent des périodes de trois jours, entre 8h et 17h ; ils couvrent au moins une situation de brouillard (visibilité $<$ 1000 m) par scène. Les estimations ponctuelles (croix vertes) sont obtenues par l'équation (4.15), les bornes des intervalles par les équations (4.14 - 4.13). Les grands intervalles sont représentés par les couleurs orange et rouge (voir encadrés 4.2). Les croix bleues représentent les mesures instrumentales 246

A-9 Comparaison entre prédiction et mesure pendant des épisodes de neige en plaine. Cas de chutes de neige intenses avec tenue de neige au sol. Les plus gros intervalles des figures b. et d. sont prédits pour les images les plus dégradées. 247

A-10 Comparaison entre prédictions et mesures pendant de faibles chutes de neige. Les prédictions sont correctes pour les scènes nancy2 et entzheim1. Les flocons (gouttelettes ?) défocalisés génèrent d'importantes erreurs de prudence en matinée sur les deux scènes neige_nancy et parc_entzheim, puis à partir de 14h sur neige_nancy. 248

A-11 a.Comparaison entre prédictions et mesures pendant un épisode de brouillard sur la scène dorans1. Il s'agit d'une scène atypique, sans premier plan, à horizon bas (a.2). Pour des basses visibilités, il n'y a plus aucune information sur la structure de la scène (a.1).b. La caméra est rarement entretenue, et des traces sont visibles sur l'image pendant les trois premiers mois. Sur la scène portail_entzheim. c.Cas d'erreur (surestimation). Le brouillard se lève au matin du 13/11/2012 sur Entzheim. A 9h30 (c.1) la mesure n'est pas représentative de la scène mais la portée optique est encore limitée à quelques kilomètres. d.Défaut de représentativé des mesures sur Roissy. Des averses se sont succédées pendant la journée du 10/10/2012. Les fluctuations de la mesure à partir de 12 h n'apparaissent pas sur la séquence d'images (comparer d.1 et d.2). 249

A-12 Différences $e^{q90}_{med.} - e^{q50}_{med.}$ pour différents modèles appris sur les paires incomparables. Les modèles ont tous été étalonnés avec les mesures colocalisées. 250

A-13 Cadres pour l'augmentation de données. 251

Liste des tableaux

1.1 Taille des Réseaux utilisés pendant la thèse. Pour chaque architecture, on indique la catégorie (première colonne), la version (colonne 2). Dans les réseaux à résidus, des étapes d'agrégation peuvent être réalisées par des couches de convolutions accessoires associées à des noyaux de dimensions spatiales réduites à 1×1. Dans les cases de la colonne 3, on indique le nombre de couches de convolution classiques, le nombre de convolutions accessoires, et le nombre de couches complètement connectées. Le nombre total de paramètres est indiqué en colonne 4. Dans la colonne 5, on indique les jeux de données sur lesquels des réseaux pré-entraînés étaient disponibles en 2018. 47

2.1 Description des sources d'images utilisées dans la thèse.(*) Le nombre exact de caméras est indicatif. Il n'a pu être évalué qu'au bout de la troisième étape d'annotation (voir section 2.1.1). Le nombre entre parenthèses, indiqué au dessous, correspond au nombre de sites internet dont les archives ont été exploitées. Le pas de temps est la durée médiane qui sépare deux images consécutives dans une séries d'images. Une version plus complète de cette table est disponible dans l'annexe B.1. 63

2.2 Description des sources d'images utilisées dans cette thèse. Cette table résume le descripitif de l'annexe A. 70

2.3 Statitistiques sur les sous-séquences triées avec *poset-mergesort*. La deuxième colonne indique le nombre de tris complets et le nombre de décompositions obtenues. Par exemple, pour AMOS, 300 sous-séquencse ont été décomposées et entre 15 et 40 des ces sous-séquences ont été complètement triées (cela dépend du paramètre). Dans les colonnes 3-5, les tailles (en nombre d'arêtes) des trois graphes sont indiquées pour chacun des trois paramètres : visibilité (v), étendue du manteau neigeux (s) et épaisseur du manteu (d). La taille moyenne des chaînes contenues dans les décompositions apparaît aux colonnes 9-11, et la taille de la plus longue chaîne contenue dans les graphes $\mathcal{G}^p$ apparaît, pour chaque paramètre p, dans les colonnes 12-14. Le ratio PA/P est défini dans le corps du texte. 80

3.1 Meilleurs scores (justesse équilibrée) obtenus sur des problèmes de classification définis à partir des labels qualitatifs. 87

3.2 Choix des hyperparamètres. Les hyperparamètres en gras ont été retenus après avoir testé différentes valeurs situées dans les intervalles entre parenthèses. Les autres termes en italique précisent la fonction pytorch utilisée. 88

3.3 Meilleur matrice de confusion sur le jeu de validation pour la classification de l'état du sol (problème « complet », voir table 3.1) . DrR : dry_road, WeR : wet_road, SG : snow_ground, SDR : snow_ground_dry_road, SR : snow_road, WhR : white_road. . . 90

3.4 Meilleure matrice de confusion sur le jeu de validation pour la classification de l'état de l'atmosphère (problème « complet », voir table 3.1). Le modèle est un ResNet50 entraîné simultanément sur quatre problèmes de classification (état du sol, état de l'atmosphère, présence de masque, présence de traînées). NP : no_precip, D : doubt, R : rain, F : fog, P :precip, S :snow. 91

3.5 Description d'AMOSvv, jeux de base pour l'apprentissage par paires des comparaisons relatives au paramètre visibilité. Entre parenthèses figurent les proportions de séquences tirées des archives AMOS. Le jeu de validation VAL_{same} contient des images venant des caméras du jeu d'entraînement ($TRAIN$). Les jeux VAL_{indep} et $TEST$ sont formés à partir de caméras indépendantes. 93

3.6 Paramétrisation des apprentissage pour l'apprentissage par paires. . 96

3.7 Comparaisons entre prédiction par image (fonctions de rang notées vv_sl_xxx, lignes 2-4), prédiction par paire (classifieurs notés vv_pl_xxx, lignes 5-6), et des méthodes pré-existantes (lignes 7-9) présentées section 3.3.4. 104

3.8 Résultats de la prédiction par paire (classifeur vv_pl_vvgg13_du0.0) sur une tâche de détection d'une visibilité inférieure à 250 m. Pour toutes les scènes de TENEBRE, les scores sont calculés sur 12 mois (automne-hiver 2012 et autmone-hiver 2017). Les séries associées aux sites Markstein et Dorans ont été exclues, faute de disposer de données fiables sur toute la période. Les deux dernières lignes, tirées de [36], sont obtenues sur des ensembles de caméras autoroutières. La dernière colonne indique le nombre d'événements rejetés parmi le nombre total de rejets. 108

4.1 Effectifs du jeu d'entraînement d'AMOSvvExt comparé à celui d'AMOSvv. 114

4.2 Comparaisons entre fonctions de rang entraînées sur AMOSvv (lignes 2-4), et fonctions de rang entraînées sur AMOSvvExt (lignes 5-7). 114

4.3 Ligne 2 : scores atteints par les trois fonctions de rang à valeurs réelles de la table 4.2. Ces fonctions de rang forment le groupe 0. Ligne 3 : score obtenu par moyennage des fonctions du groupe 0. Lignes 4 - 12 : scores des fonctions de rang bivaluées. Les lignes 7 et 12 sont obtenues par moyennage sur les groupes 1 et 2. Ces groupes diffèrent principalement par les fréquences de présentation (troisième colonne). 124

4.4 Performances sur le jeu TENEBRE après étalonnage. Pour les prédictions, la fonction de rang bivaluée a été étalonnée avec les mesures colocalisées. Les lignes 2 et 5 indiquent les taux d'appartenance et les tailles médianes des intervalles « prudents » et « fins » (équations (4.12-4.13)). Pour les lignes 3-6, ce taux est calculé après restrictions aux visibilités mesurées inférieures à 5000m. Les lignes 8-13 contiennent les erreurs relatives moyennes associées aux prédictions $\hat{v}^m$ (équation (4.15)), avec ou sans restriction sur la visibilité mesurée. 138

4.5 Effets d'un étalonnage sur des mesures distantes sur les scores (taux d'appartenance p_f et erreur relative médiane). Pour les lignes 2,3, l'étalonnage utilise les mesures colocalisées (ce sont les lignes 5 et 11 de la table 4.4). Lignes 4,5, l'étalonnage utilise les visibilités mesurées à Nancy. Lignes 6,7, l'étalonnage utilise les visibilités mesurées à Entzheim. 140

4.6 Table des erreurs relatives après seuillage à 5000 m. Les lignes paires (2-10) contiennent les résultats du modèle entraîné sur des paires intra-séquence à partir desquelles les cibles ont été obtenues (vv_sl_due111.0), les lignes impaires (3-11), ceux du modèle entraîné par intercalibration (cal0103). Les lignes 2-3 contiennent les erreurs relatives médianes après étalonnage sur les mesures locales. Pour les lignes 4-5 (resp. 6-7) les modèles ont été étalonnés sur la scène nancy1 (resp. entzheim1). Dans les ligne 8-9, on indique la moyenne des erreurs relatives médianes sur l'ensemble des scènes après étalonnage sur la scène associée à la colonne. Les lignes 10-13, on donne les erreurs relatives (médiane ou neuvième décile) associées à des visibilités inférieures à 5 km. Les deux dernières lignes de la table indiquent la taille relative des intervalles après étalonnage local. 148

5.1 Description d'AMOSss.1, jeu de base pour l'apprentissage par paire des comparaisons relatives à l'étendue du manteau neigeux. Dans la seconde ligne, on indique les effectifs avant la troisième étape d'annotation (jeu d'entraînement de AMOSss.0). La dernière ligne est obtenue après restriction aux paires d'images sur lesquelles de la neige apparaît. 155

5.2 Description d'AMOSsd.0, jeux de base pour l'apprentissage par paire des comparaisons relatives à l'épaisseur du manteau neigeux. La dernière ligne est obtenue après restriction aux paires d'images sur lesquelles de la neige apparaît. 156

5.3 Effectifs des jeux AMOSss.1 et AMOSssEXT. 165

5.4 Délais moyens (sur tous les épisodes, lors des épisodes de janvier, lors des épisodes de décembre), non-détection (ND), et fausses détections (FD) rencontrées sur les 68 séquences. 173

A-1 Résultats de la troisième étape d'annotation 212

A-2 Nombre de nouvelles relations (comparaisons strictes, incomparabilités et équivalences) 213

A-1 Récapitulatif de la répartition des images par attribut et par jeu. Dans la colonne « clips », nous indiquons le nombre de changement de scènes total dans les séries d'image d'origine. Un travail de racordement en partie automatisé (non présenté dans ce manuscrit) a permis de rassembler les clips associés à une même caméra. Ainsi, les 777 clips extraits d'AMOS sont ramenés à 389 séquences, chacune associée à une caméra différente -voire exceptionnellement, à deux caméras très proches. Le nombre d'onsets, qui n'a été déterminé que pour AMOS, correspond au nombre de « débuts observés » (voir chapitre 5) dans le jeu. Les images avec défaut correspondent à l'ensemble des images comptées dans la table A-5. 216

A-2 Détail par label de la répartition des niveaux d'enneigement. Les labels sont décrits dans l'annexe A. Le colonne snow_ground / dry_road compte les images annotée **snow_ground_dry_road**. La colonne snow_ground / wet_road compte toutes les autres images avec neige sur la route n'étant ni à l'état de traces ni entièrement couvrante. 216
A-3 Répartition des images pour les attributs relatifs à l'état de l'atmosphère. Les labels sont décrits dans l'annexe A. Les deux colonnes « doubt » correspondent l'une à l'occurrence de précipitations, l'autre à la présence dans l'image de traînées dues à des flocons en chute. 217
A-4 Répartition des images en terme de type de scène. 217
A-5 Répartition par attribut relatif à la qualité de l'image. Les labels sont décrits dans l'annexe A. 218
A-6 Répartition par types de scène (jeu AMOSssExt). Les colonnes 2 à 6 sont relatives à la nature des objets présents dans la scène. Campagne (camp.) : pré, champs, forêts de plaine. Plans d'eau (eau.) : il peut s'agir de caméras côtières (only), ou de scènes donnant sur la mer, un lac ou une rivière. Montagne (mont.) : roche affleurant, falaises, reliefs proches (« mainly/only ») ou lointains (« some »). Les séquences spéciales comptent des environnements très particuliers (cataractes, coeur urbain, chantier, base internationale en antarctique, etc). Les trois dernières colonnes décrivent le type de plan. Sol absent (sol abs) : seuls des immeubles, le feuillage des arbres sont visibles. Plan large (large) : cas où la caméra, située en altitude, offre une vue d'ensemble sur une plaine ou une vallée. . . 219
A-7 Instruments utilisés pour l'annotation automatique du jeu TENEBRE_1218. 220
A-8 Description du jeu TENEBRE_1218. 220
A-9 Comparaison entre annotation à la main et mesures sur la détection de neige au sol. La première colonne comptabilise les images associées à une mesure de hauteur de neige (sh) nulle. L'étude porte sur huit des dix séquences de TENEBRE_IH. 221

A-10 Comparaison entre annotation à la main et mesures pour la détection des précipitations. La première (resp.troisième) colonne comptabilise les images associées à un cumul nul au pas de temps 1 minute (resp 6 minutes). L'étude porte sur huit des dix séquences de TENEBRE_IH. 222

A-1 Modèles entraînés à la classification décrits section 5.4. 224
A-2 Meilleurs modèles entraînés à la prédiction par paire sur AMOSvv (voir section 3.3). 224
A-3 Meilleurs modèles entraînés à la prédiction par image sur AMOSvv et AMOSvvExt (voir sections 3.3, 4.1 et 4.2). 224
A-4 Meilleurs modèles entraînés à la prédiction par paire sur AMOSss.0 et AMOSss.1 (voir section 5.2). 225
A-5 Meilleurs modèles entraînés à la prédiction par image sur AMOSss.0, AMOSss.1 et AMOSssExt (voir sections 5.2 et 5.3). 225
A-6 Meilleurs modèles entraînés à la prédiction par paire sur AMOSsh.0, (voir section 5.2). 225
A-7 Meilleurs modèles entraînés à la prédiction par par image sur AMOSsh.0, (voir section 5.2). 225

A-1 Matrice de confusion sur le jeu de validation SYN2000 du modèle SYN_slbiv_VGG11.0, à l'époque 500. La classe $\dot{\perp}$ a été redécoupée (voir texte). Les vecteurs δ_{ij} correspondent à l'encodage des positions relatives d'intervalles défini sur la figure 4.3). 239
A-2 Matrice de confusion sur le problème à six classes pour le modèle vv_sl_due111.0. Les quatre dernières classes sont des sous-classes de la relation d'incomparabilité. Elles sont définies par les positions relatives des intervalles, chevauchements stricts ou inclusions, encodées dans les $\delta_i j$ définis section 4.2.3. 243
A-3 Prédiction des inclusions dans le jeu de test complet (647 inclusions inférées) et après restriction aux paires d'images comptant au moins une image de mauvaise qualité (417 inclusions inférées). 244

Remerciements

Mes premiers remerciements vont à Philippe Dandin et Christophe Baehr qui m'ont aidé à mettre ce projet de thèse sur les rails. Dans le même élan, je remercie Météo-France d'avoir soutenu financièrement mes travaux pendant ces trois années.

Je veux dire aussi à Yvon Lemaître, à Nicolas Viltard et à Djallel Dilmi ma reconnaissance. Leur accueil généreux et leur ouverture d'esprit m'ont été précieux. Merci aussi à Camille, Mohammed et Flavien, camarades de travail, avec lesquels le partage d'un bureau, d'un matériel ou d'un savoir-faire a toujours été agréable. Ce travail doit beaucoup à Daniel Sombret, qui est resté attentif à l'état d'avancement du projet malgré la distance. Je remercie aussi Lucie Rottner dont les conseils m'ont souvent été utiles.

Enfin, un grand merci à mes encadrants Laurent et Cécile. Leur bienveillance, leur patience et leur confiance m'ont toujours encouragé à poursuivre l'effort malgré les doutes et les déconvenues.

Chapitre 1

Introduction

La météorologie est un facteur de risque important dont dépend la bonne marche de nombreuses activités humaines. Le brouillard, la neige, les précipitations intenses, sont des phénomènes qui perturbent régulièrement les transports routiers. En plaine, l'accumulation de la neige sur les câbles et les toitures menacent l'approvisionnement en énergie et la sécurité des personnes.

Ces phénomènes à fort enjeu sont souvent très localisés. La résolution des observations et des prévisions disponibles peut ne pas suffire à préciser l'ampleur du phénomène à l'échelle locale. Dans ce contexte, le recours à des images prises sur le vif peut s'avérer essentiel.
Avec la densification des réseaux de caméras de surveillance, de webcams publiques et le nombre croissant de photos postées sur les réseaux sociaux, l'image en temps réel est devenue abondante. Elle apporte d'ailleurs déjà un complément d'informations aux prévisionnistes lors des situations de crise, notamment en cas de neige en plaine. Le nombre considérable de caméras et d'images disponibles et le coût important de l'observation humaine et des capteurs dédiés conduisent naturellement à la question de l'extraction automatique des informations météorologiques.
L'estimation de paramètres météorologiques à partir d'images prises par des caméras fixes a d'abord été abordée comme un problème de physique des capteurs à la fin des années 90. Le but est alors d'extraire des informations quantitatives précises à partir d'une caméra contrôlée dont la configuration, la position, l'orientation sont connues et l'entretien assuré.
Plus récemment, la thématique a été reprise sous l'angle de l'Intelligence Artificielle (IA). Le principal défi relevé est alors d'extraire de l'information météorologique à partir d'images singulières [1]. La

1. C'est à dire, des photographies prise sur des scènes différentes, provenant de matériels différents (traduction de l'anglais : « single image »)

diversité des scènes (rurales, citadines, routières...) et des appareils de prise de vue, appelle des performances en généralisation qui, jusqu'à la fin des années 2000, sont restées hors d'atteinte de la machine.

A partir de 2014, la mise en place de grandes bases de données et les progrès du machine learning ont changé la donne. La caractérisation à gros traits de la météorologie apparente sur l'image (weather classification) s'est rapidement développée. Les développements les plus récents s'orientent vers une caractérisation de plus en plus fine. Ce travail de thèse s'est inscrit dans ce mouvement. Le but était d'explorer des méthodes d'apprentissage susceptibles d'extraire une information utile aux prévisionnistes, notamment lors d'épisodes de neige en plaine.
Nous avons cherché à estimer des paramètres d'intérêt à partir de caméras qui n'ont jamais été vues lors de la phase d'apprentissage.

Dans cette introduction, nous revenons d'abord sur les développements historiques de cette thématique. En deuxième partie, nous présentons les outils (réseaux de neurones profonds) et les concepts (stratégies d'apprentissages) sur lesquels nos méthodes sont fondées. Le cadre de la thèse, les problématiques abordées et le plan sont présentés en troisième partie.

1.1 La caractérisation de phénomènes météorologiques d'après l'image : un rapide historique.

La caractérisation automatique de phénomènes météorologiques d'après l'image est une thématique de recherche récente. Les activités de recherche se sont structurées autour de deux grands problèmes. Le premier problème est de pouvoir utiliser une caméra donnée, fixe ou embarquée, comme un capteur météorologique. L'accent est mis sur la précision de l'estimation.
Le second problème consiste à classer des images singulières dans des classes de temps sensible assez grossières. L'accent est alors mis sur les performances en généralisation. Un problème intermédiaire consiste à chercher une caractérisation plus fine sur des images singulières ou sur des séquences d'images venant de caméras quelconques.
Au plan méthodologique, les réseaux de neurones profonds se sont progressivement imposés pour aborder les trois types de problème.

1.1.1 La caméra comme capteur de substitution

1.1.1.1 Approches par traitement d'image

L'idée d'exploiter des caméras fixes ou embarquées pour estimer des paramètres météorologiques émerge à la fin des années 90 [1]. Les capteurs spécifiques sont généralement coûteux, et, lorsque le paramètre s'y prête, une caméra convenablement installée et réglée peut être envisagée comme un capteur de substitution.
Pouir extraire des images l'information d'intérêt, le machine learning n'est pas utilisé avant la fin des années 2000. Les premiers auteurs font plutôt appel à des méthodes de traitement d'image classique. Il s'agit alors de trouver des quantités calculées à partir de l'image numérique -on parle de descripteurs d'image [2]- qui soient bien corrélés au phénomène d'intérêt [2],[3],[4].
Par exemple, le contraste au niveau de la ligne d'horizon est bien corrélé à la visiblité horizontale [1]. Sur les images couleur, l'intensité du canal des bleus est caractéristique de la présence de neige au sol [5]. Certains descripteurs ont pu être fondés sur une modélisation physique de l'influence de la météorologie sur le processus de formation d'image [6],[7],[8]. On considère alors le trajet de la lumière, son interaction avec le milieu (surfaces, hydrométéores), et la caractérisation du phénomène est présentée comme un problème inverse. Par exemple, suite aux travaux de S.Nayar et al. [9], on a utilisé le modèle de Koschmieder pour quantifier les effets des hydrométéores de jour [6] ou de nuit [7].
Les approches par modélisation physique ont donné de bons résultats sur une caméra contrôlée par l'expérimentateur. Mais quant à leur utilisation sur des images venant d'une caméra quelconque, des limites apparaissent.
D'une part, les hypothèses du modèle de formation de l'image sont souvent prises en défaut. Les hy-

2. traduction de l'anglais "visual descriptor"

pothèses physiques ne sont pas toujours vérifiées (par exemple, celle d'une atmosphère homogène, dans la loi de Koschmieder). Sur des caméras d'extérieur, la poussière, les gouttelettes, les flocons sur la lentille, la buée, les incrustations de texte, de logos, corrompent le signal. Les effets d'algorithmes de post-traitement très variés affaiblissent l'hypothèse de proportionnalité entre le signal lumineux et l'intensité du pixel.
D'autre part, l'inversion complète du modèle dépend de paramètres supplémentaires, propres à la caméra et à la géométrie de la scène. Par exemple, dans [8] le rayon réel des gouttes est déduit des traînées à partir de la distance focale. Dans [6], [7], la visibilité ne peut être déduite de l'épaisseur optique que lorsque la profondeur de la scène est connue. De façon générale, pour passer d'un descripteur corrélé au paramètre d'intérêt à une estimation quantitative, une étape d'étalonnage sur des données instrumentales est nécessaire.

1.1.1.2 Méthodes pilotées par les données. Bases météo-image.

Les méthodes évoquées précédemment ont été appliquées à des séries temporelles associant image et mesure (base de données « météo-image » [10]). Dans le cas de la visibilité, des exemples de bases de données météo-image sont fournies par les bases Matilda [11], WILD [12] ou SOMT [13].Ces bases sont issues de caméras contrôlées par l'expérimentateur et de capteurs de bonne qualité, peu distants l'un de l'autre. Elles ont permis de valider plusieurs descripteurs d'images différents basés sur des histogrammes locaux [4], des mesures de contrastes [14], [15] ou la détection de contours [16].
Pour d'autres paramètres météorologiques, des bases météo-images ont été construites à partir d'archives webcam (caméras non contrôlées) et de données météorologiques distantes. C'est en partie dans ce but que l'équipe de N.Jacobs a entrepris la collecte de nombreuses archives webcam dès le début des années 2000 [17]. Aujourd'hui, cette base appelée AMOS (Archive of Many Outdoor Scenes) contient les archives de plus de 30.000 webcams. Des bases météo-images ont ainsi été construites à partir de quelques caméras d'AMOS situées à proximité de capteurs météorologiques pour aborder l'estimation de paramètres originaux, comme la vitesse du vent [17], la nébulosité [18] ou la température [19]. Ce seront d'ailleurs les archives AMOS qui constitueront la principale source d'images utilisée dans cette thèse.

1.1.1.3 Utilisation du Machine Learning

Pour améliorer les performances sur les bases météo-image à disposition, on a cherché à exploiter une combinaison de descripteurs plutôt qu'un descripteur unique. La question du choix d'une telle combinaison a été abordée de manière empirique par apprentissage automatique (machine learning).
Les premières approches par machine learning datent de la fin des années 2000. Pendant les années 2010, ces approches vont se multiplier et le nombre de descripteurs utilisés, la complexité des modèles vont aller croissant.

Deux voies se dessinent alors. Une première série de travaux se donne pour objectif d'améliorer les performances sur les bases météo-image existantes. Les auteurs visent une caractérisation fine de la météorologie sur des images venant d'une seule caméra [14],[20],[21],[22]. Le champ des recherches est étendu à de nouveaux paramètres météorologiques, a priori difficiles à estimer d'après l'image, comme la température [19],[23],[24] ou l'humidité [25]. Côté méthode, on note que l'utilisation des aspects temporels permet d'améliorer les performances [22],[24] tandis que les réseaux de neurones deviennent progressivement les modèles les plus utilisés [24],[19],[23]. Dans ces études, les performances en généralisation sur des scènes et des caméras de modèles différents ne sont pas évaluées.
Au contraire, la deuxième voie est tournée vers la question de la généralisation à des scènes et des appareils quelconques, quitte à viser une caractérisation plus grossière, facilement accessible à un humain. Cette deuxième voie est celle de la classification du temps sensible.

1.1.2 La classification du temps sensible

1.1.2.1 Des problèmes d'apprentissage

La caractérisation du temps sensible [3] est abordée comme un problème d'intelligence artificelle. Il s'agit de classer des images provenant de scènes très variées dans des classes de temps sensible (« ensoleillé », « couvert », « pluvieux », etc) comme le ferait un observateur humain. Les premiers travaux [26],[27] sont encore intermédiaires : les images à traiter sont issues d'un même modèle de caméra embarquée. Elles sont donc relativement homogènes. Cependant, les cibles de l'apprentissage sont produites à la main.
C'est un peu plus tard que le nombre et la diversité des images, la publication de la base collectée, permettent de définir des problèmes d'apprentissage (voir par exemple [28] et [29]).

FIGURE 1.1 – le problème de la classification binaire « ensoleillé »/« nuageux » sur des images singulières est difficile : l'humaine peut tirer parti de plusieurs caractéristiques de l'image, comme la présence ou l'absence d'ombres portées.

Les études se sont d'abord focalisées sur les phénomènes météorologiques les mieux représentés dans les sources d'image, comme la discrimination entre temps ensoleillé et nuageux (figure 1.1). Depuis 2015, le nombre de classes s'est accru (voir par exemple : [30], [31], [32]).

3. En vision par ordinateur, on utilise le mot clef « Weather Classification »

Pour construire le jeu d'apprentissage, la démarche type consiste alors à rassembler un grand nombre d'images d'extérieur (au moins 10.000) prises sur des scènes et par des appareils différents à partir de sources comme Flickr, Pixabay, Picasa, MojiWeather, Poco, Fengniao, etc. Les images sont classées à la main via des plateformes de crowdsourcing, suivant des modalités propres à garantir la fiabilité du label.
Des variantes existent. On a pu, par exemple, définir le label à partir de données instrumentales, qu'elles soient obtenues par télédétection [33], [34] ou par des mesures ponctuelles faites à distance [25].
Par exemple pour la détection de la neige au sol, Wang et al. [34] utilisent comme cible de l'apprentissage un produit de classification satellitaire. La correspondance avec l'image est mauvaise et les erreurs d'annotation sont fréquentes. Les mauvaises performances en prédiction qui en résultent sont compensées par l'intégration temporelle et spatiale sur les zones et les intervalles de temps ciblés.
Dans une étude remarquable, Chu et al. [25] étudient un problème de classification à cinq classes (ensoleillé, nuageux, brumeux, pluvieux, neigeux). Ces classes de temps sensible ont été fournies par un opérateur privé (Weather Underground), à partir de stations distantes d'au plus 4 km. Leur jeu de données Image2Weather comporte 180.000 images mais les images des classes "brumeux" et "neigeux" sont moins représentées que les autres (1.500 chacune).

Les images webcam ont aussi fait l'objet d'une annotation à la main. Par exemple, sur le problème de classification binaire neige au sol/non neige au sol, Kosmala et al. [35] ont organisé l'annotation de 170.000 images issues d'un ensemble de 133 webcams relativement homogènes en terme de scène (scènes rurales) et de matériel. Sur des webcams laissées de côté pendant l'entraînement, les performances sont sensiblement dégradées.

Au plan méthodologique, les réseaux de neurones profonds se sont montrés au moins aussi bon que les approches par SVM [24], [29], [23], et les performances atteintes permettent d'envisager la classification automatique des situations météorologiques les plus communes. Des applications concrète en météorologie et en climatologie ont d'ailleurs été mises en avant par plusieurs auteurs [34], [35]).
Mais pour obtenir des prédiction utiles à la surveillance des phénomènes à enjeux, une caractérisation plus fine est nécessaire. Il ne suffit pas, par exemple, de distinguer le brouillard du temps clair, mais de pouvoir affirmer que la visibilité a passé un certain seuil et de donner des tendances. Pour cela, il faudrait idéalement disposer d'une base météo-images portant sur de nombreuses scènes et associées à des mesures de qualité. Or, à notre connaissance, il n'existe aucune base de ce type.
Pour contourner ce problème, trois approches distinctes ont été proposées.

1.1.2.2 Vers des prédictions plus fines sur une scène et un instrument quelconque

Dans le cas où on dispose d'archives webcam sur un nombre de scènes limité (moins d'une centaine) et que les images ont pu être associées à une mesure fiable, on a cherché à augmenter les performances en généralisation en dégradant la qualité de l'image d'entrée. Par exemple, Pagani et

al. [36] cherchent à détecter le brouillard épais avec un perceptron multicouche entraîné sur moins d'une centaine de séquences webcam. Ces auteurs extraient des patches (28*28 pixels) et les floutent, dans l'idée de prévenir le surapprentissage des scènes du jeu d'entraînement. Cependant, ce procédé à contre-courant des méthodes d'apprentissage récentes a le désavantage de réduire l'information disponible. De plus, il ne semble pas résoudre le problème dans la mesure où les performances restent significativement moins bonnes sur des scènes indépendantes du jeu d'entraînement [36].

La deuxième approche consiste à créer une base météo-images à partir de données de mesures faites à distance et d'exploiter une stratégie d'apprentissage robuste vis-à-vis des cibles bruitées. En effet, la mesure faite à distance est au moins légèrement corrélée à la valeur locale du paramètre. L'idée est de profiter de cette corrélation à travers une méthode d'apprentissage adaptée.
Par exemple, Islam et al. [18] modélisent le défaut de correspondance entre la mesure locale et la mesure distante (bruit de cible) par une permutation aléatoire sur les images. Ces auteurs observent qu'un modèle entraîné à la régression sur les cibles distantes permet de réordonner les images de façon à améliorer la correspondance avec la mesure distante.
Cependant, cette stratégie d'apprentissage en deux temps, avec une première étape de correction par permutation des labels, n'a jamais été évaluée sur des bases météo-image fiables.

La troisième approche consiste à affiner l'annotation à la main. Plutôt que de démultiplier les classes dont les séparations seraient nécessairement arbitraires, ou de demander une estimation quantitative à l'annotateur, tâche particulièrement difficile [37], on mise sur une estimation relative. L'annotation porte alors sur des paires d'images. Pour chaque paire, l'annotateur doit préciser si le paramètre croît ou décroît d'une image à l'autre ce qui est autrement plus simple.
A partir de ces paires d'images, des méthodes du learning to rank permettent d'estimer un indice corrélé à la quantité d'intérêt. You et al. [37] prédisent ainsi un indice de visibilité sur des scènes et des caméras très variées. Ils étalonnent ensuite cet indice par réapprentissage [4] du modèle sur des données mesures ponctuelles de bonne qualité.

C'est dans ce contexte qu'a débuté la thèse. Ce travail explore principalement la troisième approche, dans un esprit tourné vers la production de données d'opportunité exploitables à partir de caméras et de scènes très variées.

4. voir section 1.2.3

1.2 Perspectives offertes par le machine learning

Pour expliquer les orientations qui ont été suivies pendant la thèse, nous allons présenter certains concepts du machine learning et de la vision par ordinateur. Il ne s'agit pas d'une présentation exhaustive, mais plutôt d'un inventaire ciblé de la vaste boîte à outils du machine learning. Nous présentons d'abord quelques-uns des grands problèmes qui structurent la vision par ordinateur.
Nous présentons ensuite le deep learning dont les méthodes ont bousculé le domaine en profondeur. Cette présentation ne porte que sur les éléments de base : les réseaux de neurones profonds et la procédure d'apprentissage supervisé standard.
Enfin, nous évoquons des stratégies d'apprentissage qui dépassent le cadre de la vision par ordinateur et qui sont au coeur de ce travail de thèse.

1.2.1 Problèmes de la vision par ordinateur abordés par machine learning

Les applications du machine learning à la vision par ordinateur sont structurées autour de grands problèmes d'apprentissage supervisé.
Un problème d'apprentissage supervisé se présente sous la forme de couples « prédicteurs-cible » notés (x_i, y_i) répartis dans deux ensembles : le jeu d'apprentissage et le jeu de test. Ce dernier est assorti d'une (de) méthode(s) d'évaluation des scores. Les prédicteurs x_i forment un échantillon du « domaine des prédicteurs », noté $\mathcal{X}$. En vision par ordinateur, les prédicteurs sont des images, ou éventuellement des paires d'images, voire des séries d'images. Le domaine des cibles, que nous noterons $\mathcal{Y}$, caractérise la nature du problème (classification ou régression).
Souvent, on considère que $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ est muni d'une loi de probabilité $L_{\mathcal{D}}$, d'où procèdent les deux jeux de données. Dans l'idéal, on dispose de jeux bien équilibrés et représentatifs. Un jeu parfaitement équilibré correspond, par définition, à une loi marginale $L_{\mathcal{Y}}$ équirépartie. Un jeu est représentatif lorsque les x_i suffisent à bien « connaître » la loi marginale $L_{\mathcal{X}}$. Cependant, $L_{\mathcal{X}}$ n'étant pas accessible en pratique, cette loi n'est invoquée que pour donner des preuves de concept théoriques.

On aborde un problème d'apprentissage en « entraînant » une fonction paramétrée $f(x_i; w) = z_i$ (un « modèle ») sur le jeu d'apprentissage. Pendant cette phase d'apprentissage, les données test ne sont pas disponibles. Ce n'est qu'une fois le modèle entraîné qu'il est évalué sur le jeu de test.
Le but de l'entraînement est d'obtenir les meilleurs scores sur le jeu de test. Pour cela, le jeu d'apprentissage est lui-même découpé en deux (partitionnement) : un « jeu d'entraînement », sur lequel on cherche à minimiser l'écart entre les sorties z_i et les cibles y_i, et un jeu de « validation », qui préfigure le jeu de test, et grâce auquel on contrôle les performances en généralisation du modèle.
Sur le jeu d'entraînement, la minimisation est généralement faite suivant une méthode itérative. L'écart

à minimiser est calculé avec une « fonction de coût [5] », choisie en fonction de la nature des cibles et des sorties, de la méthode itérative utilisée et du score à optimiser.

Le choix d'un ou de plusieurs partionnements du jeu d'apprentissage et la méthode de sélection des poids optimaux font l'objet d'une littérature abondante [38]. De très nombreuses techniques existent. Mais pour le deep learning, compte tenu de la taille des jeux et de la durée des apprentissages, c'est la technique la plus simple qui est retenue : le partionnement est fixé une fois pour toutes et ce sont les paramètres qui minimisent la fonction de coût sur le jeu de validation qui sont sélectionnés.
En dehors du deep learning, les applications du machine learning à la vision par ordinateur, s'appuient sur deux autres catégories de modèles : les machines à vecteur de support (SVM [39]) et les arbres de décision [40]. Pour les deux premières catégories de modèles, la dimensionnalité des prédicteurs est généralement réduite à l'aide de descripteurs d'image adaptés au problème. La fonction $f(x_i, w)$ est alors cherchée sous la forme :

$$f([d_0(x_i), d_1(x_i), ...d_k(x_i)], w)$$

où les d_j sont les descripteurs.
Avec les réseaux neurones profonds, toute la chaîne de prédiction en partant de l'image est paramétrée. On parle d'apprentissage de bout en bout [6]. En travaillant avec ces modèles, on se trouve donc allégé des efforts de construction et de sélection de descripteurs convenables. En contrepartie, des efforts importants sont consacrés à la construction des jeux de données, c'est à dire à la construction du "problème d'apprentissage". Les procédures d'apprentissage sont aussi plus complexes.

Certains problèmes d'apprentissage ont joué un rôle prépondérant dans l'histoire récente de la vision par ordinateur. Ce sont des problèmes associés à des jeux de données publiés accessible à tout chercheur, sur lesquels un grand nombre de méthodes ont pu être comparées. Nous présentons quelques exemples dans les paragraphes suivants.

1.2.1.1 Classification multiclasse

Lorsque $\mathcal{Y}$ est un ensemble de catégories mutuellement exclusives (par exemple « nuageux »; « ensoleillé » , etc) on parle de problème de classification. La classification est dite binaire lorsque $|\mathcal{Y}| = 2$, multi-classe sinon. La présentation d'un problème de classification multiclasse bien connu va nous permettre d'illustrer les notions de jeux d'apprentissage, de méthode d'évaluation et de fonction de coût.

Problème posé lors de l'ILSVRC 2012 :
Un célèbre problème de classification d'image est associé au challenge ILSVRC [7] de 2012. Le jeu

5. *Cost function*. On rencontre aussi les termes de fonction de perte (*loss function*) et de fonction objectif (*objective function*). Des nuances existent entre ces termes. Certains auteurs considèrent par exemple que la fonction de coût intègre des pertes et des termes supplémentaires de régularisation. Mais les définitions ne sont pas fixées.

6. end-to-end

7. Imagenet Large Scale Visual Recognition Challenge

d'apprentissage du problème de classification a été construit à partir de la base de données ImageNet [41]. Il comportait un jeu d'entraînement de 1,2 M d'images et un jeu de validation de 50 K images. Chaque image est une photo couleur de bonne qualité centrée sur un objet représentant une classe d'objet parmi mille.
Lors de cette compétition, un réseau de neurone profond (AlexNet) a pris le pas sur les autres types de modèle [42]. Cette avance, considérée aujourd'hui comme historique, a été mesurée par une procédure d'évaluation adaptée à un problème de classification multiclasse généraliste.

Evaluation d'une classification multiclasses :
Pour les problèmes avec un grand nombre de classes, la sortie des modèles est en général un vecteur à composantes positives, de somme 1, représentant une distribution de probabilité sur $\mathcal{Y}$. La procédure d'évaluation utilisée dépend de l'application visée, mais on utilise souvent l'erreur $top-r$ définie par :

$$top - r = \frac{1}{n}\sum_{k=0}^{n} \min_{j\in[1,r]} d(\hat{y}_{kj}, y_k)$$

où n est la taille du jeu de données, les y_k représentent la classe vraie associée à l'image k, les $\hat{y}_{kj}$ représentent les j classes les plus probables proposées par le modèle pour l'image k et $d(\hat{y}, y) = 0$ si $\hat{y} = y$, 1 sinon.
A l'ILSVRC 2012, l'AlexNet présentait par exemple une erreur $top-5$ de 16 %, soit dix points de moins que les approches concurrentes. Depuis 2012, la position dominante des réseaux de neurones s'est continuellement renforcée. Cette tendance a aussi été observée en classification du temps sensible [29], [23], [43].

Fonction de coût pour la classification :
Pour mesurer un écart entre la sortie du modèle (un vecteur noté z) et la classe cible y, la fonction de coût la plus souvent utilisée est l'entropie croisée. Cette dernière a la propriété d'être différentiable par rapport aux composantes de z et de se prêter à un apprentissage par descente de gradient (voir section 1.2.2.1). Pour un couple (x, y) du jeu d'entraînement, elle est définie par :

$$\mathcal{L}(z, y) = -ln(z(y)) \tag{1.1}$$

où $z(y)$ est la probabilité de la classe y sous la loi définie par $z = f(x, w)$. Au cours de l'entraînement, les classifieurs implémentés sur des réseaux de neurones profonds, comme le réseau AlexNet, sont contraints à minimiser la valeur moyenne de cette fonction de coût sur le jeu d'apprentissage.

1.2.1.2 Classification multilabel et segmentation sémantique :

La classification d'images peut être déclinée sous d'autres formes. Dans un problème de classification « multilabel », les classes ne sont pas toutes mutuellement exclusives. Le domaine cible $\mathcal{Y}$ est

l'ensemble des parties de C où C l'ensemble des classes sémantiques. Ce type de classification a par exemple été utilisé par Laffont et al. [44] pour caractériser des images par un nombre arbitraire d'attributs, choisis dans une liste contenant en particulier (*snowy*, *hazy*, etc).

Dans un problème de segmentation sémantique, les cibles ont les mêmes dimensions spatiales que l'image. A chaque pixel est associé une classe d'objet. L'annotation complète d'images pour la segmentation sémantique est donc beaucoup plus fastidieuse et les jeux de données disponibles sont de taille nettement plus petite, de l'ordre de 10.000 images.
Les méthodes d'évaluation et la fonction de coût sont assez proches de celles utilisées pour la classification bien que, pour éviter la surreprésentation de certaines classes, une forme de pondération intervienne souvent (voir par exemple l'IoU -Intersection over Union- utilisée pour l'évaluation).

1.2.1.3 Problèmes de régression

Lorsque le domaine des cibles est un intervalle de $\mathbb{R}$ on parle de problème de régression. Là encore, il est possible de définir des variantes. En particulier, la cible peut être réduite à un réel ou avoir les mêmes dimensions spatiales que l'image.

En vision par ordinateur, le débruitage d'image [8] constitue une importante classe de problèmes de régression par pixel. La cible consiste typiquement en une version « propre » de l'image d'entrée. Des exemples connexes à notre thématique sont fournis par les problèmes de *dehazing* [45], de *deraining* [46] et de *desnowing* [47],[48], qui consistent à débarrasser l'image des effets des hydrométéores. Pour ces tâches, l'écart quadratique moyen et la MAE sont souvent utilisés, aussi bien comme pour le calcul des scores que comme fonctions de coût différentiables. Dans cette thèse, des tâches de cette nature sont illustrées dans le chapitre 5 (section 5.7) et dans l'annexe F (débruitage d'images radar).

1.2.1.4 Apprendre des relations : similarité et préférences

Les cibles peuvent être relatives non à des images, mais à des paires ou à des listes d'images. On parle alors d'apprentissage par paires (ou par liste) [9]. Trois types de problèmes peuvent être distingués :

— L'identification automatique. Par exemple, pour la reconnaissance faciale, le jeu contient des paires de photos de personnes prises dans des conditions différentes [49]. Les cibles sont les labels « identique » ou « différent ».
— L'apprentissage des préférences [10] [50]. Les cibles sont aussi relatives à des couples d'images mais elles indiquent une préférence. On peut formuler ce type de problème lorsque les phénomènes d'intérêt échappent à la classification [51]. Par exemple, un visage peut paraître plus souriant qu'un autre mais il est difficile de ranger une image de sourire dans une classe d'in-

8. image denoising
9. pairwise learning/listwise learning
10. preference learning

tensité. Les aspects qui se plient mieux à la comparaison qu'à la classification sont appelés attributs relatifs. Dans la première section, nous avons vu que You et al. avaient considéré la visibilité comme un attribut relatif.

— Dans certains cas, le problème n'est pas défini par des paires comparées, mais par des listes ordonnées [52]. Ce type de problème se rencontre dans le contexte des données de navigations récupérées par un moteur de recherche en ligne [11]. Pour améliorer un moteur de recherche par apprentissage, on peut cibler les arrangements suivant lesquels les utilisateurs ont parcouru les sites proposés.

Sur ces trois types de problèmes, deux approches peuvent être définies.

1.2.1.5 Apprentissage par paires : prédiction par paire vs. prédiction par image

Les problèmes précédents, qui sont essentiellement des problèmes de classification, peuvent être abordés comme tels. Le modèle s'écrit alors : $f(x_i, x_j; w) = z_{ij}$. Dans le cas de l'identification, $z_{ij} \in \{\dot{=}; \dot{\neq}\}$ et dans le cas des préférences, $z_{ij} \in \{\dot{>}; \dot{<}\}$ (le point permet de distinguer la catégorie cible de la relation binaire associée). Dans le contexte d'un apprentissage des préférences, contexte dans lequel nous nous placerons souvent, nous parlerons de « prédiction par paire ».

La deuxième approche consiste à présenter les images une par une au modèle. L'erreur est calculée à partir des vecteurs de sortie $f(x_i; w)$, $f(x_j; w)$ et de la cible y_{ij}. Nous parlerons de prédiction par image.

Dans un contexte d'identification automatique, la fonction de coût est choisie de façon à pénaliser des vecteurs de sortie "proches" (respectivement "éloignés") lorsque les entrées sont différentes (respectivement "identiques").

Pour minimiser la fonction de coût, le modèle doit organiser l'espace de sortie de manière à ce que la distance reflète une similarité entre les objets (ou les personnes) sur les images. Ce type de tâche est d'ailleurs appelé apprentissage de métrique [12].

Par exemple, dans [53], il s'agit de distinguer des signatures authentiques (vraies) de leur copies (fausses) à l'aide de « réseaux de neurones siamois ». Chaque signature est représentée par une matrice de taille 8×200 où sont encodés les mouvements d'un stylet. Lors de l'apprentissage, des paires de signatures (x_i, x_j) sont présentées à un réseau de neurone $f(.; w)$. La fonction de coût évalue l'écart entre la quantité $cos(f(x_i; w), f(x_j; w))$ et une cible qui dépend des signatures de la manière suivante : pour les paires de signature vraie-fausse, la cible est -1, pour les paires vraie-vraie, la cible est $+1$. Après apprentissage, le réseau sépare les paires vraie-fausse par un angle supérieur à $\pi/2$.

Lors de l'apprentissage, tout se passe comme si deux réseaux partageant les mêmes poids traitaient chacun une image, d'où le terme de « réseaux de neurones siamois ».

Une approche par réseaux siamois peut aussi être employée dans un contexte d'apprentissage des préférences. Dans ce cas, le modèle est amené à organiser l'espace de sortie comme une échelle ordinale.

11. « clickthrough data »
12. metric learning

Par exemple, dans un des problèmes abordés dans [54], l'annotation a consisté à comparer des images de chaussure de modèles différents. Pour chaque paire d'images (x_i, x_j), l'annotateur a précisé quelle chaussure lui paraissait la plus confortable.
Le modèle est une fonction $f(., w)$ à valeurs réelles implémentée sur des réseaux siamois. Pendant l'apprentissage, l'ordre des sorties $f(x_i; w) = z_i$ et $f(x_j; w) = z_j$ est comparé à celui de l'annotation à travers la fonction de coût $\mathcal{L}(y_i, z_i, z_j) = [y_i \times (z_i - z_j)]_+$. Avec, ici, $y_i = -1$ si la chaussure de x_i a été annotée comme plus confortable à celle de x'_i et $y_i = 1$ sinon.
Après apprentissage, la fonction $f(.; w)$ a appris une « échelle ordinale » relative au critère d'intérêt.
Ce type de fonction est appelé « ranking function » (ou simplement « ranker »). Dans ce manuscrit, nous utiliserons le terme de « fonction de rang ».

Pour ces problèmes comme pour les précédents, l'approche par réseaux de neurones profonds domine l'état de l'art.

1.2.2 Le deep learning

Le deep learning est un terme qui englobe une catégorie de modèles, les réseaux de neurones « profonds », et l'ensemble des méthodes qui leur est propre.
Un réseau de neurones profond est caractérisé par son architecture. La section 1.2.2.1 décrit les architectures classiques du deep learning utilisées dans le cadre de cette thèse.
L'entraînement d'un « gros » réseau de neurones fait appel à des techniques d'optimisation par descente de gradient. Avec un jeu d'entraînement de grande taille, il préférable de restreindre le calcul des gradients à des sous-échantillons pris au hasard (descente de gradient stochastique). La procédure d'entraînement standard est présentée en section 1.2.2.2.
L'entraînement se distingue cependant d'un problème d'optimisation classique en ce sens qu'il s'agit d'obtenir un modèle dont les performances sont optimales sur des images non apprises (généralisation). Les techniques utilisées pour limiter le surapprentissage sont présentée dans la section 1.2.2.3. La question de la qualité et de la quantité des données nécessaires est traitée dans la la section 1.2.2.4. Comme ce travail de thèse ne contient rien d'original sur ces sujets, cette présentation est limitée à l'essentiel.

1.2.2.1 Les réseaux de neurones profonds

Un réseau de neurones profond peut être défini comme une cascade de filtres -au sens du traitement du signal- dont les poids sont mis à jour par descente de gradient. Ces filtres sont organisés en couches de neurones associées à d'autres opérateurs. Nous limiterons la présentation aux deux types de couches de neurones les plus utilisées, les couches complètement connectées [13] et les couches de convolution [14],

13. fully connected layers
14. convolutional layers

et aux opérations de ré-échantillonnage.
Un neurone peut être défini par la combinaison d'une application affine et d'une application non linéaire, la « fonction d'activation ». Les coefficients de l'application affine sont les « poids » du neurone (on les notera w_k), analogues à des poids synaptiques dans le modèle connexionniste. Dans ce même modèle, la fonction d'activation représente la réponse non linéaire du « neurone » à la somme pondérée des potentiels d'action reçus. Dans la suite, on la note $\mathcal{A}$ de façon générique.

Couche complètement connectée :
Pour un vecteur d'entrée dans $\mathbb{R}^n$, $\boldsymbol{x} = (x_i)_{i=1..n}$, une couche complètement connectée à m neurones est définie par l'application à valeurs dans $\mathbb{R}^m$:

$$f^{c.c.}(\boldsymbol{x}; w) = \left[\mathcal{A}(w_{k,0} + \sum_{i=1}^{n} w_{k,i} x_i)\right]_{k=1..m} \tag{1.2}$$

où les $w_{k,i}$ sont les poids du neurone k. Ces couches se retrouvent à la fin des réseaux entraînés à la classification (voir figure 1.2) ; m correspond alors au nombre de classes. Les réseaux constitués d'une succession de couches complètement connectées sont appelés perceptrons multicouche.

Les fonctions d'activation utilisées dans les réseaux de neurones profonds sont simples, non-paramétrées. En dehors de la dernière couche, l'activation la plus utilisée est la fonction partie positive (aussi nommée ReLU [15]). Pour la dernière couche d'un réseau, la fonction utilisée dépend de la tâche. Pour une tâche de classification, c'est généralement la fonction *softmax* qui est appliquée au vecteur de sortie [16] pour obtenir une distribution de probabilité sur l'ensemble des classes.

$$softmax((y^k)_{k=1..m}) = \left[\frac{e^{y^k}}{\sum_{j=1}^{m} e^{y^j}}\right]_{k=1..m} \tag{1.3}$$

où m est le nombre de neurones dans la couche de sortie, y^k la sortie de la partie affine du neurone k. Combinée à la fonction *softmax*, l'utilisation de l'entropie croisée est souvent justifiée par analogie avec la régression logistique.

Couches de convolution :
Une « couche de convolution » 1D à un seul neurone est définie par :

$$f^{conv1d}(\boldsymbol{x}; \boldsymbol{w}) = \mathcal{A}(\boldsymbol{y})$$

où :

$$\boldsymbol{y} = \left(w_0 + \sum_{t=0}^{r} w_t x_{i+t}\right)_{i=1..n}$$

15. pour REctified Linear Unit
16. Dans ce cas, l'activation n'est plus propre au neurone mais à la couche toute entière.

Le vecteur des w_k, de taille r, est appelé noyau [17]. Pour pouvoir calculer la somme lorsque $i > n - r$, les x_i peuvent être complétés au bord par des valeurs nulles [18]. De nombreuses variantes existent [55]. En vision par ordinateur, les couches de convolution sont appliquées à des matrices $\boldsymbol{x}$ de taille $c*n*n'$, où c désigne le nombre de canaux, n et n', des dimensions spatiales. Dans le cas d'une couche à m neurones, le noyau W^k associé au neurone k est alors une matrice cubique de taille $c \times r \times r$ et le signal de sortie est une suite finie de m matrices de taille $n \times n'$:

$$f^{conv2d}(\boldsymbol{x}; \boldsymbol{w}) = \Big[\mathcal{A}(\boldsymbol{y^k})\Big]_{k=1..m}$$

où :

$$\boldsymbol{y^k} = \left(w_0^k + (W^k \star \boldsymbol{x})_{i,j}\right)_{i=1..n, j=1..n'} \quad \text{et} \quad (W^k \star \boldsymbol{x})_{i,j} = \sum_{\substack{d\in[1,c] \\ s,t\in[1,r]}} W^k_{d,s,t} \times x_{d,i+s,j+t} \tag{1.4}$$

Ici encore, les fonctions d'activation sont généralement des fonctions ReLU appliquées composante à composante. Les m matrices résultantes sont appelées « cartes de caractéristiques [19] ». En pratique, les noyaux sont de petite taille ($r \in \{3; 5; 7\}$) alors que le nombre de neurones par couche croît avec la profondeur de la couche, jusqu'à plus de mille.

Opérateurs supplémentaires (agrégation spatiale)

D'autres opérations consistent à modifier la dimension du signal d'entrée. Pour une tâche de classification, la dimension spatiale du signal d'entrée est progressivement réduite à un vecteur de taille $m = |\mathcal{Y}|$. Cette réduction se fait par des opérations d'agrégation spatiale (« max pooling », « average pooling », etc) qui sont généralement employées en alternance avec les couches de convolution. Après chaque étape d'agrégation, l'échelle de traitement est un peu plus globale. Par exemple, l'opération la plus fréquente (max pooling) consiste à découper la carte de caractéristique en carrés de 2×2 pixels et à prendre la valeur maximum sur chacun des carrés. Après un max pooling, les dimensions sont donc divisées par deux.

A contrario, il est possible de projeter sur des cartes de plus grandes dimensions spatiales, en particulier pour des tâches réalisées à l'échelle du pixel (segmentation, régression, débruitage). Il peut s'agir d'une opération d'interpolation bilinéaire. Il est possible de paramétrer cette opération. C'est le rôle des couches de convolution transposée [20] [55].

Enfin, le passage d'une couche de convolution à une couche complètement connectée pose un problème particulier : les cartes de caractéristiques, dont les dimensions spatiales dépendent de celles de l'image d'entrée, doivent être converties en un vecteur 1D de taille fixée. Le détail de cette opération est spécifique à l'architecture.

17. Il ne s'agit pas rigoureusement d'un noyau de convolution : la somme de l'équation 1.4 correspond plutôt à une corrélation croisée
18. 0-padding
19. feature maps
20. transpose convolution

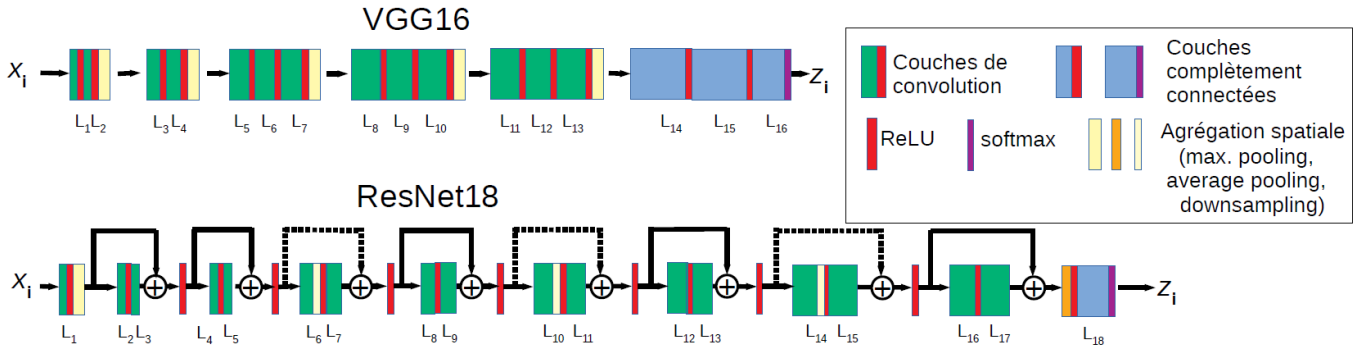


FIGURE 1.2 – Deux exemples de réseaux de neurones profonds à couches de convolutions : VGG16 [56] et ResNet18 [57].

Exemples d'architectures

La plupart des architectures peuvent être décrites à partir de ces trois types de couche (complètement connectée, convolution, agrégation). A titre d'illustration, on donne figure 1.2 une vision schématique des types d'architectures qui ont remporté les ILSVRC 2014 (VGG [56]) et ILSVRC 2015 (ResNet [57]) et permis d'approcher les compétences humaines sur des problèmes de classification d'image.

Les architectures VGG (figure 1.2) sont construites sur une cascade de blocs convolution-agrégation (max-pooling) terminée par un perceptron. Les blocs sont constitués de couches de neurones de taille variable. Le nombre de neurones par couches augmente tandis que les dimensions spatiales du signal intermédiaire diminuent à cause des max-pooling successifs. Par exemple, en sortie de la couche L_{12}, le signal est de taille $(512, \frac{n}{16}, \frac{n'}{16})$ où 512 représente le nombre de neurones de la couche L_{12} et $n \times n'$ les dimensions spatiales de l'image d'entrée.
Dans l'architecture ResNet (figure 1.2), les sorties des blocs sont ajoutées aux entrées (symboles +), d'où le nom de réseaux à résidus [21].
Dans l'architecture de type U-net [58], présentée sur la figure 1.3, initialement proposée pour de la segmentation sémantique, la dimension spatiale décroît dans la première partie du réseau (encodeur), et croît jusqu'à la taille initiale dans la seconde partie (décodeur), à travers des opérations de convolution transposée. En parallèle, le nombre de canaux augmente et les dimensions spatiales décroissent (chiffres noirs au dessus des sorties des couches de convolution. Dans ce type d'architecture, les sorties des premiers blocs sont concaténées aux sorties des derniers blocs des court-circuits (skip connections) qui permettent entre autre de conserver l'information sur les structures fines de l'image d'entrée [59].

Les réseaux de neurones usuels peuvent contenir plus d'une centaine de couches et jusqu'à plusieurs millions de poids (de l'ordre de 10 – 100 M pour les architectures ResNet et VGG – voir tableau 1.1). Ces poids sont répartis dans des couches de convolutions et des couches complètement connectées, le nombre total de couches définissant la profondeur du réseau. Le ResNet 18 possède ainsi 17 couches de convolution et une couche de sortie complètement connectée. Le plus grand nombre de

21. residuals networks

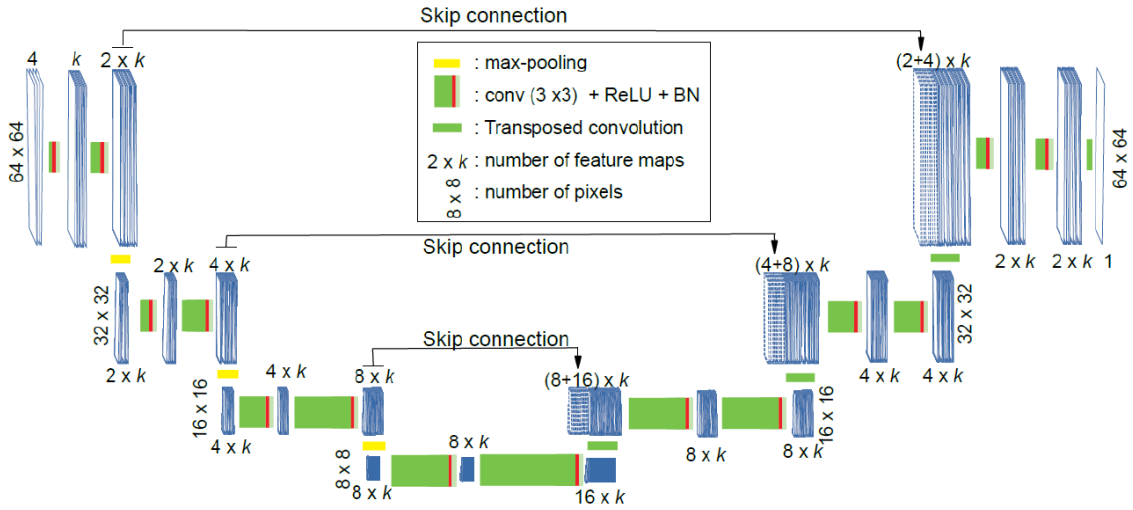


FIGURE 1.3 – Exemple de réseau complètement convolutif (full convolutional network). Ce U-net est utilisé sur des problèmes de segmentation sémantique, et de régression par pixel et de débruitage.

poids dans le ResNet50 vient à la fois d'une profondeur plus grande et d'une répartition différente du nombre de neurones par couches.

architecture	version	#Conv2d 3(7)x3(7)/1x1 #Linear	Nb de paramètres entraînables (cas d'une classification binaire)	Pré-entraînement disponible
ResNet	ResNet18	Conv: 17/0 Lin.:1	11.2 M	Imagenet Places365
	ResNet50	17/32 1	23.5 M	
	ResNet101	34/66 1	44,5 M	
	ResNet152	51/100 1	60,2 M	
ResNext	ResNext50	17/32 1	25 M	
	ResNext101	34/66 1	89 M	
VGG	VGG11	9/0 3	128.8 M	
	VGG13	11/0 3	129 M	
	VGG16	13/0 3	134.2 M	Imagenet
Densenet	densenet162	79/82 1	27 M	Imagenet, Places365
Unet		18/0 0	14.8 M	Cityscape , Mapillary

TABLE 1.1 – Taille des Réseaux utilisés pendant la thèse. Pour chaque architecture, on indique la catégorie (première colonne), la version (colonne 2). Dans les réseaux à résidus, des étapes d'agrégation peuvent être réalisées par des couches de convolutions accessoires associées à des noyaux de dimensions spatiales réduites à 1×1. Dans les cases de la colonne 3, on indique le nombre de couches de convolution classiques, le nombre de convolutions accessoires, et le nombre de couches complètement connectées. Le nombre total de paramètres est indiqué en colonne 4. Dans la colonne 5, on indique les jeux de données sur lesquels des réseaux pré-entraînés étaient disponibles en 2018.

Le profondeur des réseaux et le grand nombre de neurones par couche impliquent une capacité à encoder une classe de fonctions très vaste. Mais pouvoir entraîner ces modèles constitue un triple défi en termes de convergence, de temps de calcul et de risque de surapprentissage. Les méthodes d'entraînement, de régularisation et la question de la taille du jeu d'apprentissage seront abordées dans les parties suivantes.

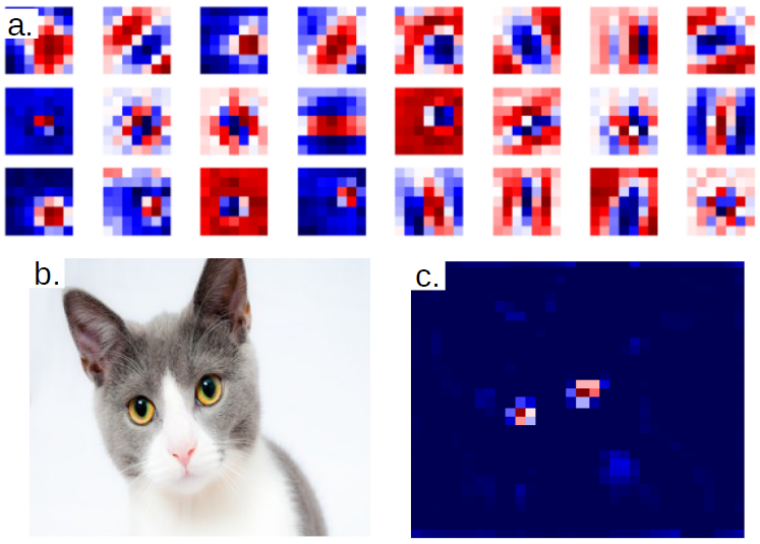


FIGURE 1.4 – a.noyaux de convolution (premier canal) de la première couche d'un ResNet50 pré-entraîné sur Imagenet. Les couleurs rouges (resp. bleues) correspondent aux poids positifs (resp. négatifs). b. image placée en entrée du même réseau. c.sortie d'un des neurones de la couche L_{10} avant application de la fonction ReLU. Les pixels rouges correspondent à des valeurs positives.

Notons pour finir que la profondeur de ces modèles les rend difficiles à analyser. Ce côté « boîte noire » peut d'ailleurs être considéré comme une faiblesse de l'approche. La figure 1.4 montre pourtant qu'on peut parvenir à interpréter le rôle de certains des neurones après entraînement. Après apprentissage d'un ResNet50 sur Imagenet, les noyaux de convolution de taille 7×7 de la première couche ont convergé, pour partie, vers des profils en « ondelette ». Les ondelettes sont particulièrement adaptées à la détection de contours et à la construction d'une représentation parcimonieuse de l'image (compression).
Les cartes de caractéristiques peuvent aussi permettre d'interpréter le rôle joué par un neurone des couches profondes. La figure 1.4 montre le champ $\boldsymbol{y}^k$ (voir équation 1.4) d'un des neurones de la couche L_{10} du ResNet50. Le neurone répond à la présence d'yeux.

1.2.2.2 Entraîner un réseau de neurones profond

Pour un apprentissage supervisé standard, une procédure d'entraînement relativement simple s'est progressivement imposée. Les deux éléments les plus importants sont d'une part la méthode d'optimisation, d'autre part l'initialisation des poids. Un certain nombre de dispositions supplémentaires sont prises pour favoriser les performances en généralisation.

Optimisation :
La fonction de coût $\mathcal{L}$ étant donnée, la quantité qu'on cherche à minimiser pendant l'entraînement s'écrit :

$$\mathbb{E}_X(\mathcal{L}(f(X;w);Y) \tag{1.5}$$

Cette minimisation est abordée par une méthode de descente de gradient. A chaque étape, les gradients de la quantité (1.5) sont évalués pour mettre à jour les poids du réseau.
Il n'est pas pratique d'évaluer cette quantité sur l'intégralité du jeu d'entraînement. Elle est donc évaluée sur des sous-ensembles d'images tirées au hasard dans le jeu d'entraînement (« mini-lots [22] »).
Ces mini-lots comptent généralement d'une dizaine à plus d'une centaine d'images. Il existe de nombreuses règles de mise à jour des poids à partir de gradients calculés sur des mini-lots. La plus simple de ces méthodes consiste à mettre à jour le j^{eme} poids w_j^n par :

$$w_j^{n+1} = w_j^n - \eta \times \frac{\partial \overline{L(f(x_i, w), y_i)}^b}{\partial w_j} \tag{1.6}$$

où η est le taux d'apprentissage [23] et la notation $\overline{Q(x_i, y_i)}^b$ représente la moyenne empirique d'une fonction $Q(.)$ sur les couples (x_i, y_i) d'un mini-lot b.
Pour évaluer les dérivées $\frac{\partial L(f(x_i, w), y_i)}{\partial w_j}$, on applique les règles de dérivation des fonctions composées [24]. En pratique, pour les appliquer efficacement, chacun des champs intermédiaires ainsi que l'ensemble des opérations appliquées à travers le réseau (chaîne de calcul [25]) sont gardés en mémoire. Les gradients sont calculés en remontant la chaîne de calcul. Les produits matriciels correspondants sont parallélisés sur les cartes graphiques. Avec notre matériel [26], l'utilisation de cartes graphiques divise le temps d'apprentissage par facteur compris entre 20 à 50.

L'apprentissage consiste à parcourir l'ensemble des données par mini-lots successifs. Lorsque les mini-lots sont tirés sans remise, une « époque » correspond à un parcours complet du jeu d'entraînement (phase d'entraînement) et du jeu de validation (phase de validation). Les apprentissages sur les très gros jeux de données (> 1M) peuvent s'échelonner sur plusieurs semaines. Ce temps d'apprentissage est une contrainte importante de ces méthodes. C'est pour cette raison que les étapes de sélection de modèle sont réduites.

Initialisation

Les bonnes pratiques en matière d'initialisation des poids ont joué un rôle important dans le développement du deep learning [60]. Lorsqu'ils ne sont pas issus d'un modèle pré-entraîné sur une autre tâche, les poids sont initialisés aléatoirement suivant des lois normales. Les paramètres de ces lois sont choisis de telle façon que les amplitudes des caractéristiques [60] (et des gradients [61]) intermédiaires sont conservées à travers la partie affine des couches de neurones, de façon à éviter les divergences.
Souvent, le problème n'est pas complètement neuf, dans le sens où d'autres problèmes ont été définis

22. Traduction de l'anglais « mini-batch ».
23. learning rate
24. chain rule
25. computational graph
26. Des processeurs intel XEON associés à des cartes GPU Nvidia geforce RTX 2080

et abordés sur le même domaine d'images. Dans ce cas, on peut initialiser les poids sur un réseau déjà entraîné.

1.2.2.3 Eviter le surapprentissage

Augmentation de données :
Prévenir le surapprentissage est un souci majeur du praticien. En théorie, plus le nombre de poids est grand comparé aux données d'entrées et plus le risque de surapprentissage est élevé. Parmi l'ensemble des techniques qui limitent ce risque, l'augmentation de données est la plus spécifique à l'apprentissage profond. Le principe est simple : combiner des transformations t qui laissent l'espace des images et l'annotation invariants ($L_{t(X)} \equiv L_X$ et $L_{Y|t(X)} \equiv L_{Y|X}$). Pour des images d'objets à classer, la réflexion par rapport à un axe vertical, un rognage (modéré) des bords de l'images vérifient généralement cette propriété.
Ces transformations sont appliquées sur chaque image du jeu d'entraînement avec une part d'aléa. Ainsi, les couples entrées - cibles vus par le modèle s'écrivent sous la forme :

$$\left({t^1}_{\alpha_1}({t^2}_{\alpha_2}(...{t^k}_{\alpha_k}(x_i)...)) \, , \, y_i \right) \tag{1.7}$$

où les α_k représentent les paramètres des transformations t^k, tirés aléatoirement. Avec des transformations suffisamment variées et un jeu initial assez grand, les effets du surapprentissage, c'est à dire des performances sur le jeu d'entraînement qui continuent d'augmenter alors que les performances en validation décroissent, ne sont pas observés. On observe plutôt un palier en validation. L'apprentissage peut être stoppé lorsque ce palier est atteint.

Régularisation : Deux autres opérations qui ont pour effet de prévenir le surapprentissage sont communément utilisées : les opérations de *drop out* et de *batch normalization*.
Le *drop out* consiste à désactiver des connexions du réseau choisies au hasard. Cette technique permet aussi d'éviter que des parties importantes du réseau restent inutilisées.
La *batch normalization* consiste à renormaliser le signal intermédiaire entre deux couches de convolution consécutives. La moyenne et l'écart-type sont calés sur des paramètres optimisables du modèles. Cette technique, qui favorise un apprentissage stable et rapide [62], peut être vue comme une forme de régularisation [63]. Ces deux opérations ne seront pas détaillées ici. Elles n'en font pas moins partie des briques de bases du deep learning et sont souvent présentées comme des éléments de l'architecture (par exemple, la batch normalization des U-net, notées BN sur la figure 1.3).

1.2.2.4 Une condition d'application incontournable : de "grands jeux" de données

Une condition préalable à un apprentissage supervisé efficace est la mise à disposition d'un jeu de données de taille suffisante. Mais qu'est-ce qu'un jeu de données de taille suffisante ? Pour répondre, précisons ce qu'on entend par efficace.
S'il s'agit de comparer le deep learning aux autres méthodes du machine learning, les bonnes perfor-

mances des réseaux de neurones profonds ne se limitent pas aux jeux de données les plus gros.
Pour des jeux de quelques milliers d'images, on constate déjà une plus-value (voir par exemple, [64] pour la classification et la segmentation [58], [54], pour les attributs relatifs). Cette plus-value s'explique cependant par le fait que les modèles peuvent être pré-entraînés sur les gros jeux de données de référence.
Mais s'il s'agit de d'explorer le potentiel réel de ces méthodes en vue d'automatiser une prédiction aussi pertinente qu'une décision humaine, il s'agit de pouvoir rassembler autant de données que pour les problèmes école du machine learning sur lesquelles les compétences humaines ont été approchées, ou au moins, autant de données que les approches par i.a. sur des problématiques proches de celle qu'on aborde.
A ce propos, compter le nombre d'images ne suffit pas. D'une part, le nombre de labels par image entre aussi en jeu. Par exemple, les jeux de données utilisés pour la segmentation sémantique ne comptaient en 2020 qu'une dizaine de milliers d'images, contre un à cent millions pour la classification. Mais les petits effectifs, en segmentation, sont compensés par le grand nombre de labels par image (un par pixel). On sait d'ailleurs qu'à réduire le nombre de labels par image, les performances se dégradent rapidement [65].
D'autre part, les données rassemblées doivent vérifier certaines propriétés considérées comme essentielles pour de bonnes performances en généralisation : la représentativité, le caractère équilibré et la fiabilité de la correspondance entre la cible et l'image. La représentativité, en particulier, est considérée comme essentielle [41]. En terme d'image, il faut assurer une diversité des points de vue, de l'éclairage et des éléments d'arrière plan. La redondance des images, d'ailleurs très importante dans les données webcam, doit être prise en compte.
En terme de cibles, le jeu doit être équilibré. L'équilibrage est un défi pour la caractérisation de phénomènes rares, généralement peu représentés dans les archives image.
Enfin, la fiabilité de la correspondance entre la cible et l'image doit aussi être prise en compte. En effet, malgré une relative robustesse vis à vis de cibles bruitées et des effets de compensation par le nombre [66], les performances d'un apprentissage supervisé restent sensibles à la qualité des annotations [67].

Au début de ce travail de thèse, nous disposions par exemple de l'ordre de 500.000 images associées à de la mesure fiable. Mais le faible nombre de scènes disponibles (une dizaine de caméras), la grande redondance de ces images, la rareté des événements d'intérêt ne permettaient pas d'espérer des performances en généralisation comparables à celles de l'état de l'art.
Nous avons donc cherché à produire des jeux de données contenant davantage de scènes. Nous avons aussi exploré des stratégies d'apprentissage adaptées aux cas où le jeu à disposition apparaît modeste en comparaison aux jeux de référence du machine learning. Ces stratégies, qui sortent du cadre de l'apprentissage supervisé, ne sont pas propres à la vision par ordinateur. Elles sont présentées dans la partie suivante.

1.2.3 Stratégies d'apprentissage transversales

Les paragraphes précédents permettent de comprendre de quoi un réseau de neurones profond est constitué et comment il est entraîné, étant donné un jeu de données idéal et une fonction de coût adaptée. Mais dans la pratique, le jeu est rarement idéal.
Pour compenser, un grand nombre de stratégies existent. Elles sont spécifiques à des situations (ou « scénarios », voir figure 1.5). En particulier, il est possible de tirer parti de données complémentaires (transfer learning), de cibles accessoires (multitask learning) ou d'images non-annotées (semi-supervised learning).
Dans d'autres situations, les labels disponibles peuvent comporter des défauts (weak supervision). Ils peuvent être bruités (inacurrate supervision) ou ne pas correspondre au niveau de traitement attendu, par exemple, être relatif à l'image entière alors qu'on attend des prédictions par pixel (inexact supervision). Là encore, des stratégies particulières sont disponibles. Dans les paragraphes qui suivent, nous en faisons une présentation ciblée. Cette présentation sera l'occasion d'introduire les principaux outils conceptuels de ce travail de thèse.

1.2.3.1 Apprentissage par transfert

Le concept d'apprentissage par transfert englobe une première partie des scénarios. Pan et Yang [68] le définissent formellement par le fait d'améliorer les scores sur une tâche cible, définie sur un domaine cible $\mathcal{D}^c = \mathcal{X}^c \times \mathcal{Y}^c$, à partir d'une tâche source définie sur un domaine source $\mathcal{D}^s = \mathcal{X}^s \times \mathcal{Y}^s$. Cet espoir d'amélioration repose sur la ressemblance entre les entrées ($\mathcal{X}^c = \mathcal{X}^s$) ou sur la ressemblance entre les tâches (e.g. $\mathcal{Y}^c = \mathcal{Y}^s$ dans le cas d'une classification).
Le premier scénario a déjà été évoqué, et constitue la voie la plus courante et aussi la plus simple à mettre en œuvre. Dans ce scénario, on dispose d'un modèle source ayant appris sur un autre échantillon d'images du domaine source. Le plus souvent, il s'agit d'une tâche de classification multiclasse sur de grands jeux de données publiques (Imagenet, Places365).
Dans ce scénario, le modèle source peut simplement être ré-entraîné sur la tâche cible. On espère que les caractéristiques extraites par le modèle source permettront d'aborder la tâche cible plus efficacement. Avec des réseaux de neurones profonds, cela revient à initialiser les premières couches du modèles avec les poids du modèle source.
Cette technique, nommée pré-apprentissage [27] (ou pré-entraînement), permet d'obtenir de bonnes performances (en comparaison aux autres méthodes d'apprentissage) sur de petits jeux de données [69].

Dans un autre scénario, des labels supplémentaires permettent de définir des tâches accessoires. L'apprentissage peut être réalisé simultanément sur plusieurs tâches. L'idée est de faciliter l'extraction de caractéristiques utiles à l'ensemble des tâches abordées [70]. En termes de performance, la plus-value (sur la tâche d'intérêt) est n'est pas systématique [71], elle peut même s'avérer contre-productive [70].

27. fine-tuning

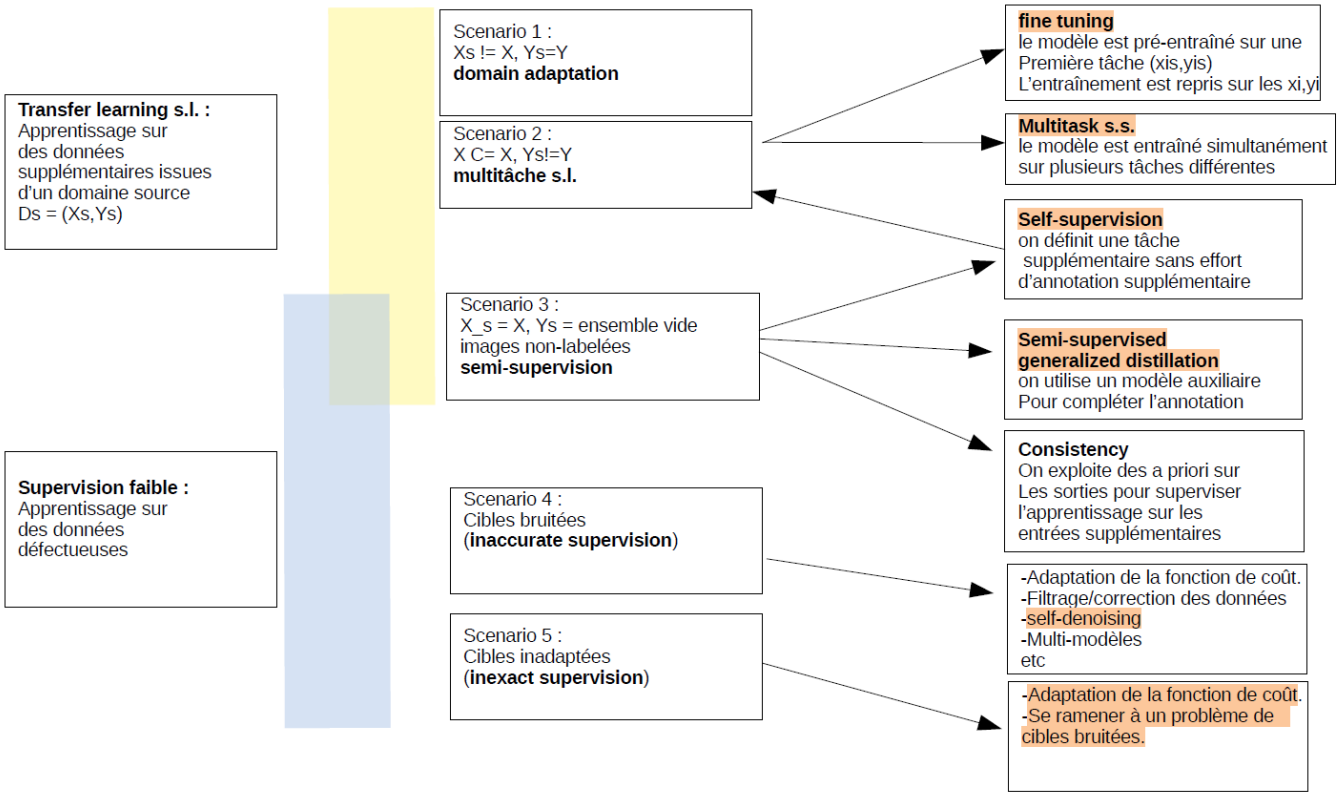


FIGURE 1.5 – Différentes stratégies d'apprentissages applicables dans le cas où des données supplémentaires sont disponibles (transfer learning), ou lorsque les données à disposition sont défectueuses (supervision faible). Les items surlignés en orange couvrent des techniques testées pendant la thèse.

L'apprentissage multi-tâches des réseaux de neurones profonds fait l'objet d'une littérature conséquente [72], en particulier sur la combinaison des fonctions de coût et sur des techniques d'optimisation spécifiques [73]. Du côté des architectures, les couches de sortie des réseaux peuvent être modifiées de nombreuses façons. Par exemple, l'encodeur du U-net peut être associé à plusieurs décodeurs, chacun associé à une tâche particulière [74].

Dans un troisième scénario, un grand nombre d'images non-annotées sont disponibles. A cette situation correspondent les approches semi-supervisées. Il en existe une grande variété, qu'on a découpé ici en trois catégories, non exhaustives.

Dans la première catégorie, on exploite les hypothèses supplémentaires sur le paramètre ciblé pour pénaliser les prédictions sur les images supplémentaires. Par exemple, la lente variation du paramètre ciblé peut conduire à pénaliser les écarts de prédictions sur les images consécutives d'une série temporelle (voir par exemple [21]). Un autre exemple concerne l'apprentissage de préférences. Pour une tâche de prédiction par paires, il est possible de pénaliser les écarts à la transitivité sur les images supplémentaires.

Dans la deuxième catégorie, on définit une tâche accessoire sur l'ensemble des images disponibles. Cette tâche ne doit rien coûter en terme d'annotation -on parle d'auto-supervision (self-supervision). Par exemple, il peut s'agir de tâche de reconstitution de l'image. Dans un premier exemple, le réseau est entraîné à reformer l'image à partir d'une information incomplète ou bruitée [75],[76],[77],[78]. Dans un second exemple, les images d'une séquence temporelle sont présentés dans un ordre aléaoire, et le réseau doit retrouver l'ordre initial [79]. Les avantages liés à cette tâche supplémentaire sont tirés par apprentissage multitâche ou par pré-entraînement.

Enfin, dans une troisième catégorie, on se place dans le cas où les images supplémentaires comme celles du jeu d'entraînement initial sont associées à des données complémentaires [28], dont on ne disposera pas pendant la phase de test [80]. Un modèle accessoire prenant en compte ces données complémentaires est appris sur le jeu d'entraînement initial et appliqué aux images supplémentaires pour étendre l'annotation. Le modèle définitif peut être entraîné sur le jeu ainsi étendu. C'est un cas de distillation généralisée [81].

Un dernier scénario couvre le cas où les domaines des entrées sont distincts mais où les cibles sont de même nature. On parle alors d'adaptation de domaine. Ces techniques ne seront pas illustrées dans cette thèse.

1.2.3.2 Cas de données bruitées ou inadaptées : la supervision faible

Le concept de supervision faible (weak-superivsion) regroupe des scénarios d'apprentissage avec des données cibles défectueuses [82]. Dans le cas où seule une partie des cibles est fournie, on retombe sur la notion d'apprentissage semi-supervisé, partagée avec l'apprentissage par transfert.

Un autre scenario fréquent est l'apprentissage avec des cibles bruitées. Dans la littérature, ce scénario est nommé supervision imprécise [29]. Dans ce scénario, la cible $\tilde{y}$ est une version bruitée de la cible y. L'effet sur l'apprentissage dépend de la méthode utilisée[83].

On montre par exemple que [67] les architectures à couche de convolutions présentées en section 1.2.2.1 et entraînées sur une tâche de classification plus robustes que des réseaux de neurones de plus petite taille face à un bruit de cible de type permutation aléatoire.

Pour renforcer la robustesse face au bruit de cible, un grand nombre de méthodes existent. Les unes sont fondées sur le filtrage progressif du jeu de données [84], [85], d'autres sur une modification de la fonction de coût et des dernières couches du réseau [86].

Enfin, moins fréquemment, il arrive que les cibles disponibles soient inadaptées. Par exemple, lorsqu'on cherche une prédiction à l'échelle du pixel à partir d'annotations à l'échelle de l'image. Dans la littérature, le terme de supervision inexacte [30] est employé [82] pour désigner ces situations.

28. traduction personelle de « privilidged data »
29. inacurrate supervision
30. « inexact supervision »

Les frontières entre ces catégories ne sont pas tranchées. Par exemple, lorsque seule une part des données est bruitée, on entre dans un cas hybride entre semi-supervision et supervision imprécise (voir par exemple [87]).
Il arrive aussi qu'un problème de supervision inexacte soit abordé comme un problème de supervision imprécise. Ce type de détour a été à l'origine de deux des méthodes proposées dans cette thèse (voir Chapitre 4 et annexe E).

1.2.4 Conclusion

Dans cette section, nous avons présenté quelques-uns des principaux problèmes d'apprentissage en vision par ordinateur. Ces problèmes relativement faciles pour l'humain sont longtemps restés hors de portée de la machine. Les performances récentes (à partir de 2012) des réseaux de neurones profonds ont changé la donne. Ces modèles, comportant plusieurs millions de paramètres et appris de bout en bout sur de grands jeux de données, permettent d'atteindre d'excellentes performances.
Mais l'apprentissage de ces modèles est délicat. De grands jeux de données combinés à des techniques d'augmentation de données et de régularisation efficaces sont nécessaires pour éviter le surapprentissage. Il a aussi fallu développer un matériel informatique et des techniques d'entraînement adaptées à de grands jeux de données de façon à limiter le temps d'apprentissage et à maîtriser la convergence du modèle vers des solutions intéressantes.
Cet environnement spécifique s'est considérablement développé depuis 2012. Six ans plus tard, la procédure d'apprentissage basique est relativement facile à mettre en oeuvre : les procédures d'entraînement sont standardisées, le matériel informatique est facile d'accès et de nombreux tutoriels existent. Cependant, les choix d'une architecture, d'une méthode d'optimisation, d'initialisation des poids, de formation des mini-lots, d'augmentation de la donnée, font chacun apparaître des hyperparamètres dont le réglage est délicat.
A cette compléxité s'ajoutent celle des stratégies d'apprentissage, à notre avis incontournables dans le cas où les jeux de données sont de taille modeste.

1.3 Cadre de la thèse et organisation du manuscrit

Dans cette troisième partie, nous précisons les objectifs et le cadre du travail. Nous présentons ensuite le plan du manuscrit.

1.3.1 Cadre de la thèse

Ce travail de thèse est un travail exploratoire. Il s'est agi de tester et d'adapter des méthodes du machine learning pour extraire de l'information météorologique à partir de données webcam d'opportunité.
Les deux principaux phénomènes ciblés sont l'enneigement des sols et les variations de la visibilité. L'enneigement, parce que le suivi des épisodes de neige en plaine est le premier besoin exprimé par l'organisme qui finance la thèse (Météo-France). Les informations potentiellement utiles aux prévisionnistes sont précises : il s'agit de dire quand la neige commence à tenir au sol, en dehors ou sur les routes, de donner la tendance, fonte ou accumulation et de pouvoir estimer aussi précisément que possible l'épaisseur du manteau.
Les images webcam, et en particulier les webcams routières permettent en effet de répondre à ces besoins. Les tendances à la fonte ou à l'accumulation apparaissent à travers les variations de l'étendue du manteau neigeux. L'épaisseur du manteau neigeux, plus difficile à évaluer, transparaît surtout des relations entre la couche de neige et les objets de la scène.
Nous avons considéré la visibilité horizontale [31] comme un autre paramètre d'intérêt. D'abord parce qu'en matière d'estimation d'après l'image, ce paramètre a déjà fait l'objet de nombreuses recherches. Il était donc plus simple d'évaluer les améliorations apportées par la recherche méthodologique entreprise. Ensuite, parce que nous espérions profiter d'une caractérisation simultanée des deux phénomènes à travers une approche multitâches. Enfin, parce qu'une estimation de la visibilité peut déboucher sur une estimation plus ou moins fine du taux de précipitations neigeuses, et donc aider à caractériser un épisode de neige en plaine [88].

Nous nous sommes concentrés sur le traitement d'images venant de webcam fixes, d'extérieur, produisant des images en couleur à intervalle régulier.
Comme la surveillance des routes constitue une application importante, les scènes routières [32] ont concentré les efforts d'annotation. De plus, si des images prises de nuit ont été échantillonnées et pour partie annotées, la plupart des méthodes sont évaluées sur les images de jour.
Malgré ces deux restrictions, les caméras et les scènes sur lesquelles le travail a été conduit sont très variées.

31. La visibilité, exprimée en mètres, est la distance maximale à laquelle un objet peut être distingué. Elle sera définie plus précisément au chapitre 2.
32. C'est à dire les scènes sur lesquelles apparaît une rue ou une route.

Pour pouvoir extraire une information à partir de scènes et de caméras très variées, l'accent a été mis sur les performances en généralisation. Les développement méthodologiques entrepris ont, pour la plupart, visé à améliorer ces performances. Cependant, il n'a pas été question de développer des architectures spécifiques. Les réseaux de neurones utilisés dérivent tous d'architectures pré-existantes qui ont fait leur preuves sur les grands problèmes du machine learning. De même, les aspects big data et calcul haute performance sont restés en arrière-plan et ne seront évoqués que très rarement dans ce mémoire. Par contre, des tâches d'apprentissage originales ont été imaginées et plusieurs stratégies d'apprentissage transversales ont été développées.

Enfin, il nous a semblé utile de prédire une indication sur la précision de l'information extraite. Par mauvais temps, la qualité des images webcam est en effet très variable. En particulier, les gouttelettes et les flocons qui collent à la lentille externe de la caméra ou à la vitre de protection sont fréquents et peuvent réduire l'information sur plusieurs images d'affilée.
D'autre part, toutes les caméras n'offrent pas la même qualité d'information. Par exemple, une vue en plongée ne dit que peu de choses sur la visibilité. Sans une surface verticale qui servirait de jauge, il est difficile d'évaluer l'épaisseur du manteau. Plus généralement, la précision de l'information disponible dépend de la scène observée, de l'orientation et de la qualité de la caméra et de l'image. A cet égard, travailler avec des cibles de résolution fixée comme le sont les classes d'intensité ne nous a pas paru complètement satisfaisant.

1.3.2 Organisation du manuscrit

Le premier problème qui s'est posé est celui de la construction de jeux d'apprentissage. Le choix d'une méthode de construction était décisif : il contraignait le choix des méthodes d'apprentissage et celui des méthodes d'évaluation. Toutes les autres problématiques abordées ont découlé de cette orientation. Ces problématiques sont illustrées sur l'estimation de la visibilité et/ou la caractérisation du manteau neigeux. Le chapitrage reprend grosso-modo l'ordre chronologique dans lequel elles ont été abordées.

1.3.2.1 Chapitre 2

Ce chapitre présente les étapes de construction de nos jeux d'apprentissage initiaux. Trois contraintes sont prises en compte.
Premièrement, il faut des jeux de données de taille suffisante pour généraliser sur des caméras quelconques.
Deuxièmement les cibles doivent être définies avec un niveau de précision suffisant pour répondre aux enjeux de la météorologie opérationnelle. Il faut par exemple pouvoir décider si la visibilité est passée sous un certain seuil ou prédire la croissance du manteau neigeux. En particulier, il ne faut pas se li-

miter à des problèmes de classification binaire comme « beau temps »/« brouillard » ou « neige »/« non neige ». De plus, la donnée collectée doit permettre de caractériser la précision de l'information disponible dans l'image.
Troisièmement, il faut anticiper l'utilisation d'approches par transfer learning ou par supervision faible (voir section 1.2.3).

Pour répondre à ces contraintes, nous avons privilégié une annotation manuelle. Des cibles ont été définies pour un problème de classification multi-classe et pour un problème d'estimation relative des paramètres visibilité, étendue de la surface enneigée et hauteur de neige.
Dans un souci de précision et d'homogénéïté, l'annotation n'a pas été réalisée par crowdsourcing. Pour compenser, d'importants efforts ont été faits pour accélérer l'annotation. En particulier, une méthode d'annotation semi-automatique a été développée pour faciliter la comparaison de paires d'images. Cette méthode a facilité la production d'un grand nombre de comparaisons expertisées (plus d'une centaine de milliers par paramètre). Ces comparaisons sont compatibles avec une relation d'ordre partiel. C'est la relation d'incomparabilité associée à cet ordre partiel qui reflète la qualité de l'information contenue dans une image.

1.3.2.2 Chapitre 3

Ce chapitre présente des résultats de base sur les différentes tâches qu'on peut définir à partir du jeux de données : classification et apprentissage des préférence. La partie A est consacrée aux problèmes de classification. La partie B est consacrée à l'apprentissage de préférences sur les comparaisons expertisées relatives à la **visibilité**. Sur ce jeu de comparaisons, nous vérifions l'intérêt de la méthode d'annotation semi-automatique présentée au chapitre précédent.
Les deux modalités d'apprentissage évoquées dans la section 1.2.1.5, prédiction par paire et prédiction par image, sont ensuite implémentées sur des réseaux à couches de convolution standards et comparées aux méthodes de l'état de l'art (ranking SVM [89]).
Ces résultats nous conduisent à proposer une première application à la détection de dépassement de seuil.

1.3.2.3 Chapitre 4

Ce chapitre est consacré à l'amélioration et à l'étalonnage des fonctions de rang (prédiction par image), toujours sur le critère de la visibilité. Pour améliorer les performances en généralisation, une méthode d'apprentissage semi-supervisé est proposée.
Pour rendre compte de la précision de l'information contenu dans l'image, nous introduisons les « fonctions de rang bivaluées », mieux susceptible de restituer l'ordre partiel contenu dans l'annotation.
Dans la suite du chapitre, nous explorons le thème de l'étalonnage des fonctions de rang.
Dans cette partie plus spéculative, nous donnons d'abord du sens à l'étalonnage des fonctions de rang

bivaluées. Nous envisageons ensuite deux méthodes d'étalonnage à partir de données issues de capteurs distants. Ces méthodes sont plus spécifiques à la visibilité. La deuxième est basée sur un nouvel apprentissage faiblement supervisé.

1.3.2.4 Chapitre 5

Sur les deux autres paramètres d'intérêt, l'étendue et la hauteur du manteau neigeux, les premiers résultats se sont d'abord avérés décevants par certains aspects. Une dernière étape d'annotation, en partie basée sur les résultats du chapitre 4, a permis d'étendre les jeux de données relatif à l'étendue.
Pour ce paramètre, nous cherchons de nouveau à améliorer les performances des fonctions de rang. Les méthodes développées au chapitre 4 sont adaptées.
Pour faire le bilan des progrès accomplis, nous comparons l'approche par fonctions de rang à l'approche par classification (chapitre 3) sur le problème de la détection de la neige au sol.
Enfin, une application pratique (détection d'un début d'enneigement, ou de la reprise de l'enneigement) est considérée.

1.4 Utilisation des annexes

Pour faciliter la lecture du manuscrit, nous avons mis en annexe les éléments les moins importants pour la compréhension. Les annexes A, B et C contiennent des éléments transversaux, auxquels on fait référence tout au long du manuscrit. Ce sont des détails sur les règles d'annotation (Annexe A), sur les jeux d'apprentissage (Annexe B) et sur la nomenclature des modèles utilisés (Annexe C).
Les annexes D et E contiennent des compléments sur les méthodes et les jeux d'apprentissages utilisées dans les chapitres 4 et 5. Les deux dernières annexes contiennent les productions scientifiques (Annexe F) et numériques (Annexe G).
L'annexe F contient les publications avec acte : un papier court pour la participation à la conférence Climate Informatics, qui s'est tenue à Paris en 2020 qui décrit la méthode d'annotation et une partie des résultats du chapitre 3, un papier long sur les résultats du chapitre 4 en cours de publication. Elle contient aussi la principale publication relatives à mes travaux sur les données radar, dans le prolongement de mon stage de fin d'études. Cet article reste en lien avec le contenu de cette thèse dans la mesure où il s'agit d'une approche faiblement supervisée par deep learning. Il s'agit d'un article long publié dans la revue Transactions on Geoscience and Remote Sensing.

L'annexe G, est mise à disposition sur le github du département SPACE [33]. Elle contient :

- Le logiciel LabelGO et les codes de la deuxième étape de la méthode d'annotation semi-automatique.
- Les codes d'apprentissage des fonctions d'ordre, ceux des approches faiblement-supervisées évoquées dans les chapitres 4 et 5

33. https ://github.com/space-latmos/

— Les graphes au format .gpickle contenant les annotations relatives aux jeu AMOS. Les images associées peuvent être récupérées auprès de Nathan Jacobs [34].
— Des séries de prédictions brutes obtenues à partir de webcam françaises.

34. jacobs@cs.uky.edu

Chapitre 2

Collection des images et annotation

Ce chapitre présente les étapes de construction de nos jeux d'apprentissage initiaux. Nous avons abordé cette construction en deux temps : la collection des images (section 2.1) et l'annotation (section 2.2).
Nous avons vu au premeir chapitre qu'un jeu d'apprentissage doit contenir un échantillon large, représentatif et non redondant du domaine d'intérêt pour obtenir des performances en généralisation intéressantes.
Pour obtenir des performances en généralisation sur des images webcams, il faut donc rassembler des image prises avec des matériels variés, sous des angles et sur des scènes variées et dans diverses conditions d'éclairage. Il faut aussi des images représentatives des situations météorologiques qu'on cherche à distinguer les unes des autres. Pour une caractérisation fine des événements de neige, il faut pouvoir trouver des images associées à des degrés d'enneigement et à des visibilités horizontales qui couvrent toute la gamme des possibles, et surtout pendant les périodes de croissance du manteau. Pour pouvoir caractériser la précision de l'information contenue dans l'image, il nous paraît aussi important d'échantilloner des « bruits » de nature et d'intensité variées.
Pour obtenir une grande diversité de matériels et de scènes, nous avons multiplié les sources d'images webcams utilisées. La principale source est la base d'archives AMOS [17]. Pour récupérer des images d'AMOS qui offrent une bonne couverture des situations d'intérêt, des modalités d'extraction spécifiques sont mises en oeuvre. Le processus de sélection des images comporte quatre étapes qui débouchent sur plusieurs jeux d'images : un jeu destiné à l'annotation, et un jeu beaucoup plus vaste, destiné à la mise en œuvre de stratégies d'apprentissage faiblement supervisées.
Pour compléter ces jeux, des webcams françaises ont été échantillonnées par nos soins pendant les épisodes de neige des hivers 2018-2019 et 2020-2021. Nous présenterons aussi les archives de caméras installées en station météorologique qui ont été mises à notre disposition en début de thèse (archives TENEBRE). Portant sur une dizaine de scènes, elles seront principalement utilisées à des fins d'évaluation.

L'annotation des images représentait un second défi. Nous avons exclu la piste d'une annotation via des mesures distantes ou par la télédétection. En effet, il ne nous paraissait pas possible de rassembler

de cette manière des cibles d'apprentissage fiables, c'est à dire représentatives de la scène observée localement par les caméras [1]. Or, pour évaluer et comparer des approches par apprentissage, il vaut mieux disposer d'un jeu de test sûr, lui même de grande taille. En outre, il nous paraissait difficile d'apprendre sur des cibles imprécises à évaluer la qualité de l'information disponible dans l'image. Enfin, l'étendue apparente du manteau neigeux, est relative à la scène (donc à la caméra) considérée. Ce n'est pas une grandeur physique à laquelle nous aurions accès par la mesure.
Nous avons donc privilégié une annotation manuelle.
Nous avons commencé par définir des problèmes de classification sur les images collectées. Mais pour pouvoir caractériser le plus finement possible les phénomènes d'intérêt, nous nous sommes progressivement tournés vers l'annotation par paires, avec l'idée de construire des jeux assez grands pour entraîner efficacement des réseaux de neurones profonds.

1. Pour limiter l'erreur, il aurait d'abord fallu chercher à localiser précisément les webcams d'AMOS utilisées, tâche a priori très fastidieuse [36], ou utiliser un procédé de correction [18], dont l'efficacité ne nous semblait pas encore démontrée

2.1 Collection des images

Quatre sources ont été exploitées pour constituer les jeux de données : la base d'images webcam AMOS [17], ouverte à la recherche, les images webcam issues d'un réseau de caméras de la Direction Interrégionale des Routes (DIR), les images des webcams du réseau Infoclimat et la base TENEBRE (proprétaire Météo France). Leurs principales caractéristiques sont précisées dans la table 2.1.
Dans cette partie, nous décrivons ces sources et les principales étapes d'extraction d'extraction et de sélection de séquences des "séquences" d'images, résumées dans la figure 2.1.
Précisons d'emblée ce que nous entendons par "séquence d'images". Dans ce manuscrit, ce terme désignera un ensemble d'images venant d'une même caméra fixe. Par fixe, on entend une caméra dont l'orientation varie peu au cours du temps, sous les seuls effets du vent et de la température. Ce terme de séquence n'implique aucune régularité temporelle, contrairement à celui de « **série** d'images ».

Sources d'images	Nombre de caméras	Scènes routières	Nombre d'images	Période	Pas de temps	Localisation
AMOS	10 K (5280*)	10 K (2615*)	2.8 M	2002-2017	88%=30 min 11%≥ 1h	Mondiale (> 35° N)
DIRs	97	97	190 K 5M	du 21/01/19 au 23/01/19 et déc. 2020 jan. 2021	87%<5 min 100%<15 min	France
Infoclimat	102	12	36 K 2.5M		43%<5 min 61%<30 min	
TENEBRE	10	2	550 K	aut.-hiv. 2012 aut.-hiv. 2017	10 min	

TABLE 2.1 – Description des sources d'images utilisées dans la thèse.(*) Le nombre exact de caméras est indicatif. Il n'a pu être évalué qu'au bout de la troisième étape d'annotation (voir section 2.1.1). Le nombre entre parenthèses, indiqué au dessous, correspond au nombre de sites internet dont les archives ont été exploitées. Le pas de temps est la durée médiane qui sépare deux images consécutives dans une séries d'images. Une version plus complète de cette table est disponible dans l'annexe B.1.

2.1.1 Exploitation de la source AMOS

La base AMOS contient les archives d'environ 30.000 sites internet alimentés par des images webcams en couleur. C'est de loin la plus large base d'images webcam accessible dans un cadre de recherche. De nombreux travaux touchant à l'estimation des paramètres météorologiques dans l'image

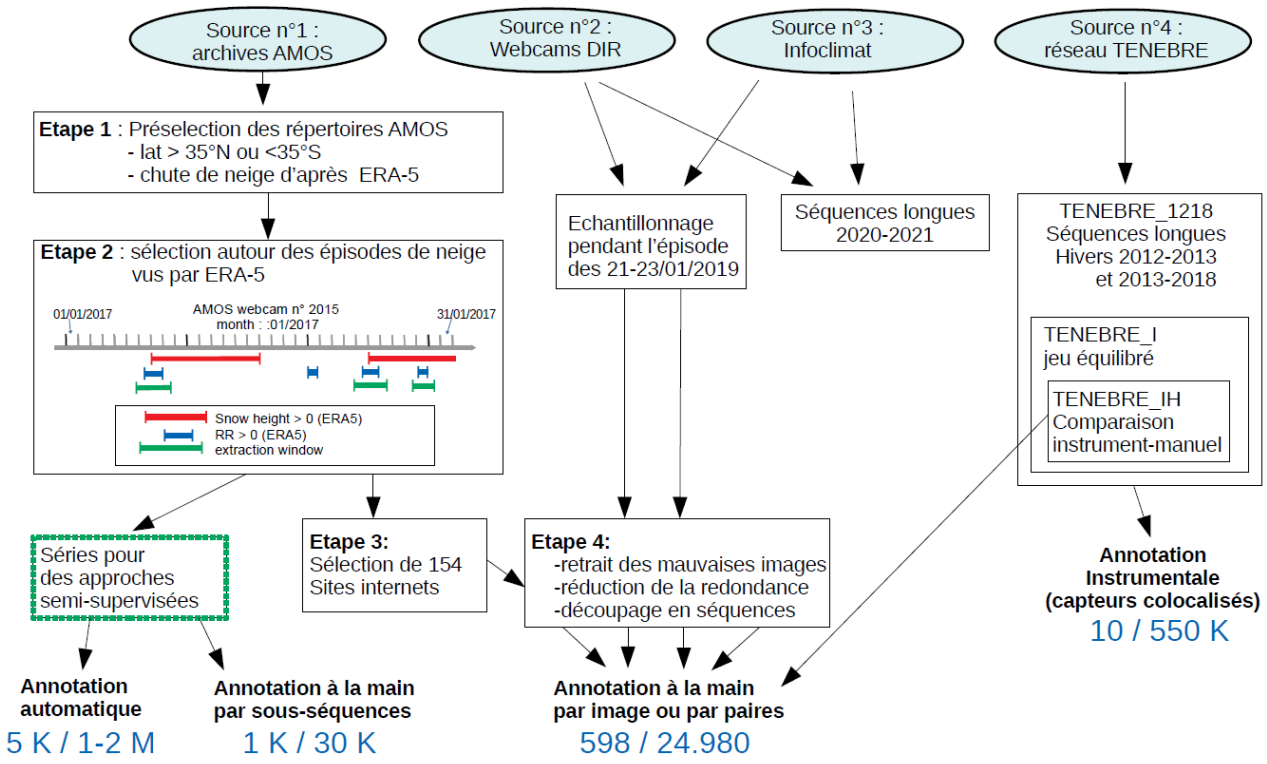


FIGURE 2.1 – Schéma synthétique des étapes d'extraction et de sélection à partir des quatre sources d'images. En bout de chaîne, en bleu, le premier nombre indique le nombre de séquences et le nombre d'images obtenus.

se sont appuyés dessus (voir chapitre 1).
Les scènes archivées dans cette base sont très variées (voir figure 2.2). Les matériels vont de la caméra panoramique haute résolution aux caméras de surveillance bon marché. Les paramètres intrinsèques des caméras et la taille des images sont eux aussi très variés (voir annexe B.1).

La base AMOS est constituée d'un ensemble de répertoires. Chaque répertoire contient les archives mensuelles d'un site internet alimenté par des images prises en extérieur par un appareil généralement fixe. La base contient les archives d'environ 30.000 sites internets.
Le pas de temps entre deux images consécutives varie d'un site à l'autre, mais il est généralement supérieur à la demi-heure (voir table 2.1). Dans un répertoire, les images proviennent souvent de plusieurs caméras utilisées à tour de rôle [2]. Parfois, une seule caméra est utilisée, mais les réglages sont modifiés à distance (zoom, rotation). De plus, au sein d'un même répertoire, les dimensions de l'image archivée, le mode de compression, peuvent changer d'une année à l'autre.

Pour aborder la caractérisation de la neige au sol sur les images de scènes routières, nous avons cherché à extraire des séquences d'images enregistrées pendant et autour des épisodes de chutes de neige. Au sein d'AMOS, ces images sont très peu fréquentes. En France, le cumul des périodes au cours desquelles la neige tombe et s'accumule au sol est de l'ordre d'une journée par an en plaine (moins

2. C'est par exemple le cas de plus de 50 % des sites associés à des scènes routières

FIGURE 2.2 – Exemples de scènes routières tirées de la base AMOS pendant des épisodes de chutes de neige. Les scènes et les caméras (modèle, orientation) sont très variées.

de 500 m d'altitude). Sur le continent américain, où se trouvent la majorité des caméras d'AMOS, ce chiffre peut être supérieur localement, mais il reste négligeable devant le nombre de données en jeu.
Pour trouver et extraire ces images, nous avons cherché les potentiels épisodes de neige par recoupement entre le géo-référencement approximatif des caméras et la réanalyse météorologique globale ERA-5 [90]. Il a ensuite fallu réduire la redondance des séries extraites et rassembler des images en séquences , et donc repérer les fréquents changements de scène au sein des séries d'images extraites. Les paragraphes suivants donnent le détail de ces étapes, aussi illustrées sur la figure 2.1.

2.1.1.1 Etape 1 : croisement avec la réanalyse ERA-5

Par le biais des adresses IP, chaque site internet archivé est associé à des coordonnées géographiques [17]. Les archives associées aux sites géo-référencés dans la bande comprise entre 35° nord et 35° sud ont été écartées.
Nous avons ensuite récupéré la liste des mois disponibles pour chacun des sites restants (15.465 sites). Dans le même temps, nous avons extrait de la réanalyse ERA-5 la hauteur de neige tombée au cours des six dernières heures (« snowfall », SF6h) et l'épaisseur du manteau neigeux (« snowheight », SH6h) autour des caméras d'AMOS.
Les requêtes au Climate Data Store, qui gère les données ERA-5, ont porté sur les quatre points de grille situés autour de chaque couple de coordonnées. Un enneigement moyen est calculé à partir des quatre épaisseurs. Les archives pour lesquelles l'enneigement moyen est resté nul sur toute la période disponible (moins d'une centaine de sites internet) n'ont pas été échantillonnées.

2.1.1.2 Etape 2 : Restriction aux épisodes de chutes de neige

Pour concentrer la collecte sur les séries les plus riches en terme de variété d'enneigement, de visibilité, d'éclairage et de précipitations, nous avons défini les fenêtres temporelles d'extraction autour des épisodes de chutes de neige avec tenue de la neige au sol (figure 2.1, étape 2). Pour ce faire, nous délimitons d'abord des « périodes d'enneigement » à partir des séries d'enneigement moyen. Ce sont les périodes au cours desquels l'enneigement moyen est strictement positif. Chaque période d'enneigement est ensuite découpée en épisodes de chutes de neiges définis autour des maximums locaux des séries de hauteur de neige moyenne. Des marges sont prises de façon à échantillonner aussi des images sans neige et des images avec neige au sol par beau temps.
Au total, 5.280 sites archivés et 8.054 séries d'images ont ainsi été échantillonnées.

2.1.1.3 Etape 3 : Sélection des scènes routières

Pour sélectionner les scènes à annoter (figure 2.1, étape 3), nous avons extrait une image par site. L'image est prise au cours d'une période d'enneigement (suivant la prédiction ERA-5), de jour.
Sur les 5.280 images regardées, il y avait 2.615 scènes routières. Le reste contient des scènes variées (voir annexe B.2).
La neige est visible au sol sur 1/3 des images de scène routières. Elle couvre une partie de la route sur 5% des images. Ces proportions nous ont paru suffisantes pour construire un jeu de données [3]. Dans les séries d'images extraites, la redondance est très importante.

2.1.1.4 Etape 4 : Réduction de la redondance et formation des séquences

Cette étape a été commune à toutes les séries d'images destinées à l'annotation à la main, qu'elles proviennent d'AMOS ou des autres sources. Pour réduire la redondance, les images ont été sélectionnées à la main. La sélection s'est faite selon les règles suivantes :

— Conserver toutes les images consécutives pendant les épisodes de neige où la neige tient sur la sol, de jour ou de nuit.

— En dehors de ces épisodes, échantillonner plus densément les périodes de fonte de neige et les événements au cours desquels la visibilité décroît (pluie, brouillard, pollution, etc).

— En dehors des périodes de précipitations, lorsque le ciel est visible, prêter attention à la nébulosité du ciel et à l'éclairage. En favoriser la diversité. Les phases de transition (aube et crépuscule) sont donc légèrement surreprésentées.

— les images qui ne comportent aucune information sur la météorologie (voir figure 2.3.a-b) sont laissées de côté. Au contraire, les images de mauvaise qualité mais contenant encore de l'infor-

3. Mais loin d'être suffisante pour annoter directement les données.

mation (voir figures 2.3.c-d) nous intéressent. Elles sont donc sur-échantillonnées.

— Conserver enfin toutes les images qui pourraient prêter à confusion. Par exemple, des scènes rurales blanchies par la gelée blanche, les scènes autoroutières où les reflets du soleil peuvent faire apparaître de longues traces blanches sur les routes, etc.

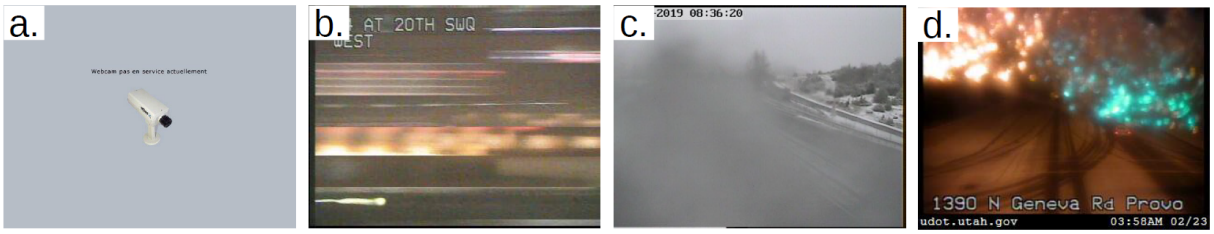


FIGURE 2.3 – a-b. Images ne comprenant aucune information météorologique, rejetées pendant l'élagage manuel. c-d. Pendant les épisodes de précipitations (neige ou pluie), les images de mauvaises qualité sont fréquentes. Ici, la neige collée à la vitre de protection masquent la moitié de l'image. Mais ces images contiennent encore une information relative à la météorologie locale. Elles sont donc conservées.

A l'issue de cette sélection, les changements de scène ont été repérés. De cette façon, les images résultantes ont pu être réparties en séquences. Au final, parmi les 2615 sites internet associés à des scènes routières, seule une fraction des images AMOS ont été traités à l'étape 4. A l'issue de cette étape, 12.475 images réparties en 388 séquences [4] ont été sélectionnées pour être annotées à la main. Par ailleurs, de l'ordre de 5.000 séries d'images AMOS ont été mises de côté pour des apprentissages faiblement supervisés.

2.1.2 Les webcams des DIRs et du réseau Infoclimat

Une limite importante des archives AMOS tient au pas de temps qui sépare deux images consécutives. Ce pas de temps, de l'ordre de la demi-heure, ne permet pas un suivi fin des épisodes de neige (table 2.1).
En comparaison, sur les sites internet mettant à disposition des images webcam, le pas de temps des webcams publiques est plutôt de l'ordre de 5 minutes.

Pour apprendre et surtout évaluer des modèles sur des données plus réalistes, il paraissait intéressant d'archiver quelques épisodes de neige par nos propres moyens. A cette fin, deux campagnes d'archivage ont été menées. La première a eu lieu les 21, 22 et 23 janvier 2019, autour d'un épisode de neige en plaine (5 - 10 cm de neige mesurés sur les départements de l'IDF, de l'est et du centre). La seconde a eu lieu entre novembre 2020 et juin 2021, période au cours de laquelle plusieurs épisodes de neige en plaine ont eu lieu. Les séquences longues (voir figure 2.1) issues de la seconde campagne ont été

4. Ce nombre a varié au cours de la thèse. En racordant les séquences issues de caméras identiques il est passé de 777 à 388.

réservées pour la construction de cas d'études (elles n'ont pas été annotées). Les webcams archivées proviennent de deux réseaux différents, décrits ci-dessous.

2.1.2.1 Webcams des DIRs

Le premier réseau appartient aux Directions Interrégionales des Routes (DIR). 192.282 images venant de 98 caméras [5] donnant sur des scènes routières ont été archivées. Après réduction de la redondance, nous avons restreint ce jeu de données à 8.715 images dont 5.501 images de jour (voir table 2.1).

2.1.2.2 Webcams du réseau infoclimat

Le second réseau de webcams est accessible en ligne sur le site de l'association infoclimat. 102 webcams ont été archivées. Ces caméras portent sur des scènes de natures différentes de celles des DIRs. On compte seulement douze scènes routières. Soixante-quinze d'entre elles sont situées dans des aérodromes. Sur la même période que les caméras des DIRs, 35.880 images ont été archivées. Comparé aux DIRs, il y a moins d'images parce que le pas de temps est généralement plus long. Après réduction de la redondance, il est resté 1.285 images dont une grande majorité d'images de jour (1.124).

2.1.3 Les archives du réseau TENEBRE

Pour évaluer les permormances des modèles, nous utiliserons les archives du réseau de caméras TENEBRE (propriété Météo-France). Huit caméras seront exploitées. Ces caméras sont installées dans des stations météorologiques et colocalisés avec des capteurs fiables (voir annexe B.3). Les archives utilisées dans ce manuscrit couvrent douze mois : les six mois d'octobre 2012 à mars 2013 et les six mois d'octobre 2017 à mars 2018. Pendant ces douze mois, chaque caméra « voit » deux à cinq périodes d'enneigement.
Le pas de temps est généralement de 10 minutes. Parmi les caméras, deux sont panoramiques. Dans ces cas, les images ont été redécoupées. Après redécoupage, nous obtenons ainsi dix scènes indépendantes. Parmi ces scènes, seule entzheim1 et entzheim3 correspondent à des scènes routières standard (voir figure 2.4).
Ce jeu de dix séquences est appelé TENEBRE_1213.

A partir de ces archives, nous avons formé deux jeux de données. Un jeu équilibré, contenant de l'ordre de 5.000 images par scène (TENEBRE_I). Les apprentissages conduits à partir de ce jeu, n'ont donné aucun résultat intéressant et nous ne l'évoquerons que rarement au long de ce manuscrit.
Nous avons aussi extrait des séquences d'images (de l'ordre de 200 par scène) de manière à rôder notre procédure d'annotation et à évaluer la correspondance entre la mesure instrumentale et la scène. Nous reviendrons sur la construction et la valorisation de ce jeu, appelé TENEBRE_IH , à la fin du chapitre.

5. Certaines des caméras des DIRs sont présentes dans la base AMOS. C'est le cas, par exemple, des archives associées aux sites numéros 32.769 et 32.770.

FIGURE 2.4 – Les dix scènes du jeu TENEBRE_1213. Seules les deux premières scènes peuvent être considérées comme des scènes routière. La dernière scène (Markstein) a généralement été exclue des analyses du fait de nombreuses mesures erronées.

2.2 Annotation

2.3 Annotation des images

A l'issue de la phase de collection, 24.980 images (en comptant celles de TENEBRE_IH) répartis dans 598 séquences (voir la table 2.3) ont été annotées à la main.
Faute de moyens, et surtout faute de recul [6], l'annotation a été réalisée par l'auteur. Cette tâche fastidieuse a coûté du temps, mais elle a débouché sur un jeu fiable, riche (annotation sur plusieurs critères) et homogène, dans la mesure où les critères d'annotation ont été suivis avec constance.
Le lecteur intéressé pourra consulter ces critères dans l'annexe A, où ils sont illustrés. Dans cette partie, nous décrivons plutôt les grandes étapes de l'annotation. Les questions liées à l'interprétation de l'image et à l'application des critères sont détaillées dans l'annexe A.

Sources d'images	Nombre de séquences	Nombre d'images	Images avec neige au sol	Cas de précipitations	Images de mauvaise qualité
AMOS	388	12.475	5.374	7.008	2.674
DIRs	98	8.715	5.155	6.442	1.219
Infoclimat	102	1.285	962	534	285
TENEBRE_IH	10	2.505	1.352	776	706
Total	598	24.980	12.843	14.760	4.884

TABLE 2.2 – Description des sources d'images utilisées dans cette thèse. Cette table résume le descriptif de l'annexe A.

A travers l'annotation, nous avons défini plusieurs problèmes de Machine Learning. La première étape a permis de définir des problèmes de classification (section 2.3.1). Au cours de cette étape, des labels [7] sont attribués aux images individuelles et les paires d'images consécutives sont comparées sur chacun des paramètres d'intérêt.
Pour compléter l'annotation par paires, qui nous a semblé plus avantageuse, une deuxième étape d'annotation a été réalisée. L'objectif était de construire un jeu de données de taille sensiblement supérieure,

6. Au début de la thèse, nous n'avions aucune expérience en matière de données webcam. Les phénomènes d'intérêt étaient précisés, mais la diversité de leur manifestations dans les images nous échappait. Le choix des critères d'annotation n'avait rien d'évident. Si, au terme de ce travail, il nous semble possible de déléguer l'annotation, c'est grâce à l'expérience acquise au cours de ces campagnes et à la lumière des résultats obtenus. Nous espérons, dans la perspective d'une utilisation opérationnelle, qu'une annotation par crowdsourcing pourra être envisagée à partir cette expérience.

7. Nous utilisons ce terme anglais pour désigner les cibles d'un problème de classification.

non pas en nombre d'images, mais en nombre de cibles (comparaisons). Pour augmenter le nombre de comparaisons, un algorithme d'annotation semi-automatique a été adapté (section 2.3.2.2).
La troisième étape d'annotation a été mise en œuvre bien plus tard, pendant la troisième année de thèse, pour corriger les défauts observés sur les prédictions relatives à l'étendue du manteau neigeux. Cette étape est en partie basée sur l'utilisation de modèles décrits au chapitre 4. Nous n'en proposons ici qu'un bref aperçu (section A.2).
Enfin, l'annotation des images du réseau TENEBRE à partir de données instrumentales est décrite en section 2.3.4.1. Parmi ces images, celles de TENEBRE_IH ont aussi été annotées à la main. Cela a permis de contrôler la correspondance entre l'annotation manuelle et la donnée instrumentale.

2.3.1 Première étape d'annotation

Pendant la prremière étape d'annotation, les séquences sont parcourues dans l'ordre chronologique [8]. Un logiciel d'annotation [9], conçu pour l'occasion, a permis d'annoter les images. La figure 2.5 en présente l'interface.
Des labels sont attribués, les uns relativement à l'image courante, les autres, relativement à la paire formée par l'image courante et l'image qui précède. Ces paires d'images sont dites « consécutives ». Mais encore une fois, le durée qui sépare deux images « consécutives » est variable : la régularité du pas de temps s'est perdue après l'étape de réduction de la redondance.
Tous les labels posés lors de cette étape ne sont pas destinés à définir un problème d'apprentissage. Certains serviront plutôt à interpréter les résultats.
Les paragraphes suivants décrivent les principaux labels. Une description exhaustive peut être trouvée dans l'annexe A.

Labels relatifs à l'image :
Les attributs les plus importants concernent l'état du sol et l'état de l'atmosphère. Pour caractériser l'état du sol sur les scènes routières, nous avons distingué cinq classes que l'on rencontre successivement au cours d'un épisode de neige typique (voir figure 2.6).

Les frontières entre ces cinq états ne sont pas toujours nettes. Des labels supplémentaires indiquent les cas ambigus (attribut **snow_traces**, voir annexe A-I.1.). L'information à disposition peut-être insuffisante pour statuer et des classes de rejet (labels **doubt**) sont prévues à cet effet. L'état du sol n'est complètement indéterminé que dans de rares cas, soit à cause d'un masque (par exemple figure 3.c-d, partie A), soit de nuit, sans éclairage artificiel à proximité (label **dark_night**). Le choix du label peut enfin être compliqué par les situation de contrejour avec reflets sur la voie. Des traces blanches appa-

8. Ce choix n'est pas anodin : l'interprétation de l'image courante est souvent facilité par la consultation des images précédentes.
9. voir sur https ://github.com/space-latmos/

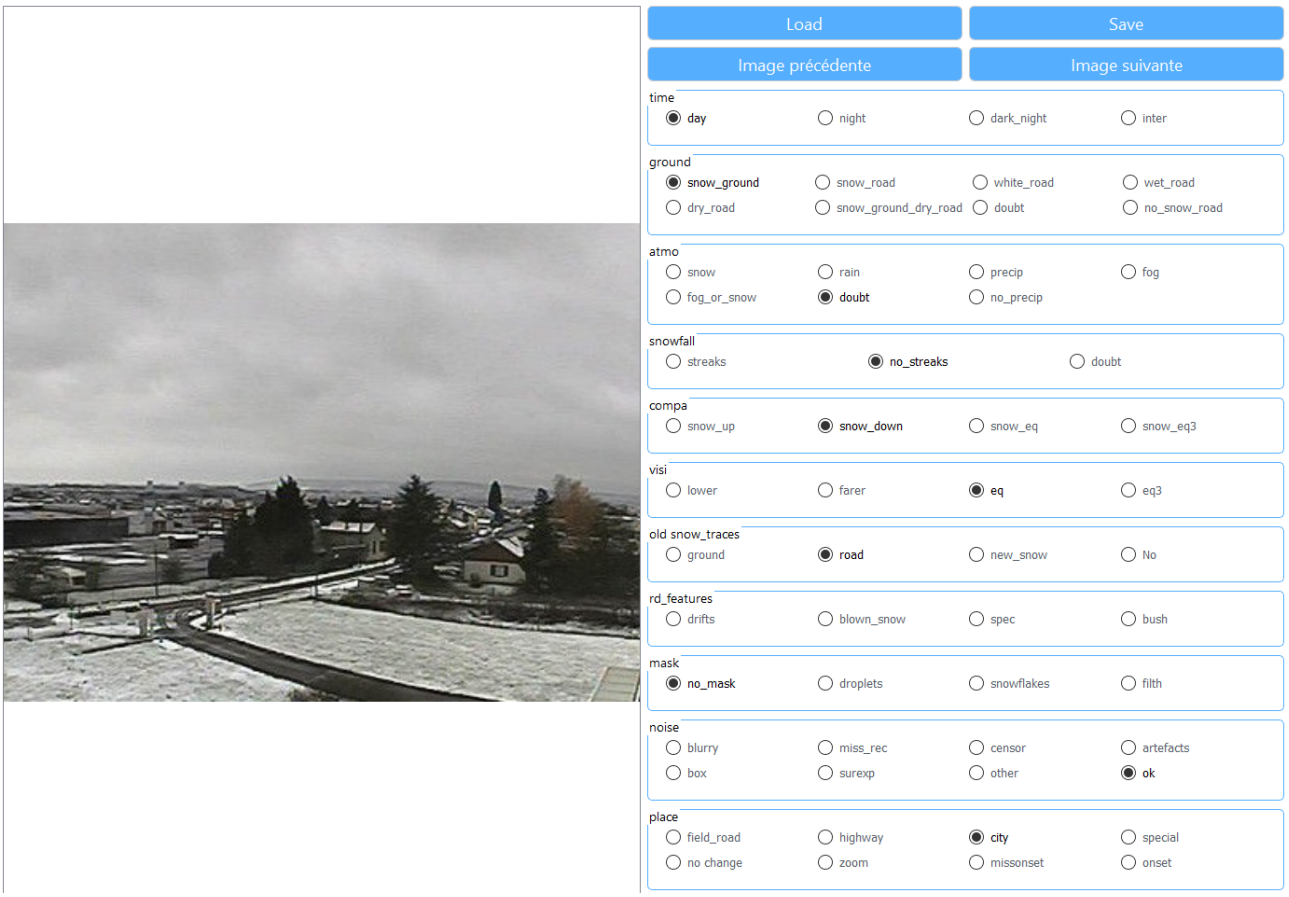


FIGURE 2.5 – Interface de LabelGO, logiciel d'annotation développé début 2019 pour l'annotation d'images. Après annotation complète, l'image est remplacée par la suivante dans l'ordre chronologique. Les attributs et les labels sont détaillés à l'annexe A.1.

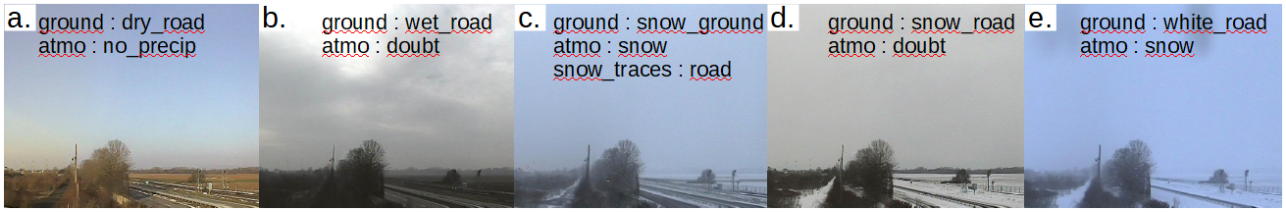


FIGURE 2.6 – Annotation de l'attribut « état du sol ». Cinq classes sont distingués : a. Par beau temps, la route est sèche (**dry_road**). b. Sous la pluie, ou au début des chutes de neige, la route s'assombrit et des reflets apparaissent (**wet_road**). c. La neige tient sur le sol, mais pas sur la route (**snow_ground**). La neige tient sur la route, mais ne la couvre pas entièrement (**snow_road**), la neige couvre entièrement la route (**white_road**).

raissent aussi lors de l'épandage de sel sur les routes ou en période de gelée blanche (figure 2.7).

La nature des précipitations est plus difficile à préciser. Quatre classes, **no_precip**, **snow**, **rain**, et **fog** sont proposées, mais le label **doubt** est utilisé dans plus d'un tiers des cas (voir figure 2.6.b,d et annexe B.1).

La neige peut être différenciée de la pluie ou du brouillard lorsque les traînées (**streaks**) associées aux flocons en chute libre sont présents dans l'image. L'état du sol, les flocons collés à la vitre de protec-

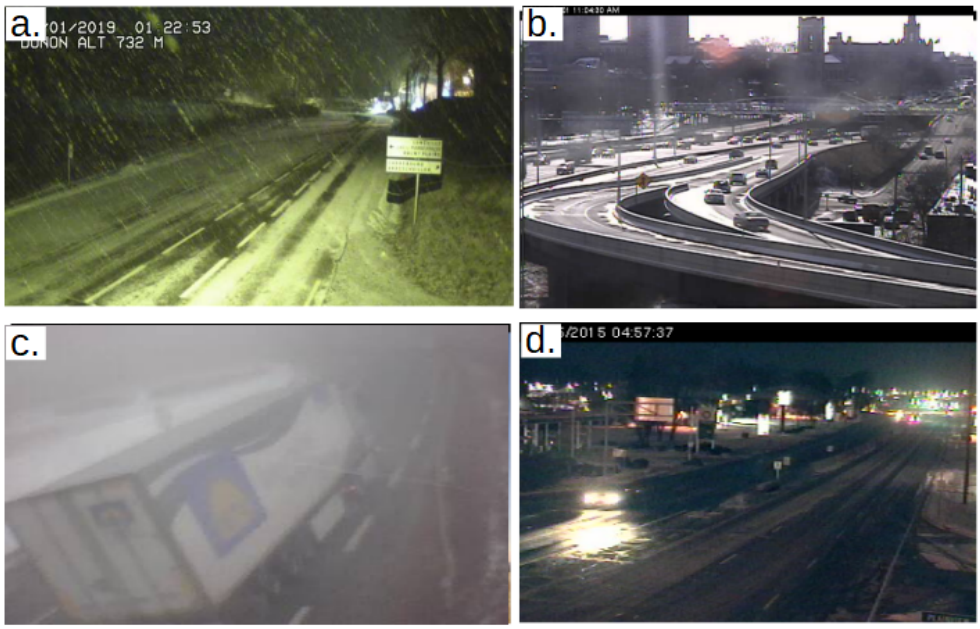


FIGURE 2.7 – Le contraste entre les parties claire et sombre de la chaussée sont dus à l'humidité. b. la route est exposée au soleil après une averse de neige. Les reflets du soleil peuvent donner l'impression que la route est couverte de neige. c. un cas de gelée blanche. d. sel fraîchement déposé sur la voie.

tion (**snowflakes**) sont d'autres éléments qui permettent de trancher. La présence de ces éléments est annotée avec chacun deux niveaux différents.
De nombreux facteurs réduisent l'information contenue sur les paramètres d'intérêt. Parmi eux, les flocons (voir figure 2.3.d) et les gouttes collés à la vitre (ou la lentille) de protection sont particulièrement fréquents pendant les précipitations. D'autres bruits (attribut **noise**), liés à la surexposition du capteur, à un défaut de focalisation, ou à la mauvaise transmission de l'image sont aussi renseignés. Enfin, la nature de la scène, le type d'éclairage (diurne, crépusculaire, artificiel), et d'autres éléments annexes sont rapportés.

Annotation relative aux paires consécutives :
Pendant la première étape, chaque image est aussi comparée avec la précédente. Pour faciliter la comparaison, l'image courante est superposée à l'ancienne. Les paires consécutives n'ont été comparées que sur deux critères : la visibilité et l'extension du manteau neigeux. Pour décrire le résultat des comparaisons, nous introduisons des notations qui seront utilisées tout au long du manuscrit (voir encadré 2.1).

De jour, la visibilité horizontale est définie par la distance maximale à laquelle on peut distinguer la silhouette d'un objet sombre situé à l'horizon.
La visibilité dépend de l'intensité des chutes de neige. La corrélation entre les deux est jugée suffisante pour prédire des classes d'intensité utiles à la météorologie opérationnelle [91]. La définition précédente n'est généralement pas applicable sur nos images : d'une part, la profondeur de la scène n'est

pas connue, et de l'autre, l'horizon n'est pas toujours visible. Par contre, il possible d'utiliser d'autres caractéristiques, qui covarient avec la visibilité, pour comparer deux images. Les critères utilisés sont détaillés dans l'annexe A.1.2.

La figure 2.8 illustre le critère n°2 (blanchiment des objets lointains). Les critères de l'annotation sont précisés et illustrés dans l'annexe A.1.2.

FIGURE 2.8 – les images consécutives a – f ont été parcourues dans l'ordre chronologique. Le contraste apparu entre l'arbre au premier plan et la végétation au second plan, dans le rectangle blanc, conduit à décider que la visibilité est plus grande d'après l'image b que d'après a (visi : lower).

Les deux autres paramètres qui nous intéressent sont l'étendue et l'épaisseur du manteau neigeux. Pour les paires d'images consécutives, il n'a pas été nécessaire d'annoter indépendamment ces deux paramètres : tant que la neige n'a pas atteint une route, ces paramètres covarient ensemble, Des règles spécifiques sont utilisées pour signaler les exceptions.

Lors de cette première phase, 24.980 images ont été annotées manuellement pour un total d'environ 200.000 labels, dont 50.000 portant sur les attributs principaux, c'est à dire sur l'état du sol et l'état de l'atmosphère.

2.3.2 Deuxième étape d'annotation

Pour compléter l'annotation par paires, nous avons comparé des images non-consécutives prises dans les séquences d'images à disposition.
Nous estimions que ces paires pouvaient conduire à de meilleures performances. En effet, certaines caractéristiques de l'image évoluent avec les paramètres d'intérêt sur les images consécutives. Par exemple, lorsque la visibilité baisse, la nébulosité (proportion du ciel couvert par des nuages) augmente, la luminance du ciel diminue et le plafond nuageux s'abaisse.
Ces relations ne sont plus vérifiées sur les paires non-consécutives ; ces dernières pouvaient donc aider à mieux séparer la variable d'intérêt des variables concurrentes. De plus, les paires non-consécutives permettent de mieux caractériser la qualité de l'information contenue dans l'image. On peut en effet considérer qu'une image souvent jugée incomparable aux autres contient moins d'information sur le paramètre d'intérêt.
Par rapport à l'annotation de nouvelles séquences d'images, un approfondissement de l'annotation par paires comporte aussi deux avantages pratiques. D'abord, l'étape de réduction de la redondance, coûteuse en temps, est évitée. Ensuite, il est possible d'augmenter sensiblement le nombre de paires comparées pour un effort d'annotation limité, en exploitant la transitivité, la règle d'équivalence (voir

ENCADRÉ 2.1 – Relations d'ordre : notations et terminologies

On utilisera la notation $x_i \prec_v x_j$ lorsque la visibilité sur l'image x_i est jugée inférieure à la visibilité sur x_j. Dans ce cas, la paire $\{x_i; x_j\}$ est dite strictement ordonnée.
Il arrive aussi que deux images x_i, x_j ne puissent pas être arrangées dans un ordre strict. Elles sont alors dites incomparables, ce qu'on note $x_i \perp_v x_{i+1}$.
Des notations analogues sont utilisées pour les autres paramètres : $\prec_s$, $\succ_s$, $\perp_s$ pour l'étendue ; et $\prec_d$, $\succ_d$, $\perp_d$ pour l'épaisseur. Lorsqu'il n'y a pas de risque de confusion, la lettre en indice est omise.
Lorsque ces relations définiront les cibles d'un apprentissage (prédiction par paire), ces cibles seront notées $\dot{\prec}$, $\dot{\succ}$ et $\dot{\perp}$.
Dans certains cas, deux images x_i, x_j peuvent sembler contenir la même information sur le paramètre d'intérêt. Cette situation se produit principalement dans trois cas de figures, discutés en annexe A. Nous la notons $x_i \sim x_j$ (c'est à dire $\sim_v$ dans le cas de la visibilité, $\sim_s$ et $\sim_d$ pour les autres paramètres).
Ce label est porté avec précaution : la relation $\sim$ doit toujours pouvoir être propagée à travers les autres comparaisons. Précisément, on doit pouvoir appliquer la règle suivante :

Règle 1 ***(Equivalence)*** *: Soient deux images x_i, x_j annotées comme « équivalentes ». Pour toute autre image x_k prise dans la même séquence, les implications suivantes seront considérées comme vraies :*

$$x_i \prec x_k \Rightarrow x_j \prec x_k \tag{2.1}$$
$$x_k \prec x_i \Rightarrow x_k \prec x_j \tag{2.2}$$
$$x_k \perp x_i \Rightarrow x_k \perp x_j \tag{2.3}$$
$$x_k \sim x_i \Rightarrow x_k \sim x_j \tag{2.4}$$

En particulier la relation $\sim$ définit une relation d'équivalence sur les images de la séquence.

Pour désigner une paire d'images strictement ordonnées, nous utiliserons parfois le terme de « comparaison stricte ». Une « incomparabilité » vaudra pour « une paire d'images incomparables » et une « équivalence » pour « une paire d'images équivalentes ».

encadré 2.1) et les labels déjà posés.

Nous avons limité l'annotation aux paires d'images prises dans une même séquence (paires intra-séquence). Cette restriction est importante mais légitime. En effet, entre images de séquences différentes (paires « inter-séquence ») les critères utilisés pour la comparaisons sont difficilement exploitables. La comparaison reste possible, mais elle devient plus grossière et la relation d'incomparabilité tiendra à une différence marquée entre les scènes plus qu'à un défaut de qualité dans l'image. Bien sûr, la comparaison d'images venant de caméras différentes présente un intérêt [37]. Mais pouvoir ranger les images d'une webcam quelconque dans l'ordre des visibilités, ou d'un autre paramètre, constitue déjà un défi scientifique. En nous restreignant aux paires intra-séquences, nous nous donnions les moyens de le relever, et de reprendre éventuellement, plus tard, avec l'avantage de séquences déjà triées, une annotation plus complète.

Une autre restriction importante doit être signalée : faute de temps, les images de nuit ont été écartées.

Pour produire un grand nombre de comparaisons rapidement, nous avons adapté un algorithme de tri basé sur des comparaisons binaires. Nous l'avons cherché dans une classe d'algorithmes qui sont à la fois économes en nombre de requêtes (l'annotateur doit être sollicité le moins possible durant la phase de tri) et qui permettent de tirer parti des relations issues de la première étape. L'algorithme de tri-fusion répondait à ces deux critères [92]. Pour tenir compte de la présence de relations d'incomparabilité, nous avons opté pour un algorithme de tri-fusion adapté au cas d'une relation d'ordre partiel [93].

Les sous-sections suivantes donnent le détail de la deuxième étape d'annotation. Nous expliquons d'abord comment, avant d'appliquer l'algorithme, nous nous sommes servis des labels quantitatifs pour produire de nouvelles comparaisons sans effort d'annotation supplémentaire (section 2.3.2.1). En section 2.3.2.2, nous décrivons dans ses grandes lignes notre version du tri-fusion pour une relation d'ordre partiel. Enfin, les jeux résultants sont présentés.

2.3.2.1 Exploitation des annotations de l'étape 1

Pour produire des annotations sans effort d'annotation supplémentaire, nous avons d'abord exploité deux règles d'extension. La première consiste à considérer des classes relatives à l'état du sol et à celui de l'atmosphère comme des catégories ordinales. Pour la visibilité, on considère ainsi que toute image associée au label **fog** (niveau 1, figure 2.9) correspond à une visibilité plus faible qu'une image associée au label **no_precips**.

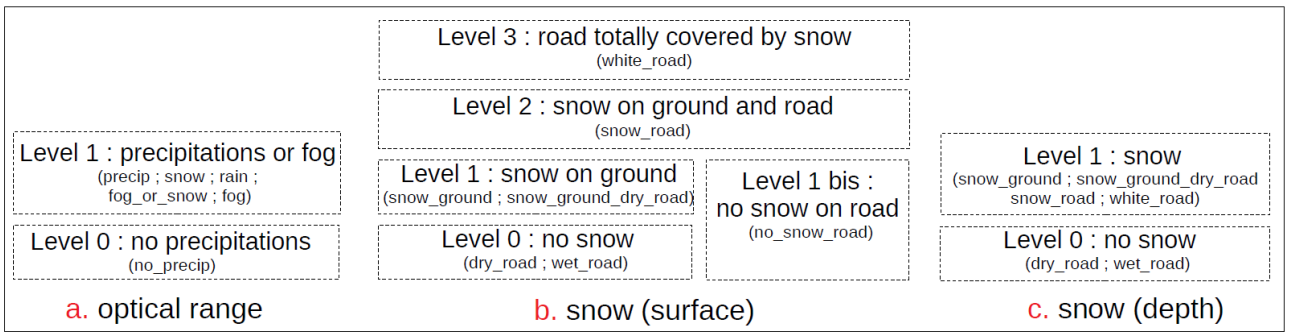


FIGURE 2.9 – définition des niveaux utilisés pour la règle 2 à partir des labels qualitatifs.

Par exemple, si l'on tient compte des labels associés aux images de la figure 2.10, on obtient automatiquement quatre nouvelles relations (comparaisons strictes).
De manière plus formelle, on ajoute les comparaisons impliquées par la règle suivante :

Règle 2 ***(Comparaison par niveau)*** *:*
Soient x_i une image de niveau k et x_j de niveau k'. On a : $k < k' \Rightarrow x_i \prec x_j$.
En dehors du cas de la visibilité, on considère que toutes les images de niveau 0 sont équivalentes.

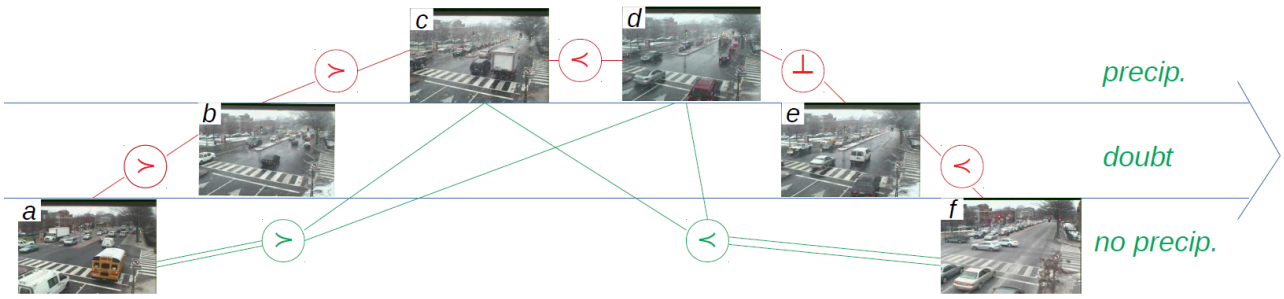


FIGURE 2.10 – Les relations obtenues pendant la première étape (paires consécutives) sont représentées en rouge. La règle 2 permet de tirer des relations des labels relatifs à l'image (en vert).

De plus, pour tous les paramètres, la relation $\prec$ est considérée comme une relation d'ordre partiel strict. La règle 3 consiste à combiner la transitivité de $\prec$ avec la règle d'équivalence :

Règle 3 ***(Propagation)*** *:*
- Appliquer l'implication 2.4) de la règle d'équivalence pour déduire de nouvelles équivalences par transitivité.
- Propager toutes les comparaisons strictes et les incomparabilités en appliquant les implications 2.1-2.3 de la règle d'équivalence.
- Ajouter toutes les comparaisons strictes qui sont dans la clôture transitive de la relation $\prec$.

Pour stocker les comparaisons intra-séquences, nous utilisons trois graphes par paramètre p : un graphe orienté $\mathcal{D}^p$ et deux graphes non orientés $\mathcal{U}^p$ et $\mathcal{E}^p$. Les noeuds des graphes sont les images à trier. Les arêtes de $\mathcal{D}^p$ représentent les paires d'images strictement orientées, celles de $\mathcal{U}^p$ représentent toutes les paires incomparables et celles de $\mathcal{E}^p$, les équivalences.

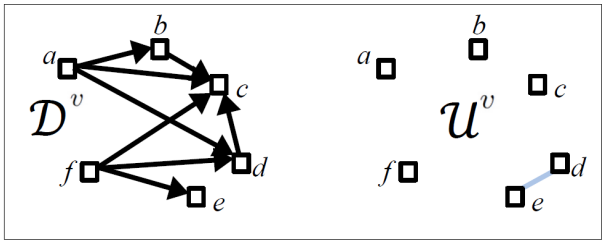


FIGURE 2.11 – Graphes associés aux relations de la figure 2.10. Comme il n'y a pas d'équivalences dans cette séquence, $\mathcal{E}^v$ n'est pas représenté.

Du fait de la règle d'équivalence, les graphes quotients $\mathcal{D}^p/\sim_p$, $\mathcal{U}^p/\sim_p$ et $\mathcal{E}^p/\sim_p$ sont bien définis. Appliquer la deuxième règle revient à chercher la clôture transitive du graphe orienté $\mathcal{D}^p/\sim_p$.

2.3.2.2 Tri par fusion pour les paires non-consécutives

Les comparaisons automatiques obtenues à partir des règles 2-3 sont grossières. Pour compléter le jeu par des comparaisons plus fines, un algorithme de tri fusion (*Poset-mergesort*) pour les ensembles

partiellement ordonnés [93] a été adapté.
Pour décrire notre version de cet algorithme, nous rappelons la définition suivante :

Definition 1 (**Chaîne**) Soit $(P, \prec)$ un ensemble partiellement ordonné (Poset [10]).
Un sous-ensemble de P totalement ordonné par $\prec$ est appelée chaîne.

Poset-mergesort est un algorithme de tri par comparaisons binaires adapté à un Poset. Il a été conçu pour obtenir une représentation du Poset appelée *chainmerge data structure*. Cette représentation est souhaitable parce qu'elle permet d'accéder rapidement à n'importe laquelle des $|P| \times (|P| - 1)/2$ comparaisons possibles.
L'algorithme, basé sur la recherche d'une décomposition en chaînes de taille minimale, est efficace dans le sens où il requiert relativement peu de comparaisons binaires. Il a encore deux avantages : il est relativement simple à mettre oeuvre et comme il généralise le tri fusion, il nous a semblé adapté au cas où les images sont déjà partiellement triées. Nous l'avons utilisé non pour obtenir une représentation particulière de la donnée, mais pour alléger l'effort d'annotation.

La recherche d'une décomposition en chaînes repose sur une réduction récursive du nombre de chaînes. A chaque étape, les comparaisons manquantes sont fournies par l'annotateur (voir figure 2.12). La deuxième partie de l'algorithme consiste à compléter les relations entre les images de chaque chaîne de la décomposition.
L'algorithme se termine lorsque la relation d'ordre partiel est entièrement déterminée (voir figure 2.13).

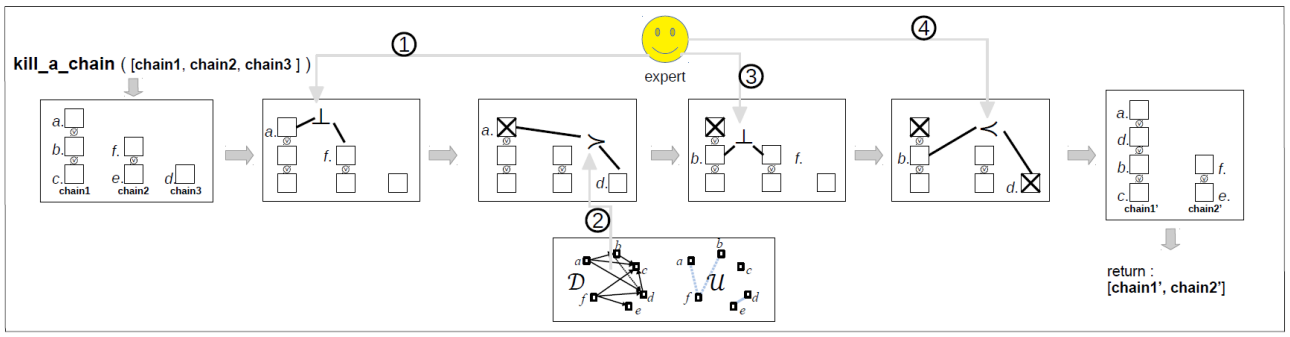


FIGURE 2.12 – Réduction du nombre de chaînes. Les images a, b, c, d, e, f sont celles des figure 4 et 5. A chaque étape, on compare les sommets de deux chaînes prises au hasard parmi les n chaînes de l'entrée. L'annotateur est sollicité (1,3,4) dès que la comparaison n'est pas contenue dans un deux graphes. Dès qu'une comparaison stricte est trouvée, le plus grand élément est mis de côté (croix noires). Lorsque toutes les images d'une des chaîne ont été mises de côté, un réarrangement en $n - 1$ chaînes peut être effectué.

Notre version de l'algorithme est adapté aux données à disposition : il ne tourne pas sur les images d'une séquence mais sur les noeuds des graphes quotients $\mathcal{D}^p/\sim_p$ et $\mathcal{U}^p/\sim_p$. Les comparaisons manuelles sont faites entre deux représentants des classes choisis au hasard. Les classes d'équivalence

10. Pour partially ordered set

sont mises à jour dès qu'une nouvelle équivalence est annotée par l'expert.
Nous avons inclus des fonctionnalités permettant de contrôler la cohérence de l'annotation, de visualiser la décomposition et de corriger des erreurs. De cette manière, nous nous assurons que les graphes résultants représentent bien une relation d'ordre partielle sur chacune des séquences.

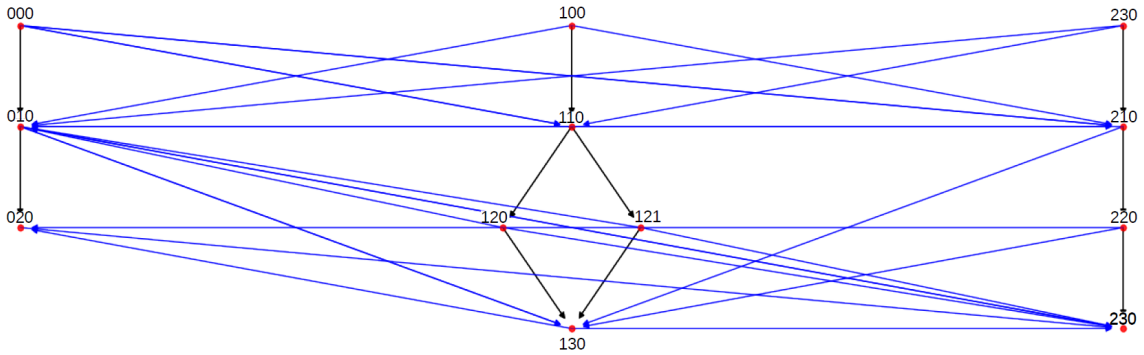


FIGURE 2.13 – Relation d'ordre partiel entre douze images (noeuds du graphe) triées suivant le critère de la visibilité. Les nombres associés aux images sont composés de trois numéros : le premier indique l'indice de la chaîne dans la décomposition, le second indique la situe dans la chaîne, le dernier la situe dans la classe d'équivalence de l'image (seule une classe comporte deux images : 120,121). Les arêtes noires figurent la décomposition en chaînes obtenue par *poset-mergesort*.

La table 2.3 contient la description des graphes obtenus pour chacun des paramètres. Toutes les séquences disponibles n'ont pas été entièrement triées. Pour la majorité des séquences, nous nous sommes arrêtés à une décomposition en chaînes sur des sous-séquences de soixante images au maximum (sauf exception, avec TENEBRE_IH, où entre 100 et 200 images par séquence ont été classées). Les nombre de décompositions obtenues et de tris menés jusqu'à terme sont donnés dans la deuxième colonne.
Dans la table 2.3, seules les arêtes associées à ces sous-séquences sont comptées (colonnes 3-5). Pour les deux paramètres relatifs à la neige, les images sans neige (niveau 0) ont été mises de côté le temps du tri et les arêtes qu'elles forment avec les images de niveaux supérieurs ne sont pas comptées dans la table.
Le nombre d'images moyen par classe d'équivalence est souvent supérieur pour les paramètres associés au manteau neigeux, qui varie plus lentement (colonne 6-8). Enfin, même en s'arrêtant à la décomposition, l'algorithme débouche sur une annotation nettement plus fine qu'avec la seule règle 2. Les chaînes comptent en moyenne un à deux niveaux de plus que sur la figure 2.9 et les plus longues chaînes observées peuvent compter jusqu'à vingt niveaux (colonnes 9-14).

Pour une minorité de séquences, le tri a été mené jusqu'à son terme. Cela permet d'évaluer l'efficacité globale du procédé. Pour une séquence de taille n, nous comptons la proportion de paires comparées manuellement parmi les $n \times (n-1)/2$ paires possibles (ratio PA/P). Plus la séquence est grande, et plus ce ratio est faible. Dans la table 2.3, nous indiquons ces ratios pour des séquences de plus de 20 éléments. Ces valeurs atteignent 30% pour les paramètres relatifs au manteau neigeux (soit 57 comparaisons pour une séquence de vingt images). Cette faible proportion tient non seulement à

	posets décomp.	Nb d'arêtes (en milliers) G – U E			nb. d'images par classe d'éq.			taille moyenne des chaînes			taille de la plus longue chaîne			Ratio PA/P		
Param.		v	s	e	v	s	e	v	s	e	v	s	e	v	s	e
AMOS	15-40 300	80 – 42 6,2	38 – 9 1,1	41 - 13 1,5	1,1	1,4	1,6	3	4	3,7	16	20	19	0,37	0,35	0,23
DIRs	30 90	68 – 23 1,5	50 - 15 1,3	45 - 23 1,7	1,8	1,9	2,5	3,1	4,6	4,1	24	27	16	0,43	0,3	0,31
Infoclimat	96 102	4 - 2 0,3	3 - 2 0,4	3 - 2 0,3	2	2,2	2.1	2,8	3,2	2,9	8	9	9	0,45	0,32	0,31
TENEBRE qual	2-5 3-5	17 - 7 0,2	/	/	1.1	/	/	3,5	/	/	17	/	/	0,41	0,3	0,28

TABLE 2.3 – Statitistiques sur les sous-séquences triées avec *poset-mergesort*. La deuxième colonne indique le nombre de tris complets et le nombre de décompositions obtenues. Par exemple, pour AMOS, 300 sous-séquencse ont été décomposées et entre 15 et 40 des ces sous-séquences ont été complètement triées (cela dépend du paramètre). Dans les colonnes 3-5, les tailles (en nombre d'arêtes) des trois graphes sont indiquées pour chacun des trois paramètres : visibilité (v), étendue du manteau neigeux (s) et épaisseur du manteu (d). La taille moyenne des chaînes contenues dans les décompositions apparaît aux colonnes 9-11, et la taille de la plus longue chaîne contenue dans les graphes $\mathcal{G}^p$ apparaît, pour chaque paramètre p, dans les colonnes 12-14. Le ratio PA/P est défini dans le corps du texte.

l'utilisation de l'algorithme, mais aussi à l'utilisation des équivalences (entre 1,1 et 2,5 images par classe d'équivalence en moyenne) et aux règles 2-3.

A la fin de la campagne d'annotation semi-automatique, 83.000 annotations par paires ont été réalisées sur les images prises de jour. En se restreignant aux sous-séquences sur lesquelles l'annotation a été densifiée, le nombre total de comparaisons atteint environ 200.000 pour la visibilité. La quantité d'arêtes est du même ordre pour les deux autres paramètres, si l'on compte les arêtes formées avec les images sans neige.

2.3.3 Troisième étape d'annotation

Les premiers modèles appris sur les paires relatives à l'étendue présentaient de nombreuses fausses détections en dehors des périodes de neige. Pour obtenir des résultats ayant un intérêt au plan opérationnel, nous avons procédé à une étape d'annotation supplémentaire « par sous-séquences ». Pour chacune des sous-séquences soumises à l'annotateur, ce dernier précise si l'étendue du manteau croît, décroît, ou si elle reste constante. Dans ce dernier cas, il précise si les images peuvent être considérées comme équivalentes. Ces annotations sont converties en comparaisons par paires et intégrées aux graphes obtenus à l'étape 2.
Comme l'extraction des sous-séquences fait intervenir un modèle décrit au chapitres 4, le détail de cette dernière étape est laissé à l'annexe A.2.

2.3.4 Annotation des images TENEBRE. Correspondance avec l'annotation à la main.

Dans cette partie, nous précisons comment les images de TENEBRE sont annotées à partir des mesures colocalisées (section 2.3.4.1). Nous présentons ensuite une rapide étude sur la correspondance avec des labels portés à la main sur le jeu TENEBRE_IH. Nous commentons d'abord (section 2.3.4.2) la correspondance avec les labels qualitatifs : l'état du sol est comparé à la mesure d'épaisseur -positive ou nulle- et l'état de l'atmosphère avec la donnée pluviomètre. Nous nous intéressons ensuite à la concordance entre l'annotation à la main et la mesure (section 2.3.4.3). Des éléments complémentaires sont donnés dans l'annexe B.3.

2.3.4.1 Annotation par des mesures colocalisées

Les images ont été associées à toutes les mesures (vent, température, nébulosité,etc) disponibles en vue d'autres utilisations. Nous n'évoquons ici que les mesures de précipitations, de visibilité et d'épaisseur de neige.
Les capteurs utilisés pour ces trois mesures sont des capteurs standards (voir table B.3). Les données au pas de temps 1 minute sont utilisées dès qu'elle sont disponibles (visibilité, précipitations). Nous avons associé chaque image avec la mesure qui lui correspond. Pour les précipitations, on utilise aussi des données au pas de temps 6 minutes. L'image est associée à la première mesure après la date de prise de vue. Pour l'épaisseur de neige, on utilise une donnée horaire et une interpolation linéaire à la date de la prise de vue. Ces opérations ont été appliquées à l'ensemble des images des archives TENEBRE couvrant les périodes début octobre 2012 - fin mars 2013 et début octobre 2017 - fin mars 2018 (i.e. le jeu TENEBRE_1218, voir section 2.1.3).

Pour contrôler la qualité de nos critères d'interprétation et de notre méthode d'annotation, nous les avons appliqués à un sous ensemble des images TENEBRE_1218, le jeu TENEBRE_IH.
Les images du jeu TENEBRE_IH ont été sélectionnées dans TENEBRE_1218 à partir des mesures instrumentales de façon à obtenir un jeu équilibré : pour chaque scène, on sélectionne au hasard au plus 50 images associées à une épaisseur de neige supérieure à 10 cm, puis on réitère l'opération pour des épaisseurs comprises entre 5 et 10 cm puis entre 0,1 et 5 cm. Enfin, on tire 50 images sans neige au hasard. On complète en reprenant le même procédé avec le paramètre visibilité : on choisit au hasard 50 images supplémentaires avec une visibilité inférieure à 200 m, puis 25 images avec une visibilité comprise entre 1000 m et 200 m puis 25 images avec une visibilité supérieure à 1000 m.
Le jeu résultant est décrit dans les tableaux A0-3, en annexe A-II. C'est sur ce jeu qu'on étudie la correspondance entre les deux types d'annotation.

2.3.4.2 Correspondance entre les mesures et les annotations qualitatives

Sur la discrimination neige au sol/non-neige au sol, les annotations humaine et instrumentales sont contradictoires sur 8 % des images (voir le détail section B.3). Sur ces 8 % (131 images), on compte au plus 1,8 % d'erreurs d'annotation manuelle contre au moins 4,4 % de valeurs instrumentales en

désaccord avec l'image. Le reste des 8 % est associé à des cas tangents (traces de neige).
Enfin, ces contradictions sont très majoritairement des cas de neige apparente mais non mesurée (7,5%).

Sur la classification précipitations/non précipitations, la correspondance est nettement moins bonne. Si l'on se restreint aux trente images sur lesquelles des traces de flocons attestent de chutes de neige (label streaks), seules cinq d'entre elles sont associées à un taux de précipitation "instantané" (cumuls sur une minute) non nul. Avec les taux moyennés sur six minutes, la correspondance n'est que légèrement meilleure (six images sur trente). Ces mauvais résultats sont en partie [11] dus à la difficulté d'enregistrer les chutes de neige avec des pluviomètres.
En considérant toutes les images associées à des précipitations pendant l'annotation (labels **precip**, **snow**, **rain**), seules 9% d'entre elles sont associées à des mesures (au pas de temps 6 minutes) non nulles. D'un autre côté, quelques cas d'annotations imprudentes (classe **no_precipitation** au lieu de **doubt**) ont été repérés, mais ces erreurs sont très rares (moins de 1%).
Cette situation nous a dissuadé d'entraîner des modèles de prédiction à partir de la pluviométrie mesurée.

2.3.4.3 Correspondance entre les mesures et l'annotation par paires

Nous avons aussi cherché à contrôler la qualité de l'annotation par paires. Pour un ensemble de paires d'images strictement ordonnées, nous nous sommes appuyés sur le taux de concordance avec la mesure, c'est à dire la proportion de paires d'images arrangées par l'annotatuer dans le même ordre que les mesures instrumentales.
Nous avons étudié ce taux de concordance pour les deux paramètres visibilité et épaisseur de neige. Les résultats de cette étude sont synthétisés dans les paragraphes suivants.

Pour l'épaisseur de neige, le taux de concordance est calculée sur les paires strictement ordonnées associées à la scène nancy2. Sans restriction particulière, elle est supérieure à 97 %. Les vingt discordances observées correspondent pour moitié à des mesures non représentatives de la scène, déjà évoquées. Les autres sont des erreurs d'annotation, généralement dues à un manque de prudence (il aurait fallu rejeter la comparaison).
Si l'on se restreint à des mesures séparées d'au moins trois centimères d'épaisseur, la concordance est parfaite. Pour des différence plus faibles (0.5 - 1 cm), le taux de concordance reste supérieure à 80 %.
Enfin, si l'on considère les paires strictement ordonnées suivant l'étendue du manteau neigeux, le taux de concordance tombe à 88 %. Cet écart (88% contre 97%) met en lumière la qualité des critères d'annotation pour la profondeur.

11. L'écart-type sur les timestamp, qui atteint cinq minutes pour certaine caméras de TENEBRE peut aussi avoir joué un rôle, mais ce n'est pas, à notre avis, le facteur le plus important.

Sur le critère de la visibilité, le taux de concordance a été calculé pour cinq caméras différentes. Sur toutes les paires strictement ordonnées disponibles, il va de 91 % (Roissy) à 97,5 % (entzheim3).
Si l'on se restreint à des paires de visibilités (v_i, v_j) où la différence relative $\frac{|v_j - v_i|}{v_i}$ est supérieure à 20 %, le taux de concordance est légérement meilleur sur toutes les scènes et toujours supérieur à 93 %.
Cette concordance assez fragile tient en partie à une incertitude sur la mesure assez élevée : pour le DF320, elle est de l'ordre de $\pm 20\%$ dans 90% des situations. Cela peut aussi s'expliquer, en partie, par des erreurs sytématiques sur l'horodatage (incertitude de l'ordre de ± 5 min.).

Les cas de discordance ont été passés en revue. Dans l'ensemble, les discordances sont effectivement dues à un défaut de représentativité de la mesure. Dans le cas particulier de la scène portail_entzheim, les mauvais résultats sont aussi dus à la difficulté de la scène. Les erreurs d'annotation qui ont été relevées sur cette scène ont d'ailleurs été profitables : elles n'ont pas été reproduites pendant la construction des jeux d'apprentissage et de test.
Nous notons aussi que la mesure associée à la caméra de Roissy est moins souvent représentative de l'image que les autres, probablement à cause de la distance entre l'instrument et la caméra.

2.3.4.4 Conclusion de l'étude

Cette étude aura permis plusieurs choses. D'une part, comme elle a été réalisée avant la deuxième étape d'annotation, les erreurs révélées par des contradictions avec la mesure ont été prises en compte au moment de la construction des jeux d'entraînement. D'autre part, elle précise ce qu'il est possible d'atteindre, en matière de performances, avec les données TENEBRE prises comme vérité terrain. Sur le problème neige/non neige, par exemple, un modèle parfait ne dépasserait pas 95 % de justesse. Pour un modèle entraîné à la prédiction de préférences (par image ou par paires) sur le paramètre visibilité, le constat est le même. Pour certaines caméras, comme Roissy, la justesse dépasserait difficilement 90%. Ces éléments seront à prendre en compte pour bien juger de la marge de progression restante.

Chapitre 3

Résultats de base : classification et apprentissage par paires. Cas de la visibilité.

Dans ce chapitre, nous présentons les principaux problèmes d'apprentissage que nous avons abordés à partir des jeux présentés au chapitre précédent. Les problèmes de classifications sont abordés en section 3.1, à partir de réseaux de neurones à couche de convolution standards. Dans la section 3.2, nous définissons un problème de learning to rank à partir des paires d'images comparées à la main (3.2.1). Le nombre d'image à disposition est du même ordre de grandeur que pour la classification, mais le nombre de paires annotées est supérieur d'un ordre de grandeur. Nous introduisons aussi les métriques adaptées à la prédiction d'une relation d'ordre partiel, de façon à comparer des modèles sur leur capacité à restituer aussi l'incomparabilité. Nous présentons enfin des méthodes d'apprentissage par paires, pour la prédiction par paires et pour la prédiction par images (sections 3.2.2-4).
Ces méthodes seront appliquées aux trois paramètres d'intérêt (visibilité, étendue et épaisseur du manteau neigeux). Mais dans ce chapitre on se concentre sur le cas de la visibilité (section 3.3). C'est en effet par ce paramètre que nous avons commencé l'annotation, dans l'idée d'aborder la caractérisation de la couverture neigeuse avec des algorithmes d'annotation et d'apprentissage déjà rodés.
Nous confirmons d'abord l'intérêt de la deuxième étape d'annotation (3.3.1) et celui des paires d'image incomparables (3.3.2). Nous comparons aussi des réseaux entraînés à la prédiction par paires à des réseaux siamois (3.3.3) et des méthodes plus classiques (3.3.4). Enfin, une première application à la détection de basses visibilités est présentée (3.3.5).

3.1 Calssification de l'état du sol et de l'atmosphère

Dans cette partie, nous présentons les résultats obtenus à partir d'architectures pré-entraînées sur les deux principaux problèmes de classification qu'il est possible de définir à partir de l'annotation qualitative : la classification de l'état du sol (degré d'enneigement) et de l'atmosphère (nature des précipitations).

3.1.1 Définition des problèmes de classification

Les problèmes de classification ont été abordés avant que toutes les images évoquées au chapitre précédent soient disponibles. Les modèles présentés dans cette section ont été entraînés sur l'ensemble des images [1] d'AMOS à disposition à l'exception des scènes de nuit mal éclairées (9.000 images/380 scènes différentes). La validation a été réalisée sur des images [2] issues des caméras des DIRs (6.000 images/50 scènes différentes), avec la même exception concernant les scènes de nuit mal éclairées.
Le but de cette étude préliminaire était d'évaluer le potentiel d'un apprentissage par classification. Nous l'avons fait à partir des seules performances en validation.

Pour chaque attribut disponible, nous nous sommes intéressés à deux types de problème (table 3.1). Le premier type, le problème « complet », est un problème de classification multiclasses où les labels présentés au chapitre 2 sont quasiment tous représentés par une classe. Le second type est un problème de classification binaire, déjà abordé dans la littérature, dont les classes agrègent différents labels.

Pour ces problèmes, les classes sont légèrement déséquilibrées (voir annexe A-I, tables A0-3, lignes 1 et 6). Pour évaluer les performances en validation, nous indiquerons donc plutôt la justesse équilibrée [94] définie par :

$$\mathcal{A}_b = \frac{1}{|\mathcal{C}^{val}|} \sum_{c \in \mathcal{C}^{val}} \Big[\frac{1}{|X_c^{val}|} \sum_{x \in X_c} \mathbb{1}_{c_{pred}(x)=c} \Big] \tag{3.1}$$

où $\mathcal{C}$ est l'ensemble des classes du problème et X_c^{val} désigne l'ensemble des images du jeu de validation associées à la classe c.

3.1.2 Méthodes d'apprentissage

Les réseaux de neurones profonds qui ont été comparés (table 3.2) sur ces problèmes sont téléchargés, initialisés et entraînés avec la bibliothèque pytorch. Le nombre de neurones N_l de la dernière couche complètement connectée est réglé sur le nombre $N_c(a)$ de classes cibles pour l'attribut a.

Les réseaux sont soit initialisés au hasard, suivant une méthode standard [61], soit à partir de réseaux pré-entraînés sur ImageNet ou sur le jeu Places365. Les différentes méthodes d'optimisation et

1. Ces images correspondent au jeu d'entraînement de la table A-1 (ligne 2) en annexe A-II
2. Ces images correspondent à 90 % des images de la table A-1 (ligne 6) en annexe A-II. Ici utilisées pour la validation, elles sont basculées dans le jeu d'entraînement pour les apprentissages par paires.

Attribut	Classification binaire	meilleure $\mathcal{A}_b$ (archi.)	Classification multiclasse	meilleure $\mathcal{A}_b$ (archi.)
Eclairage	Problème : day/night Classe 1 : {**day**} Classe 2 : {**night** ; **dark_night**}	0.98 ResNet18	Problème complet : Classe 1 : {**day**} Classe 2 : {**inter**} Classe 3 : {**night** ; **dark_night**}	0.7 ResNet18
Etat du sol	Problème : snow/no snow Classe 1 : {**Dry_road** ; **wet_road** (no traces) } Classe 2 : { **Snow_Ground** ; **Snow_Road** ; **White_Road** ; **Snow_Ground_Dry_Road** }	0.79 ResNet50 (f.c. pondérée)	Problème complet : Classe 1 : {**Dry_Road**} Classe 2 : {**Wet_Road**} Classe 3 : {**Snow_Ground_Dry_Road** ; **Snow_Ground** } Classe 4 : {**Snow_Road**} Classe 5 : {**White_Road**}	0.55 ResNet50 (f.c. pondérée)
Typologie des précipitations	Problème : precip. /no precip. Classe 1 : {**No_Precip** ;**Doubt** } Classe 2 : { **Rain_Ground** ; **Fog** ; **Precip** ; **Snowfall** }	0.74 ResNet50 (multi-attrib., f.c. pondérée)	Problème complet : Classe 1 : {**No_Precip**, **Doubt** } Classe 2 : {**Rain ; Fog ; Precip**} Classe 3 : {**Snowfall**}	0.54 ResNet50 (multi-attrib., f.c. pondérée)

TABLE 3.1 – Meilleurs scores (justesse équilibrée) obtenus sur des problèmes de classification définis à partir des labels qualitatifs.

les hyperparamètres des apprentissages sont donnés dans la table 3.2. Le code est mis à disposition dans l'annexe numérique. La fonction de coût est l'entropie croisée, pondérée ou non par les tailles des classes.

Des apprentissage multi-attributs (Multi-Attribute Learning, MAL) ont aussi été mis en oeuvre. Dans ce cas, $N_l = \sum_{c \in \mathcal{C}} N_c(a_i)$ et la fonction de coût est simplement la somme des entropies croisées associées à chacun des sous-problèmes.

Enfin, les classifieurs entraînés sur les problèmes complets ont été comparés aux classifieurs entraînés sur les problèmes de classification binaire. Pour établir ces comparaisons, les sorties de la couche softmax associées à une même classe du problème binaire sont sommées, et la classe prédite est celle qui correspond à la somme la plus grande.

3.1.3 Résultats

Nous avons commencé par entraîner des modèles sur une tâche simple : la classification jour/nuit. Sur cette tâche, la délimitation relativement arbitraire entre classe jour (**day**) et classe **inter** (réunissant aube et crépuscule) expliquent une justesse de 0,7 sur le problème à trois classes (table 3.1). Lorsque la classe **inter** est exclue du jeu de validation, les modèles atteignent 98 % de justesse.

Pour tous les problèmes abordés, la fonction de coût (voir figure 3.1) et la justesse équilibrée (équation (3.1)) sur le jeu de validation atteignent un palier au bout de 50 à 75 époques.

La performance atteinte est sensiblement meilleure lorsque le réseau a été pré-entraîné (figure 3.1). De plus, la « taille » du réseau joue de manière différente selon que le réseau est pré-entraîné ou pas : un réseau plus gros (ResNet50, ResNet101, VGG16) n'est plus performant qu'un petit réseau (ResNet18)

Architectures	- ResNet18, ResNet50, ResNet101 - VGG16
Augmentation de données	- réflexion par rapport à un axe vertical - perturbations dans l'espace HSV (*colorjitter*) - perturbation affine (*RandomAffine*) - rognage (*RandomCrop*) - réduction (utilisation des moyenne et variance calculées sur ImageNet) - dimensions spatiales des images d'entrée : $\mathbf{256 \times 256}$ (196×196 - 448×448)
Mini-batches	taille : $\mathbf{32}$ images (16 - 32) sélection : même fréquence pour toutes les séquences.
Optimisation	taux d'apprentissage initial (lr) : $\mathbf{10^{-4}}$ (10^{-5} à 10^{-3}) méthodes : - **ADAM [95]** - SGD + scheduler ($\times 0.1$ à $\times 0.5$ toutes les 20-30 époques)
Fonction de coût	-entropie croisée (CE) sans pondération -CE pondérée par l'inverse des fréquences des classes. -somme des CE pondérées (multi-attributs)
Autres	"freezing" (si pré-entraînement) : Pour un nombre variable de couches de convolution (**aucune** ou toutes), les poids du réseau ne sont pas mis à jour.

TABLE 3.2 – Choix des hyperparamètres. Les hyperparamètres en gras ont été retenus après avoir testé différentes valeurs situées dans les intervalles entre parenthèses. Les autres termes en italique précisent la fonction pytorch utilisée.

que s'il a été pré-entraîné (figure 3.1). Pour les problèmes de classification binaire, les justesses équilibrées sont systématiquement supérieures lorsque le classifieur a été entraîné sur un problème complet puis ramené à une prédiction binaire.
Enfin, lorsque la fonction de coût utilisée est pondérée (table 3.2), la justesse pondérée, mécaniquement, est meilleure.
Les résultats sont détaillés et mis en perspective par attribut dans les sous-sections suivantes.

3.1.3.1 Classification de l'état du sol

Le modèle ResNet50 pré-entraîné sur ImageNet avec une fonction de coût pondérée a fourni la meilleure justesse équilibrée (0,55) sur le problème complet. Sans pondération, la justesse équilibrée retombe à 0,45.
Avec un ResNet50, un pré-entraînement sur Places365 ne permet pas d'obtenir de meilleurs perfor-

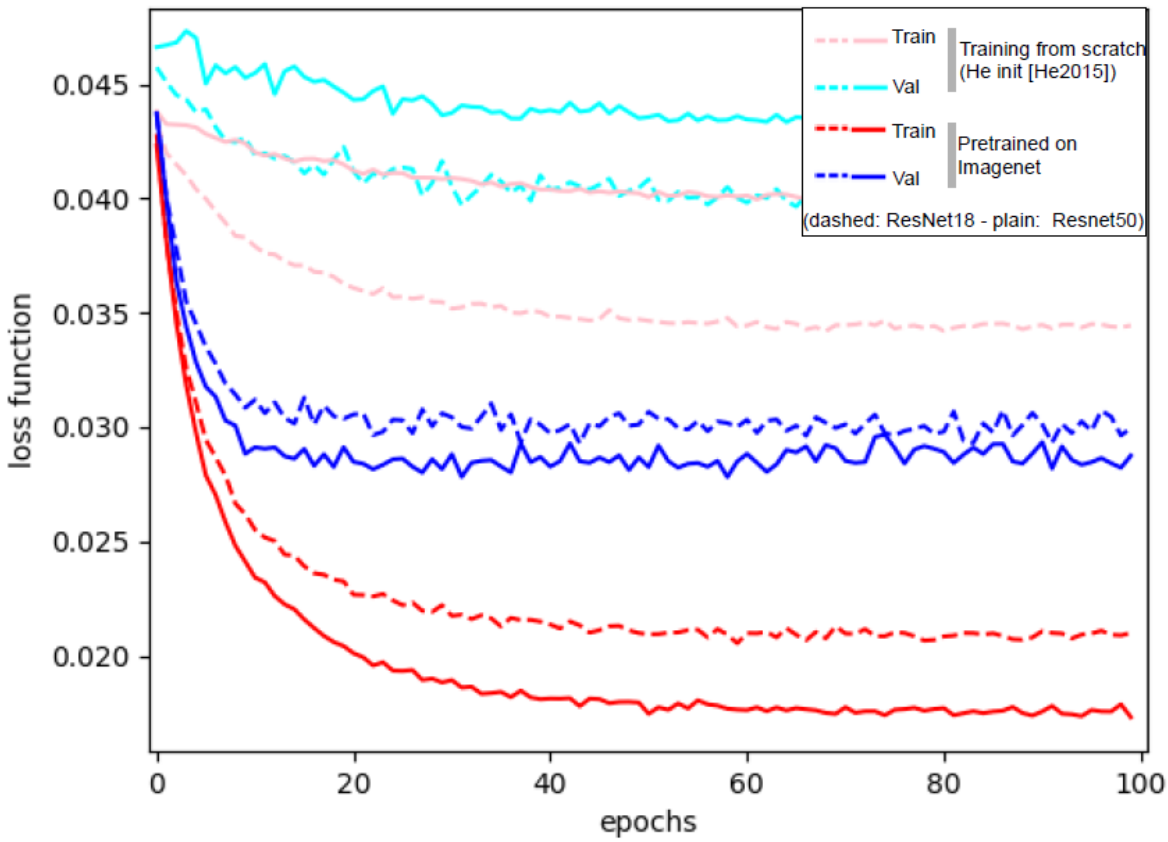


FIGURE 3.1 – Evolution de la fonction de coût (entropie croisée - équation 1.1) pendant l'apprentissage d'un ResNet50 et d'un ResNet18 sur le problème de la classification de l'état du sol (problème complet).

mances. La ressemblance entre les images d'extérieur de Places365 et celles de nos jeux de données n'a pas apporté de plus-value, l'apprentissage multi-attributs n'a pas eu d'effet non plus.

Sur le problème à deux classes (neige/non neige), la meilleure justesse équilibrée est de 0,79. Elle est atteinte par un ResNet50 entraîné au problème à cinq classe et ramené à une prédiction binaire. Ce même modèle atteint (en validation) une sensibilité [3] de 0,85 et une spécificité [4] de 0,72.

Sur un problème analogue (neige/non neige) défini à partir de séquences webcams, Kosmala et al. [Kosmala2019] obtiennent une sensibilité de 0,83 sur des caméras indépendantes de leur jeu d'entraînement. La spécificité est par contre nettement meilleure (0,94).

Leur méthode est relativement proche de la nôtre : un SVM est entraîné sur les caractéristiques extraites par la dernière couche de convolution d'un VGG16 (couche L_{13}, figure 1.2) pré-entraîné sur Places365. Comme les méthodes sont proches, la différence entre les spécificités s'explique plutôt par les images en jeu. Dans notre jeu de validation, toutes les images sont prises en hiver, par des conditions d'éclairage et des conditions météorologiques très diverses (et en particulier de nuit), avec une surreprésentation des images difficiles à classer.

Au contraire, le problème abordé par Kosmala et al. [35] est plus simple dans la mesure où leur jeu contient des images prises autour de midi, tout au long de l'année, et que les images de mauvaises

3. Proportion des prédictions de neige au sol parmi les cas de neige au sol observés.
4. Proportion des prédictions d'absence de neige au sol parmi les cas d'absence de neige au sol observés.

pred. / obs.	{DrR}	{WeR}	{SG;SDR}	{SR}	{WhR}
{DrR}	151	120	85	14	16
{WeR}	109	274	70	23	44
{SG;SDR}	228	435	1806	454	178
{SR}	71	61	558	1240	257
{WhR}	2	2	6	45	129

TABLE 3.3 – Meilleur matrice de confusion sur le jeu de validation pour la classification de l'état du sol (problème « complet », voir table 3.1) . DrR : dry_road, WeR : wet_road, SG : snow_ground, SDR : snow_ground_dry_road, SR : snow_road, WhR : white_road.

qualité ont été retirées.

Nous observons d'ailleurs des cas d'erreurs similaires à ceux rapportés par Kosmala et al. :

- les images sur lesquelles la neige est éparse, discrète ou difficile à identifier (images de nuit) sont souvent mal classées. Elles représentent une majorité des 228 + 435 cas de non-détections de la matrice de confusion (cf. table 3.3, nombres en rouge).
- les masques sur la lentille (flocons et gouttelettes), les reflets du soleil sur la chaussée génèrent des faux positifs.

Des causes plus spécifiques à notre jeu de données ont été observées, comme la présence de véhicules blancs dans l'image (fausses détections), la neige mouillée, plus terne, courante en ville et sur les grands axes routiers. Lorsque la scène contient un plan d'eau, la présence d'écume déclenche aussi des fausses détections.

On relève enfin des faux négatifs lorsque la surface enneigée suit les contours du marquage au sol.

Dans le problème complet, les erreurs sont nombreuses entre les classes d'enneigement voisines (en bleu, table 3.3). Elles concernent en particulier les nombreux cas tangents où la neige a atteint le bas-côté mais pas la bande de roulement.

3.1.3.2 Classification de l'état de l'atmosphère

Le deuxième problème de classification a été abordé avec les mêmes modèles. La meilleure justesse équilibrée (0,54) est aussi obtenue avec un ResNet50 appris sur une tâche de classification multi-attributs (état du sol, précipitations, masque et présence de traînées).

Dans ce problème, la classe **Rain**;**Fog**;**Precip** regroupe les cas de pluie (Rain), de basse visibilité (**Fog**) et les précipitations indifférenciées (**Precip**). Sur le jeu de validation elle est peu représentée et très mal prédite.

pred. / obs.	{NP;D}	{R;F;P}	{S}
{NP;D}	2369	205	794
{R;F;P}	57	23	139
{S}	685	296	2121

TABLE 3.4 – Meilleure matrice de confusion sur le jeu de validation pour la classification de l'état de l'atmosphère (problème « complet », voir table 3.1). Le modèle est un ResNet50 entraîné simultanément sur quatre problèmes de classification (état du sol, état de l'atmosphère, présence de masque, présence de traînées). NP : no_precip, D : doubt, R : rain, F : fog, P :precip, S :snow.

Les 23 cas observés et correctement prédits (case centrale, figure 3.4) sont des cas de brouillard sans neige au sol, plus faciles à différencier à l'oeil des cas de chutes de neige. Les 794 fausses détections de neige sont en grande majorité des images difficiles à classer (label **doubt**).
Exceptionnellement, des fausses détections de chutes de neige se produisent par beau temps. Nous n'avons rencontré ces fausses détections que sur les scènes sans second plan, où l'horizon n'est pas visible (ce sont les ombres portées au sol qui excluent les chutes de neige). Dans les cas en question, il y a systématiquement de la neige au sol. Cela suggère que le modèle utilise maladroitement la corrélation entre chute de neige et neige au sol.
Les faux négatifs comptent de nombreux cas de ciel gris difficiles à classer sans l'aide du contexte (images précédente et suivante).

Sur le problème de classification binaire précipitations/non-précipitation obtenu en fusionnant les deux dernières classes du problème complet (voir table 3.1, le meilleur score en validation (justesse équilibrée de 0,74) est encore obtenu par un modèle entraîné au problème complet et « binarisé ».
Or ce problème a déjà été abordé dans la littérature, avec un certain succès, dans différents contextes (caméra embarquée [26], images singulières [25], caméra routière [32]). Mais, là encore, les scores fragiles présentés dans cette section sont, au moins en partie, dus à la difficulté du problème. Dans les recherches citées, les images « sans précipitations » sont sélectionnées en dehors des épisodes de précipitations, ce sont donc principalement des images de « beau temps ». Or ces dernières sont faciles à distinguer à l'oeil des images de mauvais temps. Notre problème est plus difficile parce qu'il concerne presque exclusivement des images prises dans ou autour des épisodes de précipitations, sur lesquelles le ciel apparaît menaçant. Contrairement aux études citées, il intègre aussi des images de précipitations nocturnes.
Quand il s'agit de distinguer des degrés dans le « mauvais temps », soit entre des intensités de pluie [26], [32], soit entre la classe nuageux et pluvieux [25], les résultats sont nettement moins bons. Par exemple, dans [25], près de 20 % des images associées à de la pluie sont classées dans la classe nuageux.

3.1.4 Conclusion et épilogue

Sur les problèmes de classification relatifs à un niveau d'enneigement du sol (attribut état du sol) ou à la nature des précipitations (état de l'atmosphère), la situation en terme de qualité d'apprentissage est la même : d'une part, les réseaux de neurones pré-entraînés dominent les modèles initialisés au hasard. D'autre part, la taille du réseau n'est avantageuse que si le réseau a été pré-entraîné et enfin, un palier en validation est vite atteint (en 50 à 75 époques).
Le premier constat est cohérent avec la littérature [64] dans le cas de jeu d'entraînement de petite taille. Les deux autres constats vont aussi dans ce sens : le jeu est trop petit pour un apprentissage autonome (sans pré-entraînement) des phénomènes d'intérêt.
Nous notons aussi que l'apprentissage simultané sur plusieurs attributs peut avoir un effet positif (sur l'attribut état de l'atmosphère). Le nombre de classes joue aussi un rôle positif : les modèles entraînés sur les problèmes complets et évalués sur le problème à deux classes présentent des scores légèrement supérieurs aux modèles directement entraînés sur le problème à deux classes.

Sur les problèmes de classification binaire, les performances quantitatives peuvent être comparées aux scores présentés dans la littérature. Nos scores sont généralement inférieurs.
Ce constat tient sans doute en partie à la taille modeste de notre jeu de données. Mais il s'explique aussi par sa difficulté. Cette difficulté tient au mode de sélection des images, prises autour et pendant des épisodes de mauvais temps. Comparé aux jeux existants, ce mode de sélection réduit la différence d'apparence entre les représentants des différentes classes. Une difficulté supplémentaire est liée à la présence d'images de nuit et d'images de mauvaise qualité, concentrées pendant l'étape de réduction de la redondance (chapitre 2). Les images diificile à traiter représentent d'ailleurs une grande part des cas d'erreur constatés.

Ces résultats, inexploitables en pratique, ont l'intérêt de mettre en lumière les limites des approches existantes en matière de surveillance météorologique : la discrimination réussie entre des images de pluie et de beau temps n'implique pas la capacité à distinguer une averse de chutes de neige ou d'un temps menaçant sur des images venant de webcams indépendantes.

Ces problèmes de classifications seront repris, plus tard, avec des jeux complets (voir table 1, annexe A-II). Sur les images du jeu de test, pour le problème binaire neige au sol/non neige au sol, la justesse équilibrée est alors meilleure de dix points (88 %). Cependant, ces « progrès » sont principalement dus à la nature du jeu de test, constitué essentiellement d'images de jour. Les fausses détections restent nombreuses. Les performances des ces nouveaux modèles seront mises en perspective au chapitre 5.

3.2 Méthodes d'apprentissage par paires

Dans cette partie, nous commençons par définir un problème d'apprentissage des préférences pour la visibilité (section 3.2.1), en décrivant les jeux d'apprentissage et les métriques utilisées. Nous précisons ensuite (section 3.2.2) les aspects communs aux deux approches qui seront tentées : la prédiction par paire (section 3.2.3) et la prédiction par image (section 3.2.4). Les sections suivantes précisent les caractéristiques propres à chacune de ces modalités.

3.2.1 Définition du problème

3.2.1.1 Jeux d'apprentissage. Dédoublement du jeu de validation et mélange des sources.

	jeu	séquences	images	comparaisons strictes		incomparabilités		équivalences	
				graphe	arêtes	graphe	arêtes	graphe	arêtes
AMOSvv	$TRAIN$	320 (84%)	9.850	$\mathcal{G}_0^v$	142.311	$\mathcal{U}_0^v$	34.428	$\mathcal{E}_0^v$	3.144
	VAL_{same}	32	851	$\mathcal{G}_{vals}^v$	4.813	$\mathcal{U}_{vals}^v$	2.842	$\mathcal{E}_{vals}^v$	324
	VAL_{indep}	58	1.333	$\mathcal{G}_{vali}^v$	6.184	$\mathcal{U}_{vali}^v$	5.299	$\mathcal{E}_{vali}^v$	502
	$TEST$	166 (17 %)	2.620	$\mathcal{G}_{test}^v$	15.017	$\mathcal{U}_{test}^v$	9.466	$\mathcal{E}_{test}^v$	963

TABLE 3.5 – Description d'AMOSvv, jeux de base pour l'apprentissage par paires des comparaisons relatives au paramètre visibilité. Entre parenthèses figurent les proportions de séquences tirées des archives AMOS. Le jeu de validation VAL_{same} contient des images venant des caméras du jeu d'entraînement ($TRAIN$). Les jeux VAL_{indep} et $TEST$ sont formés à partir de caméras indépendantes.

La constitution des jeux d'entraînement, de validation et de test apparaît dans la table 3.5. On trouvera en annexe une description plus détaillée (tables de l'annexe B).
Le partitionnement a été réalisé de manière à mettre en évidence les performances en généralisation. Trois formes de généralisation peuvent être distinguées. Les deux premières sont la généralisation sur de nouvelles images venant des mêmes caméras (généralisation dite "faible") et la généralisation sur des images venant de caméras indépendantes (généralisation "forte"). Comme il s'agit pour nous de traiter des données d'opportunité venant de webcams quelconques, ce sont les performances en généralisation forte qui nous intéressent le plus. Néanmoins, pour les premières expériences, nous avons souhaité contrôler l'écart entre les performances en généralisation faible et forte.
Pour cette raison, le jeu de validation d'AMOSvv a été dédoublé : VAL_{same} contient des images originales prises par des caméras du jeu d'entraînement, tandis que VAL_{indep} ne contient que des images prises par des caméras indépendantes.
Une troisième forme de généralisation peut être définie. C'est la capacité d'un modèle entraîné sur

des comparaisons grossières (c'est à dire des paires d'images pour lesquelles la différence de visibilité est très nette) à prédire correctement des comparaisons plus fines. Puisque nous voulons nous concentrer sur le problème de la généralisation forte, il faut s'assurer que les jeux de validation et de test ne présentent pas de comparaisons sensiblement plus fines que le jeu d'entraînement. Or les séquences issues des caméras des DIRs, que nous avons nous-même échantillonnées, décrivent les phénomènes d'intérêt avec une meilleure résolution que celles d'AMOS, du fait d'un pas de temps plus court. Cela se traduit d'ailleurs par des chaînes plus longues (table 2.3) alors même que la période d'acquisition ne s'étend que sur deux à trois jours. Nous avons donc fait en sorte que chacun des trois jeux comprenne des séquences des DIRs et des séquences d'AMOS (voir annexe B).
Par contre, seul le jeu de test contient des séquences d'infoclimat. Pour ces dernières, les chaînes sont plus courtes que sur AMOS et elles portent principalement sur des scènes non routières (voir table A-4). Ces séquences représenteront donc, a priori, la partie la plus difficile de notre jeu de test.
Pour les trois jeux, la construction des graphes s'est faite suivant procédé similaire, en trois étapes. Nous procédons d'abord à l'union disjointe des graphes associés à chaque séquence (première étape). Ensuite sont ajoutées les comparaisons consécutives (deuxième étape) et les comparaisons obtenues par la règle 2 (troisième étape) qui n'étaient pas déjà dans ces graphes [5]. Après chaque étape, les quelques contradictions (2-cycles) dues aux erreurs d'annotation sont contrôlées et supprimées. La seule différence de traitement tient à une dernière opération de clôture transitive appliquée au graphe $\mathcal{G}_0^v$.

3.2.1.2 Métriques pour la restitution d'une relation d'ordre

Les performances en généralisation seront d'abord mesurées dans le cas où les prédictions sont les comparaisons strictes du problème à deux classes ($\mathcal{Y} = \{\dot{\prec}, \dot{\succ}\}$). Nous utilisons alors la justesse :

$$\text{Justesse} = \frac{C}{C+D} \tag{3.2}$$

Où C est le nombre de prédictions correctes et D, le nombre de discordances (prédiction de $\dot{\prec}$ au lieu de $\dot{\succ}$ ou de $\dot{\succ}$ au lieu de $\dot{\prec}$).

Lorsque le modèle prédit aussi des incomparabilités, la situation se complique : il faut tenir compte des paires strictement ordonnées par l'annotateur qui sont prédites incomparables par le modèle ; nous parlerons de rejet (noté R sur la figure 3.2). Nous tiendrons aussi compte des « fautes de prudence », c'est à dire des paires strictement ordonnées par le modèle mais considérées comme incomparables par l'annotateur (F), et des incomparabilités correctement prédites (I).

Les trois types d'erreur possibles, discordance, rejet et faute de prudence, sont pris en compte par les trois métriques définies figure 3.2. La Correcteness, introduite par Cheng et al. [96], reflète la proportion de concordances parmi les comparaisons strictes non-rejetées. Cette quantité généralise la

5. avec un maximum de 1000 arêtes par séquence, pour ne pas déséquilibrer le jeu

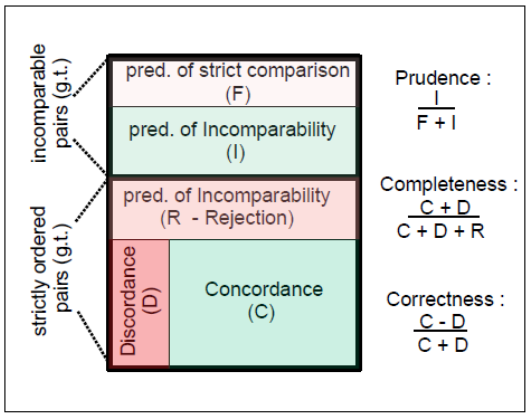


FIGURE 3.2 – Métriques pour l'évaluation d'un modèle qui prédit les trois classes de comparaison.

corrélation de rang de Kendall au cas d'une relation d'ordre partiel [96]. Elle varie de -1 à 1, tandis que les autres scores, Completeness et Correctness, sont compris entre 0 et 1.

3.2.2 Paramétrisation de la descente de gradient stochastique. Augmentation de données

Dans ce paragraphes, nous donnons les caractéristiques communes aux apprentissages par paires avec des réseaux neurones profonds. Comme dans la le première section, les principaux paramètres de la descente de gradient ont été réglés sur des valeurs standard.

3.2.2.1 Formation des batches

Chaque mini-batch comprend 32 paires d'images qui sont sélectionnées au hasard parmi les arêtes des graphes. Lorsque les deux graphes $\mathcal{D}_0^v$ et $\mathcal{U}_0^v$ (resp. $\mathcal{D}_0^v$ et $\mathcal{E}_0^v$) sont utilisés, il faut d'abord sélectionner un graphe. Le graphe $\mathcal{U}_0^v$ (resp. $\mathcal{E}_0^v$) est sélectionné avec une probabilité de $1/3$ contre une probabilité de $2/3$ pour $\mathcal{D}_0^v$. Ces paramètres de l'apprentissage sont nommés « fréquences de présentation ».

Pour tirer des paires d'images à partir du graphe sélectionné, deux modalités ont été testées. Selon la première, la probabilité $p_{i,j}$ de tirage d'une arête (x_i, x_j) est inversement proportionnelle à la moyenne des degrés des images dans le graphe sélectionné, noté $\mathcal{G}$:

$$p_{i,j} \propto \frac{d_{\mathcal{G}}(x_i) + d_{\mathcal{G}}(x_j)}{2} \tag{3.3}$$

où $d_{\mathcal{G}}(x)$ désigne le degré du noeud x dans le graphe $\mathcal{G}$.
De cette façon, on évite la surreprésentation des images qui apparaissent dans un grand nombre de comparaisons.

Augmentation de données	- réflexion par rapport à un axe vertical - perturbations dans l'espace HSV - rotation - transformation perspective - rognage (moins de 30 %) - réduction (ImageNet) - dimensions spatiales des images d'entrée : 384×384 (prédiction par paires) 256×256 (prédiction par image)
Mini-batches	taille : $\mathbf{32}$ paires d'images sélection : équation (3.3) Nombre par époque : 200 (soit 12.800 images)
Optimisation	taux d'apprentissage initial (lr) : $\mathbf{10^{-4}}$ méthodes : - ADAM (prédiction par paires) - ADAM + scheduler (prédiction par images)
Fonctions de coût	prédiction par paires : entropie croisée (CE) prédiction par image : -Hinge Loss -RankNet Loss

TABLE 3.6 – Paramétrisation des apprentissage pour l'apprentissage par paires.

Selon la seconde modalité, on pondère la quantité précédente par la taille de la séquence s associée à l'arête sélectionnée. On a alors :

$$p^s_{i,j} \propto \frac{1}{\sqrt{|X_s|}} \times \frac{d_{\mathcal{G}}(x^s_i) + d_{\mathcal{G}}(x^s_j)}{2} \tag{3.4}$$

où X_s est l'ensemble des images du jeu associées à la séquence s.
De cette façon, on évite la surreprésentation des séquences les plus longues sans donner un poids excessif aux séquences les plus courtes. Ces deux pondérations ont conduit à une légère amélioration des scores, sans qu'il soit possible de les départager. Nous avons donc conservé la plus simple, c'est à dire la première modalité, dans toutes nos expériences.

3.2.2.2 Augmentation de données

Les images ont été stockées dans des répertoires nommés d'après le répertoire AMOS ou la caméra DIR d'origine. Les transformations qui ont pu être faites hors apprentissage ont été appliquées une fois pour toutes : les images sont normalisées et mises au format torch.tensor. La stratégie suivie pour l'augmentation de données a évolué pendant la thèse, mais au sein d'une expérience elle peut être

considérée comme identique pour tous les apprentissages, sauf mention contraire.
De façon générale, nous avons limité le rognage aléatoire de façon a conserver au moins 70 % de l'image. Ainsi, il reste assez d'information pour ordonner correctement les images. Les autres transformations sont listées dans la table 3.6. En particulier, nous avons fait usage de transformations perspective et de rotations, dans des proportions « raisonnables ». La façon dont ces transformations sont paramétrées est précisée dans les codes de l'annexe numérique.
Notons que, dans les expériences présentées dans cette partie, les rognages, réflexions, rotations et transformations perspectives appliquées aux deux images d'une même paire sont identiques. Ainsi, l'alignement entre les deux images est conservé après transformation.

3.2.2.3 Méthode d'optimisation

La descente de gradient est réalisée suivant la méthode ADAM [95]. Pour la prédiction par paires, cette méthode s'est montrée au moins aussi efficace qu'une SGD standard avec décroissance régulière du pas d'apprentissage. Pour l'apprentissage de fonctions d'ordre, la méthode ADAM a été combinée à une réduction régulière du taux d'apprentissage (learning rate) global [6] Cette modalité n'est que rarement employée dans la littérature, mais elle s'est montrée efficace pour l'apprentissage par paires.

3.2.3 Prédiction par paire

Pour aborder la prédiction par paire des relations d'ordre, la première couche d'un réseau de neurones profond est modifiée de façon à prendre en entrée une paire d'images concaténées, suivant l'approche définie dans [97]. Les images originelles comportant chacune trois canaux, l'entrée en comporte six (figure 3.3).
Le réseau est entraîné sur une tâche de classification. Les cibles sont les trois classes de relation : $\dot{\succ}$, $\dot{\prec}$ et $\dot{\perp}$ (ou seulement $\dot{\succ}$ et $\dot{\prec}$). Le nombre de neurones en sortie du réseau est ramené à trois (ou deux).

Trois catégories d'architectures ont été testées : ResNet[57], ResNext[98] et VGG[56]. Comme la structure de la première couche est modifiée, il n'est pas possible d'utiliser un réseau pré-entraîné « sur-étagère ». Enfin, dans toutes les expériences, la fonction de coût est l'entropie croisée, appliquée en sortie de la couche softmax (voir chapitre 1).

Notons que la relation prédite par la classifieur entre les images d'une séquence, n'est pas, a priori, une relation d'ordre. Pour restituer un ordre partiel (qu'on espère proche de celui qui a été annoté), plusieurs méthodes existent [99],[96],[100]. Nous reviendrons sur cet aspect au chapitre 4.

6. Ce paramètre représente la valeur seuil sous laquelle les taux d'apprentissage individuels -un par poids- sont contraints d'évoluer.

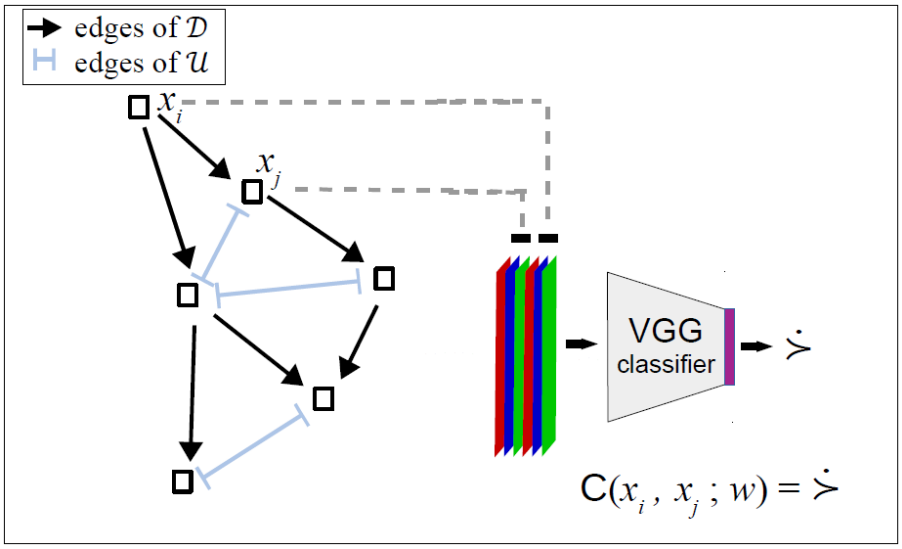


FIGURE 3.3 – Prédiction par paires d'une relation d'ordre entre les deux images par un classifieur.

3.2.4 Prédiction par image (fonction de rang)

La seconde méthode consiste à prédire un indice réel pour chaque image (figure 3.4). Pour chaque paire d'images vue pendant l'entraînement, le modèle prédit deux indices et il est pénalisé si l'ordre dans lequel les indices sont rangés n'est pas celui de l'annotation. L'apprentissage est réalisé par des « réseaux siamois » (voir figure 3.4 et section 1.2.1.5).

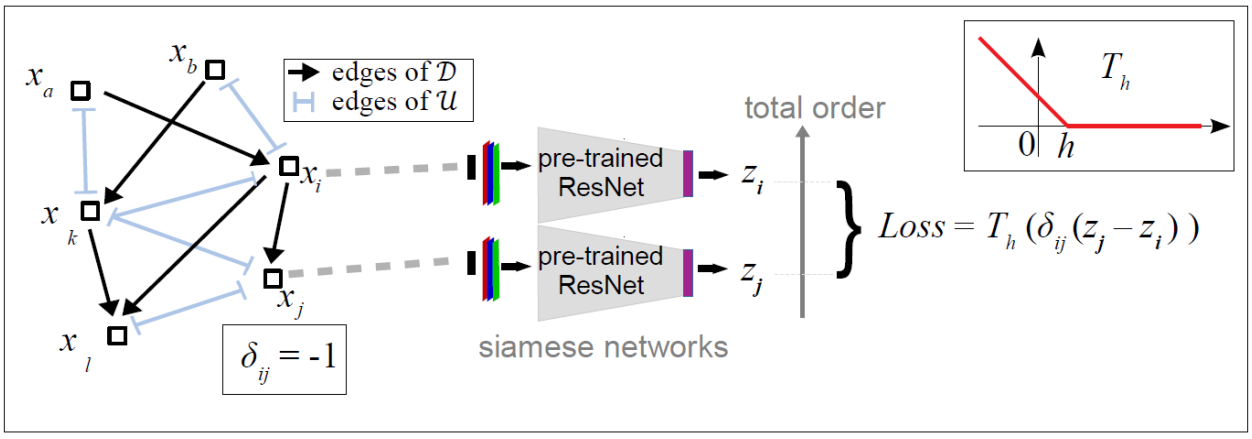


FIGURE 3.4 – Prédiction par image avec une fonction d'ordre. Lors de l'entraînement, les réseaux siamois sont pénalisés lorsque les sorties sont rangées dans le mauvais ordre (ici, par une Hinge Loss).

La couche de sortie est réduite à un neurone de type perceptron pour passer d'un classifieur standard à un modèle de régression. La couche *softmax* est alors supprimée. Nous avons testé les mêmes catégories d'architectures que pour la prédiction par paire (ResNet, ResNext, VGG) ainsi qu'un DenseNet [101]. Ici, les premières couches de neurones ne sont pas modifiées et il est possible d'utiliser des versions pré-entraînées de ces réseaux.

Pour pénaliser ces modèles, nous reprenons les fonctions de coût adaptées [102] : la Hinge Loss et la RankNet loss [103]. La Hinge Loss s'écrit :

$$\mathcal{L}^h(\delta_{i,j}, z_i, z_j) = [h - \delta_{i,j}(z_j - z_i)]_- \tag{3.5}$$

Où $z_i = f(x_i; w)$, $z_j = f(x_j; w)$; $f(.; w)$ étant la fonction de transfert du réseau. $[.]_-$ est la fonction partie négative et $\delta_{i,j} = 1$ si $x_i \succ x_j$ ($\delta_{i,j} = -1$ si $x_j \prec x_i$) et h est un paramètre qui permet d'éviter la solution triviale $f \equiv 0$ (on a fixé $h = 0.1$ dans nos expériences).
La RankNet loss [103] est une alternative à la Hinge Loss, plus lisse. Avec les mêmes notations, elle s'écrit :

$$\mathcal{L}^r(\delta_{i,j}, z_i, z_j) = ln(1 + e^{-\sigma \times \delta_{i,j} \times (z_i - z_j)}) \tag{3.6}$$

Où σ est un paramètre d'échelle, pris égal à 1 dans nos expériences.

Lorsque l'apprentissage a lieu sur les arêtes des graphes $\mathcal{D}_0^v$ (comparaisons strictes) et de $\mathcal{E}_0^v$ (équivalences), on utilise le cas d'égalité correspondant à la Hinge Loss :

$$\mathcal{L}^h_{eq}(z_i, z_j) = |z_i - z_j| \tag{3.7}$$

Ou bien celui associé à la RankNet Loss :

$$\mathcal{L}^r_{eq}(z_i, z_j) = ln(1 + ch(\sigma \times (z_i - z_j)) \tag{3.8}$$

L'apprentissage d'une fonction de rang a plusieurs avantages sur la prédiction par paires. D'une part, des réseaux pré-entraînés sont disponibles. D'autre part, une fois appris, un tel réseau est facile a déployer. Il peut être appliqué à un flux d'images sans qu'il soit nécessaire d'en stocker aucune pour ordonner la série.
Notons aussi qu'une fonction de rang induit d'emblée une relation d'ordre total sur les images. Une telle fonction répond ainsi à un objectif plus ambitieux -trier une séquence- que la prédiction par paire. Mais en l'état, les incomparabilités de notre jeu de données ne peuvent ni être apprise, ni être restituées. En particulier, une fonction de rang à valeurs réelles ne peut pas rendre compte de la qualité de l'information contenue dans l'image.

3.3 Apprentissage par paires sur AMOSvv

Dans cette partie, nous appliquons les méthodes décrites dans la partie précédente au cas de la visibilité.
Sur le jeu AMOSvv, un modèle initialisé au hasard peut être entraîné à la prédiction par paire efficacement, dans le sens où les performances en généralisation croissent encore après que l'ensemble des images ait été parcouru plus d'une centaine de fois. Nous donnons plus de corps à cette affirmation tout en montrant l'impact de la seconde étape d'annotation (section 3.3.1) sur l'apprentissage d'une architecture de type VGG.
Une autre question importante porte sur l'apprentissage de la relation d'incomparabilité (3.3.2). Celle-ci peut être simplement prise en compte en ajoutant une troisième classe. Les résultats sur le jeu de validation suggèrent un effet positif avec une métrique adaptée au problème à deux classes. Cet effet est confirmé sur le jeu test avec une métrique qui tient compte des incomparabilités.
La prédiction par paires est ensuite comparée à la prédiction par image (3.3.3). Suivant cette dernière modalité, le réseau peut être pré-entraîné. Malgré cela, les performances des meilleures fonctions de rang (ResNet50) sont moins bonnes qu'avec un modèle entraîné à la prédiction par paires.
Nous ne présentons pas ici l'étape de sélection du modèle. Celle-ci été réalisé en amont sur le jeu de validation et peut être trouvée en annexe (annexe D.1). Elle a conduit à sélectionner les architectures de type VGG (VGG11 et VGG13) pour la comparaison par paires, et des architectures de type ResNet (ResNet50) pour l'implémentation des fonctions de rang. Ce sont principalement ces réseaux qui seront évoqués dans la suite du mémoire.

3.3.1 Intérêt des paires non consécutives pour la prédiction par paire

Nous avons vérifié que les paires non-consécutives supplémentaires obtenues sur le paramètre visibilité avaient un impact sur l'apprentissage du classifieur, et plus précisément, sur ses performances en généralisation.
Pour cette première expérience, nous avons suivi les performances sur deux jeux de validation VAL_{same} , contenant des images de scènes déjà vues à l'entraînement, et VAL_{indep} contenant des images de scènes indépendantes (voir table 3.5). Les entraînements ont lieu sur le problème à deux classes ($\dot{\prec}$ et $\dot{\succ}$).
Trois configurations sont comparées : dans la configuration A, seules les comparaisons consécutives et celles qui en découlent par la règle de clôture transitive (règle 3) sont utilisées à l'entraînement. Dans la configuration B, toutes les comparaisons obtenues par application des règles 2 et 3 sont utilisées. Dans la configuration C., on utilise toutes les paires disponibles après annotation des paires non consécutives. Les effectifs des graphes correspondants sont précisés figure 3.5.
Pour éviter les fluctuations dues à l'initialisation, les modèles entraînés (VGG11) sont initialisés avec les mêmes poids.
Comme attendu, les performances en généralisation sur les images de scènes indépendantes, sont net-

tement moins élevées (figure 3.5. a-d). L'évolution des valeurs prises par la fonction de coût sur les jeux de validation au cours de l'entraînement (« courbes d'apprentissage ») traduisent la richesse du jeu d'apprentissage : sans les comparaisons supplémentaires fournies par la deuxième étape, le minimum est atteint avant les vingt premières époques (figures 3.5.a,b). Au contraire, l'apprentissage se prolonge au-delà de la cinquantième époque lorsque toutes les comparaisons disponibles sont prises en compte. Noter que l'influence des comparaisons d'images non-consécutives est plus marquée sur les performances en généralisation (courbes bleues, figures 3.5.c,d).

3.3.2 Apprendre la relation d'incomparabilité ?

Pouvoir estimer avec quelle précision un paramètre est connu est l'un de nos objectifs. Cette précision varie avec la qualité de l'image et avec la scène (voir annexe I-A). Elle est encodée dans la relation d'incomparabilité. Par exemple, une image bruitée est plus souvent considérée comme incomparable. Pouvoir restituer cette relation d'incomparabilité, c'est donc se donner un moyen de répondre à un objectif de la thèse.

ENCADRÉ 3.1 – Mécanisme d'abstention par épaississement pour la prédiction par paires

Notons $x_{i,j}$ la concaténation des images x_i et x_j, $p^2_{\dot\prec}(x_{ij})$ et $p^2_{\dot\succ}(x_{ij})$ les composantes du vecteur de scores associées à x_{ij} en sortie de la couche softmax du classifieur à deux classes. On définit alors la classe prédite par abstention de niveau λ par :

$$C^{2\to3}_{\lambda}(x_i, x_j) = \underset{c\in\{\dot\prec,\dot\succ,\dot\perp\}}{\arg\max}\, p^2_c(x_{ij}) \tag{3.9}$$

où :

$$p^2_{\dot\perp}(x_{ij}) \triangleq \lambda$$

De cette façon, seules les comparaisons strictes prédites avec un score supérieur à λ sont conservées, les autres sont rejetées.
Ce mécanisme s'étend à un classifieur à trois classes en posant :

$$C^3_{\lambda}(x_i, x_j) = \underset{c\in\{\dot\prec,\dot\succ,\dot\perp\}}{\arg\max}\, p^3_{\lambda,c}(x_ij) \tag{3.10}$$

où :

$$p^3_{\lambda,\dot\prec}(x_{ij}) = p^3_{\dot\prec}(x_{ij}) \qquad p^3_{\lambda,\dot\succ}(x_{ij}) = p^3_{\dot\succ}(x) \qquad p^3_{\lambda,\dot\perp}(x_ij) = p^3_{\dot\perp}(x_{ij}) + \lambda$$

Pour $\lambda = 0$, les prédictions du classifieur à trois classes sont donc inchangées, et pour $\lambda \leq -1$, la prédiction d'incomparabilité est désactivée. Pour $\lambda > 0$, les comparaisons associées aux scores les plus faibles sont transformées en incomparabilité. La conséquence observée en pratique, c'est que la résolution du classifieur diminue. Pour désigner ce mécanisme d'abstention, nous parlerons donc d'« épaississement ».

Configuration	Annotation	Archi.	$\mathcal{G}$	$\mathcal{U}$	$\mathcal{E}$
A	arêtes consécutive + règle 3	VGG11	15.684	11.011	2.661
B	arêtes consécutives + règle 2 + règle 3		88.009	10.800	2.635
C	toutes les arêtes disponibles		142.311	34.428	3.144

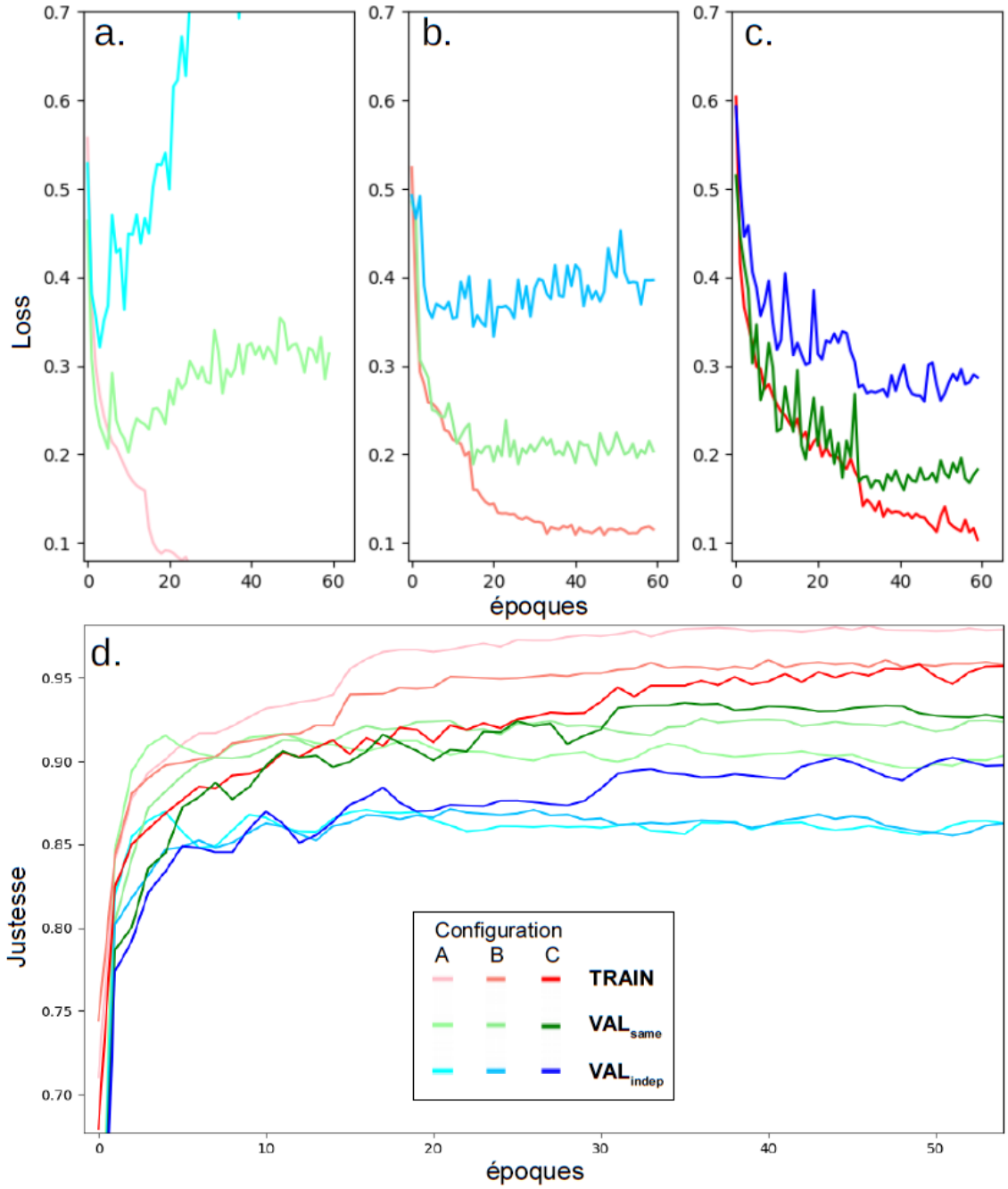


FIGURE 3.5 – a.b.c. Courbes d'apprentissage dans les configurations A,B,C définies dans la table (en haut). Dans les configurations A et B, le nombre d'arêtes est restreint. d. : Evolution des justesses (équation (5.1)) dans les trois configurations.

Il existe plusieurs façons de prédire l'incomparabilité. On peut par exemple introduire un mécanisme d'abstention (voir [104] pour des SVM, et [105], [106] pour le deep learning), qu'on règle après l'apprentissage ou qu'on apprend en utilisant de l'erreur de prédiction. Mais, dans notre cas, un grand

nombre de paires d'images incomparables est disponible pour l'apprentissage. Une alternative consiste donc à entraîner le modèle sur une tâche de classification à trois classes ($\dot{\prec}$; $\dot{\succ}$; $\dot{\perp}$).
Nous avons cherché à comparer ces deux approches à travers une expérience simple qui implique deux classifieurs (VGG11) appris suivant les modalités définies aux sections 3.2.2 et 3.2.3. Le premier est appris sur les comparaisons strictes et les paires incomparables. Le deuxième est appris sur les comparaisons strictes et associé à un mécanisme d'abstention (voir encadré 3.1).

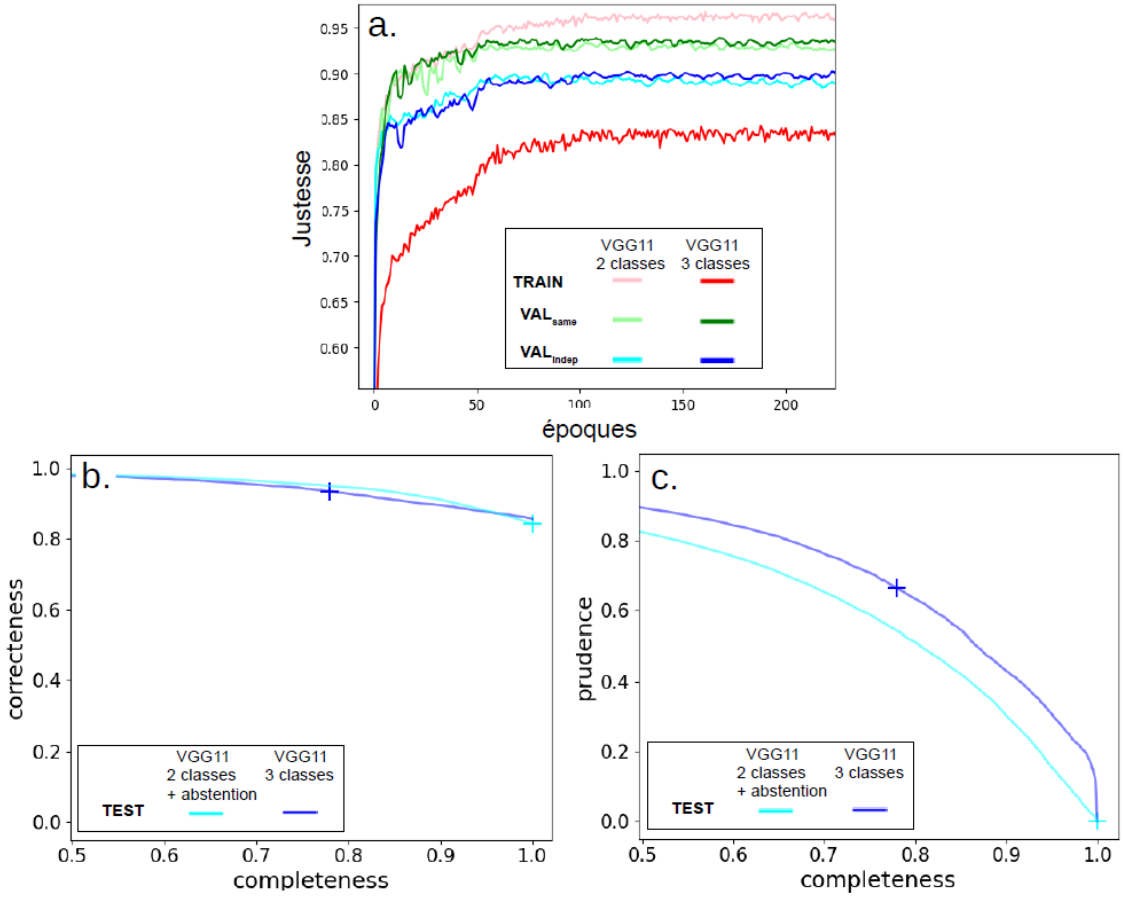


FIGURE 3.6 – Comparaison entre deux classifieurs VGG11 entraînés à la prédiction par paire. Le premier est entraîné sur $\mathcal{G}_0^v$ à classer les paires d'images parmi $\{\dot{\succ}, \dot{\prec}\}$. Le second est entraîné sur $\mathcal{G}_0^v$ et $\mathcal{U}_0^v$ à classer parmi $\{\dot{\succ}, \dot{\prec}, \dot{\perp}\}$. En a., on a désactivé la prédiction d'incomparabilité pour comparer les justesses sur VAL_{same} et VAL_{indep}. En c. et d., on compare sur le jeu de test les classifieurs après épaississement (voir encadré 3.1). Les courbes sont obtenues en faisant varier le paramètre λ.

Un premier élément de comparaison nous est donné par les performances en validation (justesse sur les comparaisons strictes). Pour comparer les courbes d'apprentissage, le modèle à trois classes est ramené à un modèle à deux classes en désactivant la troisième composante (voir encadré 3.1, dernier paragraphe).
Sur la figure 3.6.a, le modèle à trois classes atteint une justesse légèrement supérieure à celle du modèle à deux classes. Nous avons pu vérifier avec d'autres architectures VGG que les performances en validation n'étaient jamais inférieures lorsque le modèle était entraîné sur trois classes plutôt que sur

deux.

Les compromis entre Completeness et Correctness obtenus en faisant varier le seuil λ sont sensiblement les mêmes pour les deux approches (figure 4.c). Par contre, pour une même Completeness, la Prudence est de dix points supérieure (figure 4.d) lorsque les paires incomparables ont été apprises. Pour restituer les relations d'ordre contenues dans l'annotation, il est donc plus intéressant d'utiliser le classifieur appris sur les trois classes qu'un simple mécanisme d'abstention.

3.3.3 Comparaison entre prédiction par paire et prédiction par image

Nous avons comparé les performances de la prédiction par paires avec celles d'une prédiction par image. Les fonctions de rang ont été implementées sur différents modèles et entraînées suivant la procédure décrite section 3.2.4. Les meilleurs performances ont été atteintes par un ResNet50 pré-entraîné sur ImageNet (comme pour la classification, section 3.1). Toutes les étapes relatives à la sélection de modèle sont présentées dans l'annexe D.1.

models	Justesse sur VAL_{same}	Justesse sur VAL_{indep}	Justesse sur $TEST$
vv_sl_d0.0 (ResNet50)	0.924	0.910	0.898
vv_sl_d0.1 (ResNet50)	0.910	0.907	0.898
vv_sl_de00.1 (ResNet50)	0.915	0.907	0.897
vv_pl_VGG13_du0.0	0.934	0.913	0.929
vv_pl_VGG11_du0.0	0.941	0.909	0.931
Transmission mean [107]	–	–	0.73
ranking SVM [89] (linear)	–	–	0.79
ranking SVM [89] (polynomial)	–	–	0.82

TABLE 3.7 – Comparaisons entre prédiction par image (fonctions de rang notées vv_sl_xxx, lignes 2-4), prédiction par paire (classifieurs notés vv_pl_xxx, lignes 5-6), et des méthodes pré-existantes (lignes 7-9) présentées section 3.3.4.

Pour comparer les deux formes d'apprentissage, nous indiquons dans la table 3.7 les performances associées aux modèles sélectionnés sur le jeu de validation. Ces modèles sont nommés suivant la nomenclature définie dans l'annexe C.
Les meilleures performances en validation (sur VAL_{indep}) sont du même ordre, mais les courbes d'apprentissage des fonctions de rang (vv_sl_d0.0, vv_sl_d0.1 et vv_sl_de00.0) atteignent leur maximum avant cinquante époques, où la justesse en validation fluctue beaucoup, alors que les classifieurs (vv_pl_vgg13_du0.0 et vv_pl_vgg11_du0.0) apprennent plus longuement pour atteindre un palier plus haut en validation (figure 5.a).
L'avantage des classifieurs est nettement confirmé sur les paires strictement ordonnées du jeu de test d'AMOSvv (voir table 3.7). Pour vérifier que la plus-value s'étend aux paires incomparables, il faut pouvoir prédire un ordre partiel à partir d'une fonction de rang à valeurs réelles. Nous le faisons en sui-

vant une procédure proposée dans [37], par « épaississement » des fonctions de rang (voir encadré 3.2).

ENCADRÉ 3.2 – Epaississement des fonctions de rang

Soit $f(.;w)$ une fonction de rang à valeurs réelles entraînée sur des paires strictement ordonnées (et éventuellement, sur des paires d'images équivalentes). Une procédure simple permet de transformer l'ordre total induit par f sur les images en une relation d'ordre partiel :
Pour tout $\lambda \geq 0$, on pose :

$$f_\lambda(x;w) = [f(x;w) - \lambda, f(x;w) + \lambda] \tag{3.11}$$

Une telle fonction « épaissie » induit un semi-ordre [108] :

$$x_a \prec_{pred} x_b \Leftrightarrow f(x_a;w) + \lambda < f(x_b;w) - \lambda \tag{3.12}$$

Les incomparabilités prédites correspondent alors à des chevauchements d'intervalles.

En comparant les classifieurs épaissis aux fonctions de rang épaissies (encadré 3.2), il apparaît qu'à même Completeness, les premiers ont une Correctness et une Prudence plus élevées (figure 3.7.b et c). Sur les figure 3.7.b-c, nous illustrons aussi l'effet d'un post-traitement très simple sur les performances de la prédiction par paire (courbes en pointillé). Ce post-traitement consiste à changer l'ordre des images de la paire dans la concaténation et à corriger les incohérences à travers une symétrisation. En reprenant les notations de l'encadré 3.1, cette symétrisation est définie par :

$$C_\lambda^{3,sym}(x_i,x_j) = \underset{c\in\{\dot{\prec},\dot{\succ},\dot{\perp}\}}{\arg\max} \frac{p^3_{\lambda,c}(x_{ij}) + p^3_{\lambda,c}(x_{ji})}{2} \tag{3.13}$$

La claire avance de la prédiction par paire n'est pas facile à interpréter : plusieurs mécanismes peuvent y avoir contribué.
D'abord, la tâche du classifieur est très proche de celle demandée à l'annotateur. Une partie des annotations n'est que difficilement reproductible à partir d'une simple image, en particulier la relation d'incomparabilité. Prenons par exemple deux images, chacune parasitée par un flocon collé à la lentille. Si les flocons masquent la même partie de l'image, la comparaison peut encore être effectuée en exploitant l'autre partie. Si les flocons masquent des parties complémentaires, les images seront déclarées incomparables. Ce cas est très rare, mais il illustre une des limites de la prédiction par image. Cependant, ce type de cas ne peut pas expliquer la plus-value sur le diagramme Correctness-Completeness.
D'autres mécanismes peuvent être invoqués. Construire une échelle ordinale peut être plus difficile que comparer des images alignées. Il faudrait alors un jeu plus large pour apprendre une fonction d'ordre. Par ailleurs l'étape de concaténation peut jouer un rôle analogue à une technique d'augmentation de données : ce n'est plus un jeu de 10^4 images, mais un jeu de 10^5 paires sur lequel l'apprentissage est conduit.

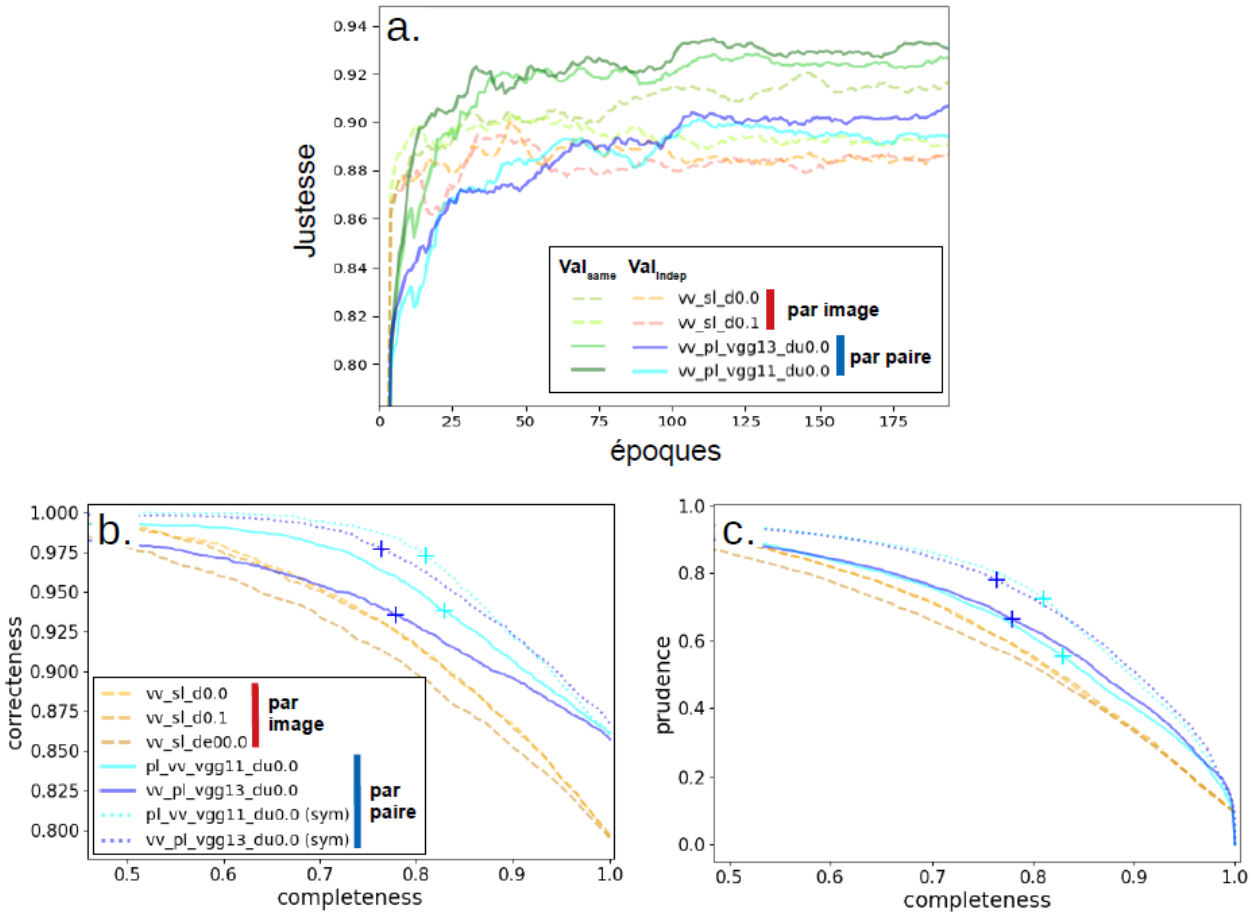


FIGURE 3.7 – Comparaison entre les meilleurs classifieurs (prédiction par paire) et les meilleures fonctions de rang (prédiction par image). a.Courbes d'apprentissages lissées par moyenne mobile (5 époques consécutives). b-c. les fonctions de rang sont épaissies pour obtenir des relations d'incomparabilité (encadré 3.2) et comparées aux classifieurs épaissis. Les meilleures performances sont obtenues par symétrisation des classifieurs (equation (3.13)).

3.3.4 Comparaison avec d'autres méthodes

Nous avons cherché à comparer les approches par deep learning à des approches déjà existantes. Pour ce faire, nous utilisons les points de comparaisons proposés dans You et al. [37].
Les premiers points de comparaison sont construits à partir de la carte de transmission estimée par Tang et al [107]. Cette carte de transmission quantifie la transparence de l'atmosphère pour chaque pixel de l'image. Elle correspond au facteur $e^{-3d/v}$ de l'équation A-1 (loi de Koschmieder, annexe A, équation (A-1)), où v est la visiblité, d, la distance entre la caméra et l'objet couvert par le pixel considéré. Trois fonctions de rang sont construites de la carte de transmission : la première est donnée par la valeur maximum sur l'ensemble des pixels de la carte, la seconde par la médiane et la troisième par la moyenne. Sur le jeu de test, les justesses associées à ces fonctions de rang vont de 0.70 à 0.73.

L'approche par ranking SVM [89] a aussi été testée avec deux noyaux classiques (lignes 8 -sans noyau- et 9 -noyau polynomial). Pour entraîner le SVM, nous avons utilisé les « Tang features » de taille 832 proposés par dans [37]. Bien que portant sur des comparaisons intra-scène, a priori plus

simples, ces modèles donnent des justesses comparables à celles obtenus dans [37]. Comparées aux réseaux de neurones profonds, ces justesses sont inférieures de dix points.
Un tel écart dans les performances est cohérent avec l'état de l'art (voir par exemple You et. al [37] pour des images singulières).

3.3.5 Une première application

Pour terminer ce chapitre, nous donnons une première application de la prédiction par paire à l'exploitation de données d'opportunité. Nous cherchons ici à indiquer quelles seraient les performances objectives d'un classifieur sur une tâche de détection des mauvaises conditions de visibilité.
Nous nous plaçons dans le cas où pour chacune des caméras à traiter il est possible d'échantillonner et d'archiver [7] quelques images de référence associées à une valeur seuil de visibilité donnée.

La procédure de détection est la suivante : chaque nouvelle image est comparée avec l'ensemble des images de références $\mathcal{R}_t$, associées au seuil d'intérêt t. La comparaison est faite par un classifieur symétrisé (vv_pl_vgg13_du0.0). On obtient alors une liste de prédictions de taille $|R_t|$, à valeurs dans $\{\dot{\prec}; \dot{\succ}; \dot{\perp}\}$.
On décide $v_{pred}(x) < t$ s'il y a plus de $\dot{\prec}$ que de $\dot{\succ}$ dans la liste, $v_{pred}(x) > t$ s'il y en a moins et la comparaison est rejetée dans les cas d'égalité.

Cette procédure a été suivie sur les images de jour (9h – 16h) des séries d'images du jeu TENEBRE_1218 (octobre 2012 – mars 2013 et octobre 2017 – mars 2018) pour des seuils critiques couramment utilisés pour la météorologie routière ou pour la prévision aérienne (200 m, 250 m, 500 m, 1000 m, 1609 m et 5000 m). Le seuil de 250 m permet de comparer indirectement notre approche à celle de Pagani et al. [36].

Pour chaque seuil t, on définit l'ensemble d'images $X_{t,m}$ dont les visibilités sont comprises entre $t(1-m)$ et $t(1+m)$. De cet ensemble on échantillonne au hasard dix images de référence ($|R_t| = 10$) pour une marge relative $m = 20\%$. On applique la procédure décrite au-dessus à toutes les images de la séquence.
La table 3.8 présente les résultats pour le seuil 250 m. Les autres résultats sont consignés dans l'annexe D.

Sur les quatre premières scènes de la table 3.8, les scores sont satisfaisants. Après un rapide passage en revue des cas d'erreur, il apparaît que les non détections ne diffèrent pas visuellement des « vrais positifs », et relèvent plutôt des discordances déjà observées entre annotations manuelle et instrumentale (voir section 2.3.4.3). Pour ces quatre caméras, la marge de progression est donc très petite.
Mais pour les quatre dernières caméras, les précisions sont plus faibles. Pour neige_nancy, les fausses détections tiennent principalement à une séries d'images affectées par un film d'eau sur la caméra qui n'a pas été « interprété » comme tel. Le passage en revue des fausses détections sur cette scène a aussi

7. le stockage des images prises en extérieur pose un problème d'ordre juridique (link)

Sequence id	cas observés	F1-score	recall	precision	rejection
nancy1	38	0.66	0.82	0.78	0/1
nancy2	38	0.63	0.86	0.7	1/29
roissy	47	0.71	0.96	0.74	9/11
entzheim3	53	0.67	0.83	0.77	0/7
entzheim1	53	0.35	0.87	0.37	0/9
neige_nancy	34	0.41	0.7	0.5	4/8
parc_entzheim	53	0.39	0.81	0.39	0/3
portail_entzheim	53	0.14	0.84	0.15	2/90
“2.5-km/2.5-km” [36]	-	0.65	0.7	0.61	0/0
“2.5-km/7.5-km” [36]	-	0.19	0.23	0.16	0/0

TABLE 3.8 – Résultats de la prédiction par paire (classifeur vv_pl_vvgg13_du0.0) sur une tâche de détection d'une visibilité inférieure à 250 m. Pour toutes les scènes de TENEBRE, les scores sont calculés sur 12 mois (automne-hiver 2012 et autmone-hiver 2017). Les séries associées aux sites Markstein et Dorans ont été exclues, faute de disposer de données fiables sur toute la période. Les deux dernières lignes, tirées de [36], sont obtenues sur des ensembles de caméras autoroutières. La dernière colonne indique le nombre d'événements rejetés parmi le nombre total de rejets.

révélé un problème d'images parasites provenant d'autres caméras intercalées dans la sériefootnote-D'autres séries présentent ce problème, sans que le phénomène ait joué sur les scores au seuil de 250 m. Les erreurs dus à ces images parasites n'ont pas été prises en compte.
Pour entzheim1, les fausses détections semblent liées à la pousse des arbres d'une haie entre les deux années qui réhausse la ligne d'horizon. Pour parc_entzheim, la grande majorité des fausses détections a lieu lorsque la neige couvre entièrement le sol. Enfin, portail_entzheim est une caméra très atypique par rapport aux images du jeu AMOSvv (voir figure 2.4) et sur laquelle les fausses détections n'ont pas un profil clair. Les nombreux rejets sont cohérents avec le peu d'information contenu dans les images sur la visibilité (scène sans second plan).

L'étude de Pagani et al. [36] nous fournit une point de comparaison. Ces auteurs ont entraîné un perceptron multicouche à détecter des brouillards associés à une visibilité inférieure au seuil de 250 m. Les jeux sont constitués d'images de jour jointes à des mesures distantes. Ils considèrent deux jeux d'images. Les images du premier jeu sont associées à une mesure relativement précise : les capteurs utilisés sont situés à moins de 2,5 km des caméras. Le second jeu intègre en plus des images issues de caméras plus distantes des capteurs (jusqu'à 7,5 km).
Rappelons les résultats obtenus par ces auteurs lorsque le perceptron est entraîné sur le premier jeu. Evaluées sur des images indépendantes venant des mêmes caméras (ligne 10), les performances en généralisation faibles sont plus de dix points inférieures à celles obtenues par notre modèle sur les quatre premières scènes -qui correspondent aussi aux scènes les moins atypiques. Evaluées sur l'ensemble des caméras disponibles (ligne 11), les performances du perceptron multicouche sont globalement inférieures à celles de la prédiction par paire.
Cependant, la comparaison avec l'expérience de Pagani et al. [36] est limitée : les caméras diffèrent,

la représentativité des mesures est discutable, leur période d'acquisition est plus longue et ces auteurs intègrent les images prises en été.
L'exercice aura néanmoins permis de s'assurer que les performances obtenues par nos classifieurs sur des scènes très différentes de celles vues à l'entraînement sont du même ordre que les performances de l'état de l'art sur des scènes plus homogènes et, pour partie, déjà vues à l'entraînement.

3.4 Conclusion

Dans ce chapitre, nous avons d'abord présenté des résultats sur des problèmes de classification définis à partir de l'annotation par image. Nous nous sommes concentré sur des problèmes relatifs au niveau d'enneigement des sols (attribut « état du sol ») et à la classification des précipitations (attribut « état de l'atmosphère »).
Des réseaux de neurones profonds standard (VGG et ResNet), pré-entraînés ou non, ont été entraînés sur des problèmes « complets » (multiclasse) ou sur des problèmes de classification binaire définis en fusionnant les classes. Sur les problèmes complets, les meilleurs scores en validation sont obtenus par des réseaux de neurones profonds de taille intermédiaire (ResNet50), pré-entraînés sur ImageNet. Le pré-entraînement améliore sensiblement les scores.

La comparaison avec l'existant souligne la difficultés des problèmes abordés : sur les images sélectionnées pendant et autour des épisodes de neige la différence entre les classes est souvent subtile, en particulier la nuit. Mais les erreurs commises restent, à notre avis, trop nombreuses pour que ces modèles soient appliqués dans un contexte opérationnel.

Les sections suivantes sont consacrées à l'apprentissage de préférences. Nous nous sommes concentrés sur le paramètre visibilité, qui a été annoté en premier. Nous avons d'abord montré que les données rassemblées au chapitre 2 permettaient d'entraîner efficacement un classifieur à une tâche de prédiction par paire. En particulier, nous avons montré l'effet très positif des paires d'images non-consécutives (deuxième étape de l'annotation) sur les performances en généralisation : les paliers atteints en validation (généralisation forte) sont supérieurs de plus de cinq points de justesse et les apprentissages se prolongent sur plus d'une centaine d'époques.
Nous avons aussi montré que les paires incomparables pouvaient être apprises pour une meilleure restitution de la relation d'incomparabilité.
De plus, comparé à un réseau pré-entraîné appris « en siamois » (prédiction par image), la prédiction par paire est plus avantageuse au plan des performances. L'avantage se creuse nettement lorsqu'on utilise un ranking-SVM plutôt qu'un réseau de neurones pré-entraîné.
Enfin, sur le jeu TENEBRE_1218, l'un de nos classifieurs a été éprouvé sur une tâche de détection de dépassement de seuil. Comparé à l'existant, les scores sont du même ordre alors que le problème abordé, la généralisation à de nouvelles scènes, est plus difficile.

Mais la prédiction de relations binaires n'est pas simple à utiliser dans une perspective opérationnelle. Pour situer une image dans une séquence, il faut en effet pouvoir archiver des images de référence et procéder à des comparaisons. Les fonctions de rang, qui prédisent « un indice » n'ont pas ce désavantage. Il est de plus envisageable de les étalonner pour fournir une valeur quantitative du paramètre. Dans le chapitre suivant, nous nous concentrons sur l'amélioration et l'étalonnage des fonctions de rang.

Chapitre 4

Amélioration et étalonnage des fonctions de rang

Comparé à la prédiction par paires, les fonctions de rang ont deux avantages appréciables. D'abord, elles permettent d'induire directement une relation d'ordre sur des séquences webcam. En particulier, il n'est pas nécessaire d'archiver l'image pour reconstruire la relation. Ensuite, il peut être envisagé de les étalonner, de façon à passer d'une prédiction de nature ordinale à une prédiction quantitative du paramètre d'intérêt.

Mais d'un autre côté, sur notre jeu de données, les fonctions de rang se sont montrées moins performantes que les classifieurs entraînés à la prédiction par paires. De plus, elles ne rendent pas compte de la relation d'incomparabilité contenue dans l'annotation.

Deux questions naturelles se posent alors : peut-on tirer partie des performances de la prédiction par paires pour améliorer les fonctions de rang ? Est-il possible d'étendre le concept de fonction de rang de manière à apprendre et à restituer une relation d'ordre partiel ?

Ce questionnement nous a amené à proposer deux méthodes d'apprentissage originales. La première repose sur une approche semi-supervisée (section 1). La seconde est basée sur le concept de fonction de rang bivaluée (section 2). Ces fonctions de rang induisent un ordre d'intervalle. Elles nous permettront d'apprendre et, dans une certaine mesure, de restituer la relation d'incomparabilité contenue dans l'annotation.

Nous nous sommes ensuite posé la question de l'étalonnage. Dans la littérature, il est généralement réalisé par régression sur des mesures colocalisées. Dans notre cas, les caméras d'intérêt ne sont pas (et ne ne peuvent pas être) associées à un instrument de mesure. Est-il possible d'étalonner nos fonctions de rang dans ces conditions ? Comme ces fonctions de rang n'ont été entraînées que sur des comparaisons intra-séquence, il n'est a priori pas possible de les étalonner sur une scène de référence. Pour contourner cet obstacle, nous suivons deux pistes de natures différentes.

La première part du constat suivant : si d'un site à l'autre, les mesures de visibilité sont mal corrélées, les distributions du paramètre sont en revanche assez similaires. Or, pour étalonner un proxy, il suffit d'en caler la distribution sur celle de la quantité d'intérêt (histogram matching). En section 3, ce procédé est d'abord testé dans le cas irréaliste où les données colocalisées sont disponibles. Les images

du réseau TENEBRE nous en donnent l'occasion. Puis, nous évaluons l'erreur commise lorsque la distribution locale est estimée à partir d'une distribution distante.

La seconde piste part du même constat, mais l'exploite de façon différente. Puisque le rang d'une image dans une longue série d'images ne dépend pas de la scène, mais de la distribution du paramètre, et que les distributions varient peu d'un site à l'autre, nous pensions qu'en ciblant ce rang lors d'un nouvel apprentissage, il serait possible de s'affranchir de la dépendance à la scène tout en conservant la monotonicité. Cette intuition est fondée et mise en pratique dans la section 4.

4.1 Une méthode semi-supervisée

Dans cette section, nous décrivons une méthode ayant permis d'améliorer les performances en généralisation des fonctions de rang. Cette méthode a consisté à utiliser la prédiction par paire pour construire un jeu étendu, AMOSvvEXt, sur lequel apprendre les fonctions de rang. Nous comptions sur les bonnes performances en généralisation du classifieur symétrisé vu au chapitre 3 pour fournir des cibles automatiques de bonne qualité sur l'ensemble des séquences AMOS non annotées. Au plan théorique, cette idée s'apparente à une distillation semi-supervisée. L'utilisation d'une deuxième image, dans la prédiction par paires, est en effet analogue à une information complémentaire [1] dont ne bénéficierait que le modèle professeur [2]. Le lecteur intéressé pourra consulter [81] pour plus de précisions.

4.1.1 Construction du jeu AMOSvvExt

Dans cette section nous donnons les lignes directrices suivie lors de la construction du jeu étendu, nommé AMOSvvExt. Les détails techniques sont donnés dans l'annexe D.2.
Cette méthode comporte une première phase de collection de séquences d'images prises de jour et associées à une même caméra. La difficulté résidait dans le fait qu'une série AMOS contient souvent les images de plusieurs caméras utilisées à tour de rôle. Cette phase a permis de collecter, à partir d'environ 5.000 sites internet archivés dans AMOS, 12,241 sous-séquences de plus de 20 images de jour. Nous avons pris soin d'éviter les caméras AMOS déjà exploitées pour la validation et pour le test [3].

Dans une seconde phase, le classifieur ayant obtenu les meilleurs scores en validation (le VGG13 vv_pl_du0.0) du chapitre 3 permet de repérer les épisodes de faible visibilité au sein de ces sous-séquences et de générer des comparisons automatiques. La procédure a été conçue suivant trois principes :

- équilibrer le jeu de données : seules les sous-séquences comportant des épisodes de faible visibilité sont échantillonnés. Ces épisodes couvrent au moins la moitié des images sélectionnées.
- sélectionner des comparaisons qui soient compatibles avec un ordre partiel de façon à prémâcher le travail d'apprentissage. Plus, précisément les comparaisons retenues sont compatibles avec un ordre d'intervalle, que des fonctions de rang bivaluées pourront restituer (voir section suivante).
- faire attention au temps de calcul : l'algorithme devra pouvoir être exploité sur un nombre de séquences beaucoup plus important.

La procédure décrite dans l'annexe D.2 permet de sélectionner 135.430 images et 4.390 sous-séquences comprenant des épisodes de faible visibilité. Ces sous-séquences sont issues de 2.596 réper-

1. traduction de priviledged information
2. Dans le cadre d'une distillation classique, le mot clef « teacher » désigne le modèle source
3. Les caméras des DIRs identifiées dans les archives AMOS ont été écartées.

jeu	séquences	images	comparaisons strictes		incomparabilités		équivalences	
			graphe	arêtes	graphe	arêtes	graphe	arêtes
AMOSvv	320	9,850	$\mathcal{G}_0^v$	142,311	$\mathcal{U}_0^v$	34,428	$\mathcal{E}_0^v$	3,144
AMOSvvExt	4,390	135,430	$\mathcal{G}_1^v$	3,682,949	$\mathcal{U}_1^v$	464,226	$\mathcal{E}_1^v$	28,683

TABLE 4.1 – Effectifs du jeu d'entraînement d'AMOSvvExt comparé à celui d'AMOSvv.

toires d'AMOS et comptent donc au moins autant de scènes différentes. La majorité sont des scènes routières.

Les comparaisons automatiques sont stockées dans les graphes $\mathcal{G}_1^v$, $\mathcal{U}_1^v$ et $\mathcal{E}_1^v$. Les arêtes de $\mathcal{E}_1^v$ proviennent de paires d'images prises en dehors des épisodes de faible visibilité.

En nombre d'images et de comparaisons, AMOSvvExt est supérieur à AMOSvv d'un ordre de grandeur. La table 5.3 donne les caractéristiques des graphes $\mathcal{G}_1^v$, $\mathcal{U}_1^v$ et $\mathcal{E}_1^v$ avant la dernière étape où les images et les arêtes des graphes $\mathcal{G}_0^v$, $\mathcal{U}_0^v$ et $\mathcal{E}_0^v$ sont ajoutées.

4.1.1.1 Impact sur l'apprentissage d'une fonction de rang à valeurs réelles

Pour évaluer l'apport d'un entraînement sur AMOSvvExt, nous avons entraîné des ResNet50 avec la même procédure d'entraînement que sur AMOSvv. La justesse atteinte en validation est supérieure de un à deux points pour les trois modèles entraînés (voir figure 4.1.a).

Sur le jeu de test, l'écart entre les deux groupes est d'environ deux points de justesse (table 4.2). Cet écart se retrouve figure 4.1.b, les Correctness au point d'abscisse 1 vérifiant :

$$\text{Correctness} = 2 \times \text{Justesse} - 1$$

L'écart se maintient pour une Completeness supérieure à 0.8. En deça, elle décroît progressivement avec la Completeness, sans pour autant changer de signe. Du côté des compromis Completeness-Prudence, il n'y a pas d'effet clair lié à l'apprentissage sur AMOSvvExt (figure 4.1.c). Mais au moins peut-on dire que le gain en Correctness n'est pas contreblancé par une perte de Prudence.

modèles	Justesse sur $\mathcal{G}_{test}^v$
vv_sl_d0.0 (ResNet50)	0.898
vv_sl_d0.1 (ResNet50)	0.898
vv_sl_de00.1 (ResNet50)	0.897
vv_sl_d1.0 (ResNet50)	0.918
vv_sl_d1.1 (ResNet50)	0.921
vv_sl_de11.0 (ResNet50)	0.922

TABLE 4.2 – Comparaisons entre fonctions de rang entraînées sur AMOSvv (lignes 2-4), et fonctions de rang entraînées sur AMOSvvExt (lignes 5-7).

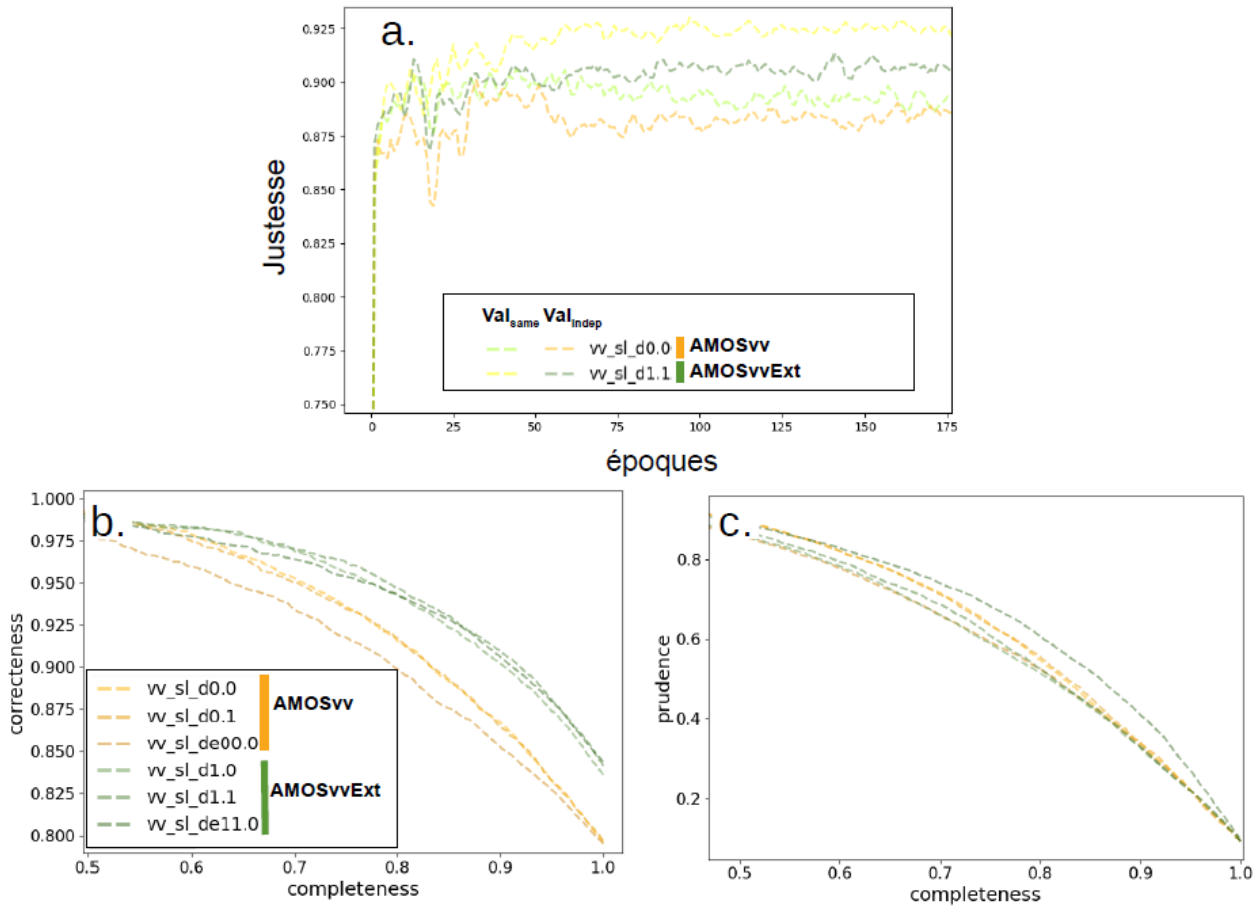


FIGURE 4.1 – Comparaison entre des fonctions de rang implémentées sur des ResNet50 et apprises soit sur AMOSvv (méthode supervisée) soit sur AMOSvvExt (méthode semi-supervisée). a. Courbes d'apprentissage représentatives pour chacune des deux méthodes. b. Comparaison sur le jeu de test à partir des métriques définies en section 3.2.1.2. Pour chaque méthode, deux fonctions de rang ont été apprises sur les paires strictement ordonnées. L'apprentissage de la troisième fonction de rang fait intervenir les équivalences (voir l'annexe C).

4.1.2 Discussion sur l'approche semi-supervisée

L'apprentissage sur AMOSvvExt a permis d'améliorer la correctness des fonctions de rang implémentées sur des architectures de type ResNet.

Cette amélioration peut tenir à différentes raisons. Nous avons mis en avant la plus naturelle -les meilleures performances du classifieur, mais d'autres éléments ont pu contribuer à cette amélioration. On peut par exemple se demander si l'étape de correction des prédictions par paire joue un rôle. Ou encore, si le fait d'avoir sélectionné des comparaisons compatibles avec une relation d'ordre partielle particulière a de l'importance. Ces questions sont intéressantes, mais elles dépassent le cadre de notre travail.

Par contre, une question qui nous intéresse davantage est de savoir si l'approche peut être appliquée à d'autres paramètres avec le même succès. Nous y répondrons dans le chapitre suivant.

Il serait aussi intéressant de tester ce type d'approche avec un jeu annoté à la main de plus grande taille. Le succès grandissant des approches semi-supervisées donne de bonnes raisons d'espérer une

plus-value. En effet, les dernières avancées sur les grands problèmes de classification se sont faites en majorité à l'aide de jeux auxiliaires [4] alors que les jeux annotés à la main qui interviennent dans ces études sont plus grands que les nôtres de plusieurs ordres de grandeur.

4. https ://paperswithcode.com/sota/image-classification-on-imagenet

4.2 Prédiction d'un ordre d'intervalle

Pouvoir évaluer la précision de l'information contenue dans l'image nous intéresse pour deux raisons. Premièrement, cela nous permettrait de faire un pas en direction d'un encadrement du paramètre d'intérêt. Ensuite, nous pensons qu'apprendre à reconnaître des causes d'incertitude, comme les gouttelettes et les flocons qui collent à la lentille, peut aider à améliorer les performances quantitatives. En effet, ces « bruits ». sont souvent évoqués pour justifier les erreurs de prédiction [32], [36], [109].

Dans nos données, c'est la relation d'incomparabilité qui caractérise la précision de l'information contenue dans l'image. Pour apprendre et restituer cette relation d'incomparabilité avec une fonction de rang, nous proposons de prédire un intervalle fermé de $\mathbb{R}$ putôt qu'un réel. Deux intervalles prédits seront considérés comme incomparables en cas de chevauchement.
Cette extension a du sens : c'est à travers de plus larges intervalles que les images de mauvaise qualité, les scènes difficiles à traiter, les situations d'atmosphère ou d'éclairage hétéorogènes pourront être prises en compte.

Dans cette section nous commençons par un point théorique sur la nature de la relation d'ordre induite par des intervalles. Nous cernons ainsi les limites de notre approche. Dans un second temps, nous précisons la façon dont nous apprenons à prédire des intervalles à partir de nos jeux de données. Nous introduisons pour cela les fonctions de rang bivaluées et des fonctions de coût adaptées.
Enfin, nous présentons et nous discutons les résultats obtenus sur le jeu de test d'AMOSvv.

4.2.1 Notion d'ordre d'intervalle et limites d'utilisation.

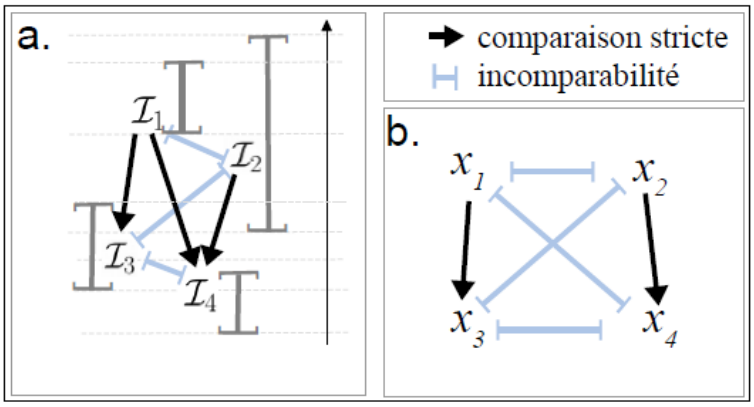


FIGURE 4.2 – a. L'ordre partiel défini sur l'ensemble des $\mathcal{I}_k$ est un ordre d'intervalles. L'incomparabilité correspond au chevauchement. b.Les éléments $x_1; x_2; x_3; x_4$ ne peuvent pas être représentés sous la forme d'intervalles. C'est un ensemble ordonné de type "2+2".

Commençons par définir la notion d'ordre d'intervalle.
Une famille $\mathcal{F}$ d'intervalles de $\mathbb{R}$ peut être muni de l'ordre partiel strict suivant :

$$\forall \mathcal{I}, \mathcal{J} \in \mathcal{F} : \mathcal{I} \prec \mathcal{J} \Leftrightarrow \mathcal{I}^+ < \mathcal{J}^- \tag{4.1}$$

où $\mathcal{I}^+$ (resp. $\mathcal{J}^-$) représente la borne supérieure de $\mathcal{I}$ (resp. la borne inférieure de $\mathcal{J}$).
Cette relation d'ordre partiel, illustrée sur la figure 4.2, est appelée ordre d'intervalle. On peut étendre la notion d'ordre d'intervalle à n'importe quel ensemble ordonné à travers la notion d'isomorphisme d'ordre :

Definition 2 *Isomorphisme d'ordre.* Soient $(E, \prec_E)$, $(F, \prec_F)$ deux ensembles munis d'ordres partiels stricts. On appelle isomorphisme d'ordre une bijection f de E dans F qui conserve parfaitement l'ordre, c'est à dire qui vérifie :

$$\forall x, y \in E, x \prec_E y \Leftrightarrow f(x) \prec_F f(y) \tag{4.2}$$

Definition 3 *Ordre d'intervalle.* Soit $(E, \prec)$ un ensemble partiellement ordonné. $\prec$ est un ordre d'intervalle si $(E, \prec)$ est isomorphe à une famille d'intervalles fermés muni de la relation 4.1.

Ainsi, par définition, la prédiction d'un intervalle pour chaque image d'une séquence induit un ordre d'intervalle sur la séquence. Pour restituer parfaitement les comparaisons contenues dans l'annotation, il faut qu'elles soient compatible avec un ordre d'intervalle.
Est-ce le cas ? La propriété suivante permet de répondre à cette question :

Proposition 1 *Caractérisation d'un ordre d'intervalle [110] Si un ensemble partiellement ordonné $(E, \prec)$ ne contient aucun sous-ensemble vérifiant $x_3 \prec x_1$, $x_2 \prec x_4$ et $x_1 \perp x_4$, $x_2 \perp x_3$, alors $\prec$ est un ordre d'intervalle. Un tel sous-ensemble est dit*[5] *« de type 2+2 » [111] (voir un exemple figure 4.2.b).*

Nous savons déjà qu'il existe des sous-ensembles de type 2+2 dans nos jeux de données (voir la figure A-10). Mais ce qui nous importe en priorité, c'est de faire mieux qu'avec des fonctions de rang épaissies, qui induisent une relation d'ordre partiel moins souple[6].
Mentionnons une autre limite, plus difficile à caractériser. Même en supposant que l'annotation est parfaitement compatible avec un ordre d'intervalle, il faut pouvoir l'obtenir à travers une prédiction par image. Or une simple image peut ne pas contenir assez d'éléments pour remonter à des largeurs d'intervalles convenables. Pour préciser ce point, prenons l'exemple d'une image floutée par la condensation sur la lentille externe ou sous la vitre de protection, et floutée de telle manière que la distance aux objets de la scène (profondeur) ne puisse pas être estimée (voir par exemple la figure 4.6.b dans ce même chapitre). En prédiction par paires, des informations sur la profondeur de la scène peuvent toujours être déduite de la seconde image. En prédiction par image, comme la profondeur ne peut pas être estimée à partir de l'image floutée, l'incertitude portant sur la visibilité est plus grande. Il est alors impossible de remonter à l'intervalle qui conviendrait (problème mal posé).

Malgré ces limites, incompatibilité et caractère mal posé du problème, il nous semblait intéressant d'expérimenter une prédiction d'intervalle par image.

5. traduction de 2+2 free-poset
6. précisément, une fonction de rang épaissie induit unsemi-ordre [108].

4.2.2 Fonctions de rang bivaluées

Pour prédire des intervalles, nous les paramétrons par leurs bornes. Nous passons par des fonctions de rang de la forme $f(x_i; w) = (z_i^-, z_i^+)$ où $z_i^- < z_i^+$ représentent les bornes de l'intervalle associé à x_i. Nous les appelons fonctions de rang bivaluées.

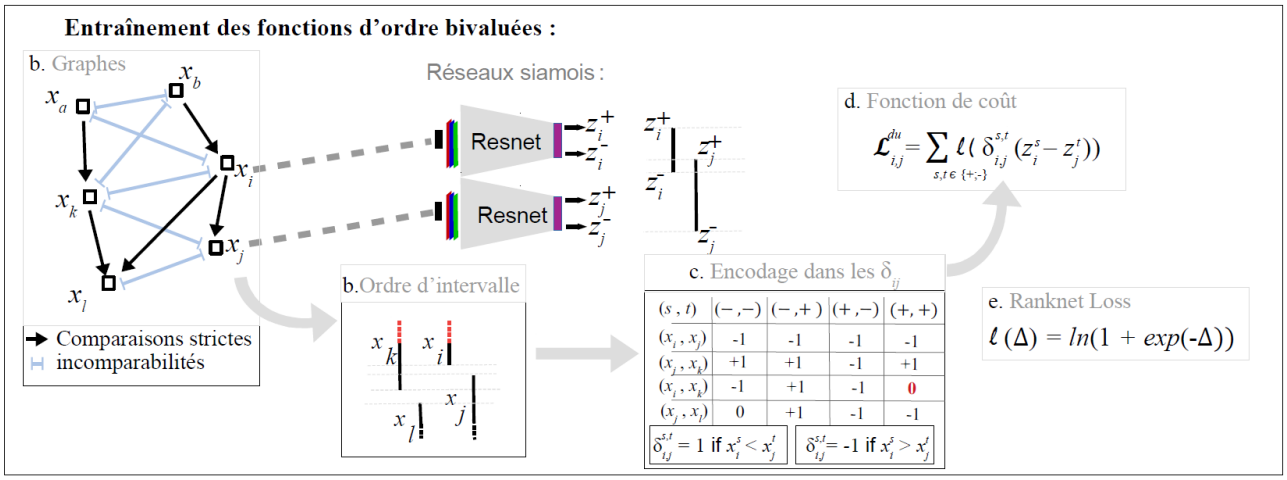


FIGURE 4.3 – Entraînement d'une fonction de rang bivaluée. Les sorties (z_i^-, z_i^+) du modèle induisent un ordre d'intervalle. La relation d'ordre contenue dans les graphes (a.) peut être partiellement représentée sous la forme d'un ordre d'intervalle (b.). La position relative des bornes supérieures des intervalles associés à x_i, x_k (segments rouges) ne peut pas être précisée à cause d'un sous-ensemble de type 2+2 ($\{x_a; x_b; x_i; x_k\}$). Les positions relatives des bornes sont encodées dans les $\delta_{i,j}$ (c.). Dans la fonction de coût (d.), les bornes mal positionnées sont pénalisées par la RankNet Loss (e.).

Comme les fonctions d'ordre du chapitre précédent, nos fonctions d'ordre bivaluées sont implementées sur des réseaux de neurones à couches de convolution standard et apprises en siamois (voir figure 4.3). A ce niveau, la seule modification consiste à prendre deux neurones plutôt qu'un dans la dernière couche du réseau. Pendant l'apprentissage, les contradictions entre les comparaisons contenues dans les graphes et l'ordre d'intervalle induit par la fonction $f(.; w)$ sont pénalisées par une fonction de coût spécifique qui généralise les fonctions utilisées précédemment.

4.2.3 Fonction de coût

Nous cherchons à apprendre une fonction de rang bivaluée $f(., w)$ qui préserve au mieux la relation d'ordre contenue dans l'annotation (notée $\prec$). Pour deux images x_i et x_j telles que $x_j \prec x_i$, les intervalles prédits $[z_i^-, z_i^+]$ et $[z_j^-, z_j^+]$ doivent donc vérifier $z_i^- > z_j^-$, $z_i^- > z_j^+$, $z_i^+ > z_j^-$ et $z_i^+ > z_j^+$. Ces contraintes sont encodées dans les $\delta_{i,j}^{s,t}$ de la deuxième ligne de la table c., figure 4.3.
De deux images incomparables, on ne peut déduire en général que deux inégalités. Par exemple, $x_j \perp x_k$ implique les deux contraintes : $z_j^- \leq z_k^+$ et $z_j^+ \geq z_k^-$, ce qui est codé par $\delta_{j,k}^{-,+} = +1$ et dans $\delta_{j,k}^{+,-} = -1$ (3^{me} ligne de la table). Cependant, dans certains cas, les relations avec une troisième image contraignent les positions relatives des deux autres bornes.

Par exemple, figure 4.3, nous avons $x_i \perp x_k$, $x_i > x_j$ et $x_k \perp x_j$. Pour conserver ces trois relations, il est nécessaire d'avoir $z_i^- > z_k^-$ et $z_j^+ < z_k^+$, ce qu'on code par $\delta_{i,k}^{-,-} = -1$ et $\delta_{j,k}^{+,+} = +1$.
Autrement, ces valeurs sont réglées sur zéro (e.g. $\delta_{i,l}^{-,-} := 0$). Lorsqu'il y a une contradiction, ces valeurs sont aussi réglées sur zéro (e.g. $\delta_{i,k}^{+,+} := 0$ dans la table). On peut remarquer que ces contradictions correspondent exactement aux sous-ensemble de type 2+2 (e.g. $\{x_a; x_b; x_k; x_i\}$).

La fonction de coût intègre ces contraintes. Lorsque les paires d'images « équivalentes » ne sont pas utilisées, elle est donnée par :

$$\mathcal{L}^{du}(\delta_{i,j}, x_i, x_j; w) = \frac{1}{4} \sum_{s,t \in \{-,+\}} \ell(\delta_{i,j}^{s,t}(z_i^s - z_j^t)) \tag{4.3}$$

où la fonction ℓ peut être définie soit par la Hinge Loss par la RankNet Loss (voir chapitre 3).

Il fallait enfin contraindre les deux composantes de l'output à être rangées dans l'ordre croissant. Pour cela, nous ajoutons à la fonction de coût définie par (4.3) la pénalisation suivante :

$$[z_i^+ - z_i^-]_- + [z_j^+ - z_j^-]_- \tag{4.4}$$

où $[.]_-$ représente la fonction partie négative.
Si l'annotation est compatible avec un ordre d'intervalle, minimiser la fonction coût revient à respecter toutes les contraintes entre les bornes et donc à conserver l'ordre d'intervalle.
Cette fonction de coût a été testée sur un problème d'apprentissage construit à partir d'images de synthèses. Nous nous sommes placés dans le cas idéal où l'annotation est compatible avec un ordre d'intervalle et où le problème est bien posé. La démarche est détaillée dans l'annexe D.4.
Une fois entraîné, un réseau de neurones à couche de convolution restitue l'ordre d'intervalle ciblé avec une justesse quasi-parfaite sur le problème à trois classes. Nous sommes alors passés aux tests sur les données réelles.

4.2.4 Utilisation des images équivalentes et moyennage

Avant de décrire les résultats, nous précisons ici deux derniers aspects d'ordre méthodologique. Il s'agit d'abord de la façon dont les paires d'images équivalentes sont apprises et ensuite de la manière dont les modèles peuvent être combinés pour obtenir une fonction de rang plus performante.

4.2.4.1 Prise en compte des équivalences :

Un autre aspect de notre approche concerne la façon dont nous avons utilisé les graphes contenant les relations d'équivalence ($\mathcal{E}_0$ et $\mathcal{E}_1$).

Sur deux images équivalentes, l'ensemble des valeurs de visibilités plausible est le même. Nous contraignons donc les intervalles de sortie du modèle à être égaux, via la fonction de coût :

$$\mathcal{L}^e(x_i, x_j; w) = \frac{1}{2} \sum_{s\in\{-,+\}} \ell_e(z_i^s - z_j^s) \tag{4.5}$$

Selon que la Hinge Loss ou la RankNet Loss est choisie, on applique le cas d'ex-aequo correspondant (équations (3.7-3.8), chapitre 3).

Sans autre disposition, l'utilisation des paires d'images équivalentes sur l'apprentissage n'a pas semblé avoir d'effet. Il y a plusieurs raisons possibles à cela. D'abord, les paires d'images équivalentes sont déjà présentes dans les graphes $\mathcal{U}_0$ et $\mathcal{U}_1$, en tant que paires d'images incomparables. Mais surtout, ce sont des images qui sont généralement très proches visuellement.
Nous avons donc pris le parti de les utiliser à travers une tâche d'identification un peu plus corsée : il s'agira de reconnaître des intervalles de visibilités égaux sur des scènes identiques, mais vues sous des angles différents. En pratique, plutôt que de présenter des paires d'images bien alignées aux réseaux siamois, nous appliquons deux transformations perspectives indépendantes sur les deux images.
Nous espérions ainsi favoriser l'émergence de caractéristiques invariantes par changement d'orientation de la caméra, changements qui se produisent inévitablement sous l'effet du vent et de la température.

Considérant cette tâche d'identification comme une tâche de nature différente, nous avons intégrons l'expression 4.7 comme dans un apprentissage multitâche, à travers une pondération (poids w_{du} et w_e mise à jour au cours de l'entraînement, suivant l'exemple donné dans [112] :

$$\mathcal{L}^{due} := \frac{1}{2w_{du}^2}\mathcal{L}^{du} + \frac{1}{2w_e^2}\mathcal{L}^e + ln(1 + w_u^2 + w_e^2) \tag{4.6}$$

où les poids w_{du} et w_e sont des poids entraînables intégrés au modèle et :

$$\mathcal{L}^e(x_i, x_j; w) = \frac{1}{2} \sum_{s\in\{-,+\}} \ell_e(z_i^s - z_j^s) \tag{4.7}$$

Cette tâche supplémentaire donnait aussi la possibilité d'augmenter la taille du jeu d'apprentissage. En effet, la même image vue sous deux angles différents pose déjà un problème d'identification non trivial. Nous avons donc ajouté les boucles (couples (x_i, x_i)) aux arêtes des graphes $\mathcal{E}_0$ et $\mathcal{E}_1$.

4.2.4.2 Moyennage des fonctions de rang :

Comme les fonctions d'ordre à valeur réelle, les fonctions de rang bivaluées sont additives. On définit la moyenne de l'ensemble des fonctions de rang bivaluées $\{f_i(., w_i)\}_{i=0..n}$ par :

$$\bar{f}_i(x) = (\frac{1}{n}\sum_{i=1}^{n} z_i^-, \frac{1}{n}\sum_{i=1}^{n} z_i^+) \tag{4.8}$$

où $f_i(x, w_i) = (z_i^-, z_i^+)$.
Comme sur le jeu de synthèse (voir annexe D.4), ce sont des fonctions bivaluées moyennes qui vont permettre d'atteindre les meilleures performances.

4.2.5 Entraînement des modèles

Pour entraîner les fonctions de rang bivaluées, nous suivons la procédure définie au chapitre 3 (section 3.2.4) pour les fonctions de rang à valeurs réelles. La seule différence consiste à introduire des paires incomparables dans les mini-lots durant l'apprentissage. La fréquence de présentation des paires incomparables est contrôlée par le paramètre p_u, celle des paires strictement ordonnées par p_g et celle des équivalences par p_e. Sauf mention contraire, lorsqu'un modèle est entraîné avec la fonction de coût $\mathcal{L}^{du}$, c'est selon les fréquences de présentation $p_u = 1/3$, $p_g = 2/3$. Avec $\mathcal{L}^{due}$, nous prenons $p_g = 1/3$, $p_u = 1/6$, $p_e = 1/6$.
Sur le jeu de validation, au cours d'une époque, toutes les arêtes de $\mathcal{G}^v_{vali}$ et $\mathcal{U}^v_{vali}$ sont traitées. Ainsi, comme pour les expériences de la section précédente, les fréquences de présentation sont toujours de $p_g = 6.184/11.483 \approx 0.54$ et $p_u \approx 0.46$.

Des fonctions de rang bivaluées ont été entraînées sur AMOSvv et AMOSvvExt. Nous donnons des exemples de courbes d'apprentissage figure 4.4. Contrairement au cas des fonctions d'ordre à valeurs réelles, la justesse est calculée sur les trois classes. Les justesses en validation sont inférieures aux justesses à l'entraînement de 15 à 20 points.
D'autre part, sur le jeu de validation, les modèles entraînés sur AMOSvvExt atteignent des justesses de l'ordre de 73% - 74% contre 70% - 71% pour ceux entraînés sur AMOSvv. Par la suite, nous nous concentrons donc sur la comparaison des modèles entraînés sur AMOSvvExt.

Sur AMOSvvExt, nous avons entraîné sept modèles (ResNet50) dont deux avec $\mathcal{L}^{du}$ et cinq avec $\mathcal{L}^{due}$ (voir table 4.3). Les trois premiers (groupe 1) sont entraînés avec des fréquences de présentation standard $p_u = p_e = 1/6$. Les quatre derniers sont entraînés avec $p_u = p_e = 1/3$. Elles forment le groupe 2. Le dernier de ces modèles (vv_sl_due101.0) a été entraîné avec les arêtes de $\mathcal{U}^v_0$ au lieu de $\mathcal{U}^v_1$.
Les poids sélectionnés sont ceux qui maximisent la justesse associée au problème à trois classes sur le jeu de validation.

4.2.6 Evaluation des fonctions de rang bivaluées

Nous donnons ici les performances quantitatives des fonctions de rang bivaluées sur le jeu de test ; d'abord sur le problème à deux classes, en désactivant la prédiction des incomparabilités, ensuite sur le problème à trois classes, en les comparant à des fonctions de rang à valeurs réelles épaissies.
Des aspects plus qualitatifs sont présentés dans un second temps. Nous montrons d'abord que les tailles des intervalles prédits sont cohérentes. Les plus grands intervalles correspondent par exemple

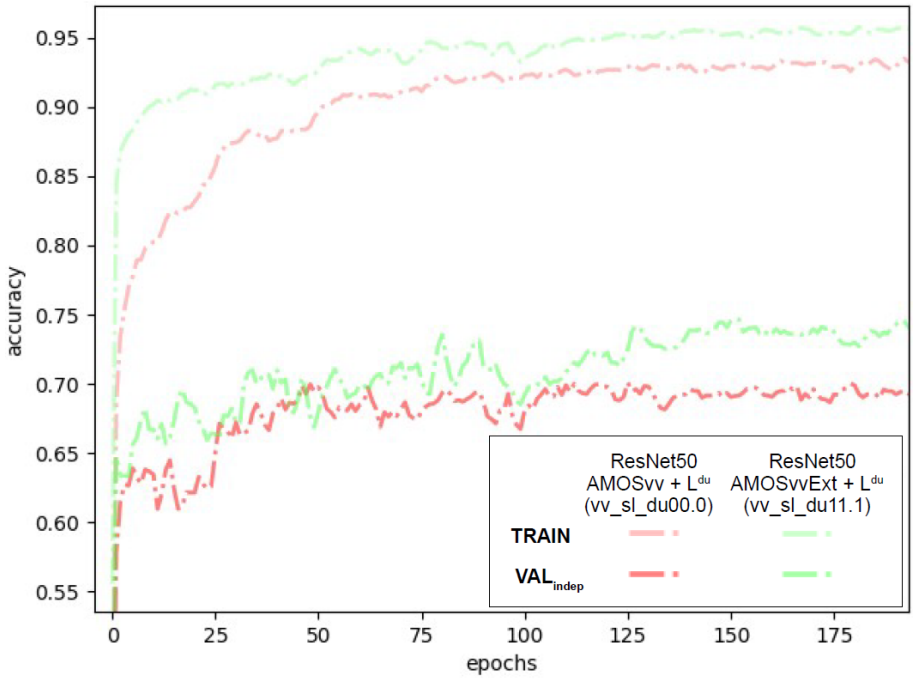


FIGURE 4.4 – Entraînement avec $\mathcal{L}^{du}$ sur AMOSvv ($\mathcal{G}_0^v$ et $\mathcal{U}_0^v$) et AMOSvvExt ($\mathcal{G}_1^v$ et $\mathcal{U}_1^v$). La justesse est calculée sur les trois classes. Les courbes d'apprentissage sont légèrement lissées (moyenne mobile sur trois époques consécutives).

aux images bruitées et aux scènes les plus difficiles à annoter. Pour terminer, nous décrivons quelques cas d'erreur systématiques.

4.2.6.1 Comparaison sur le problème à deux classes :

Pour la comparaison avec des fonctions de rang à valeurs réelles, la procédure d'« épaississment » est étendue aux fonctions de rang bivaluées (encadré 4.1). En particulier, les fonctions de rang bivaluées sont converties en fonctions de rang à valeurs réelles en considérant les centres des intervalles.

ENCADRÉ 4.1 – Epaississement des fonctions de rang bivaluées

Soit $f(.;w)$ une fonction de rang bivaluée. La procédure d'épaississement consiste à moduler la taille des intervalles prédits autour des centres d'intervalle. Pour tout $\lambda \geq 0$ et $x \in \mathcal{D}_X$ on pose :

$$f_\lambda(x;w) = [z - \lambda r, z + \lambda r] \tag{4.9}$$

où $z = \dfrac{f(x;w)^+ + f(x;w)^-}{2}$ et $r = \dfrac{f(x;w)^+ - f(x;w)^-}{2}$.
En particulier, pour $\lambda = 1$, $f_\lambda(.;w) \equiv f(.;w)$ et pour $\lambda = 0$, l'intervalle est réduit à son centre tandis que la prédiction d'incomparabilité est désactivée.

Sur le problème à deux classes défini par les 15.017 arêtes de $\mathcal{G}_{TEST}^v$, les centres d'intervalles sont systématiquement mieux rangés que les sorties des fonctions de rang à valeurs réelles.

modèles	fonction de coût	p_g, p_u, p_e	Justesse (2 classes)	Justesse (3 classes)
fonctions de rang à valeurs réelles	RankNet	1, 0, 0 (2/3, 0, 1/3)	0.918 - 0.922	/
mean_real_valued_gr0	/		0.927	
vv_sl_du11.0	$\mathcal{L}^{du}$	2/3, 1/3, 0	0.931	0.637
vv_sl_du11.1			0.934	0.620
vv_sl_due111.0	$\mathcal{L}^{due}$	2/3, 1/6, 1/6	0.931	0.670
mean_bivalued_gr1	/		0.940	0.642
vv_sl_due111.1	$\mathcal{L}^{due}$	1/3, 1/3, 1/3	0.928	0.720
vv_sl_due111.2			0.923	0.698
vv_sl_due111.3			0.924	0.698
vv_sl_due101.0			0.924	0.707
mean_bivalued_gr1gr2	/		0.939	0.685

TABLE 4.3 – Ligne 2 : scores atteints par les trois fonctions de rang à valeurs réelles de la table 4.2. Ces fonctions de rang forment le groupe 0. Ligne 3 : score obtenu par moyennage des fonctions du groupe 0. Lignes 4 - 12 : scores des fonctions de rang bivaluées. Les lignes 7 et 12 sont obtenues par moyennage sur les groupes 1 et 2. Ces groupes diffèrent principalement par les fréquences de présentation (troisième colonne).

D'autre part, les prédictions moyennées sont toujours meilleures que les prédictions individuelles (de 0.5 à 1 point de justesse supplémentaire). La fonction moyennée sur les trois modèles du groupe 1 atteint par exemple 94% de justesse, contre 92.7% pour la moyenne des trois fonctions d'ordre du groupe 0 (voir table 4.3).

4.2.6.2 Effet des fréquences de présentation et du moyennage :

La fréquence de présentation des paires incomparables est l'un des paramètres du problème d'optimisation. Plus elle est élevée, plus le réseau priviligie l'abstention. C'est ce que l'on voit sur la figure 4.5 (4.9) : le doublement de la fréquence de présentation des incomparabilités (groupe 2) déplace l'équilibre vers une Completeness plus faible et une prudence plus élevée. Les points du groupe 2 semblent simplement avoir été obtenus à partir de ceux du groupe 1 par épaississement. C'est aussi le cas sur le diagramme Prudence-Correctness, non montré. Sur ces diagrames, les courbes reflètent donc, en première approximation, l'ensemble des compromis accessibles par modulation des fréquences de présentation.
Sur cette figure, la plus-value liée au moyennage atteint jusqu'à 1 point de Correctness.

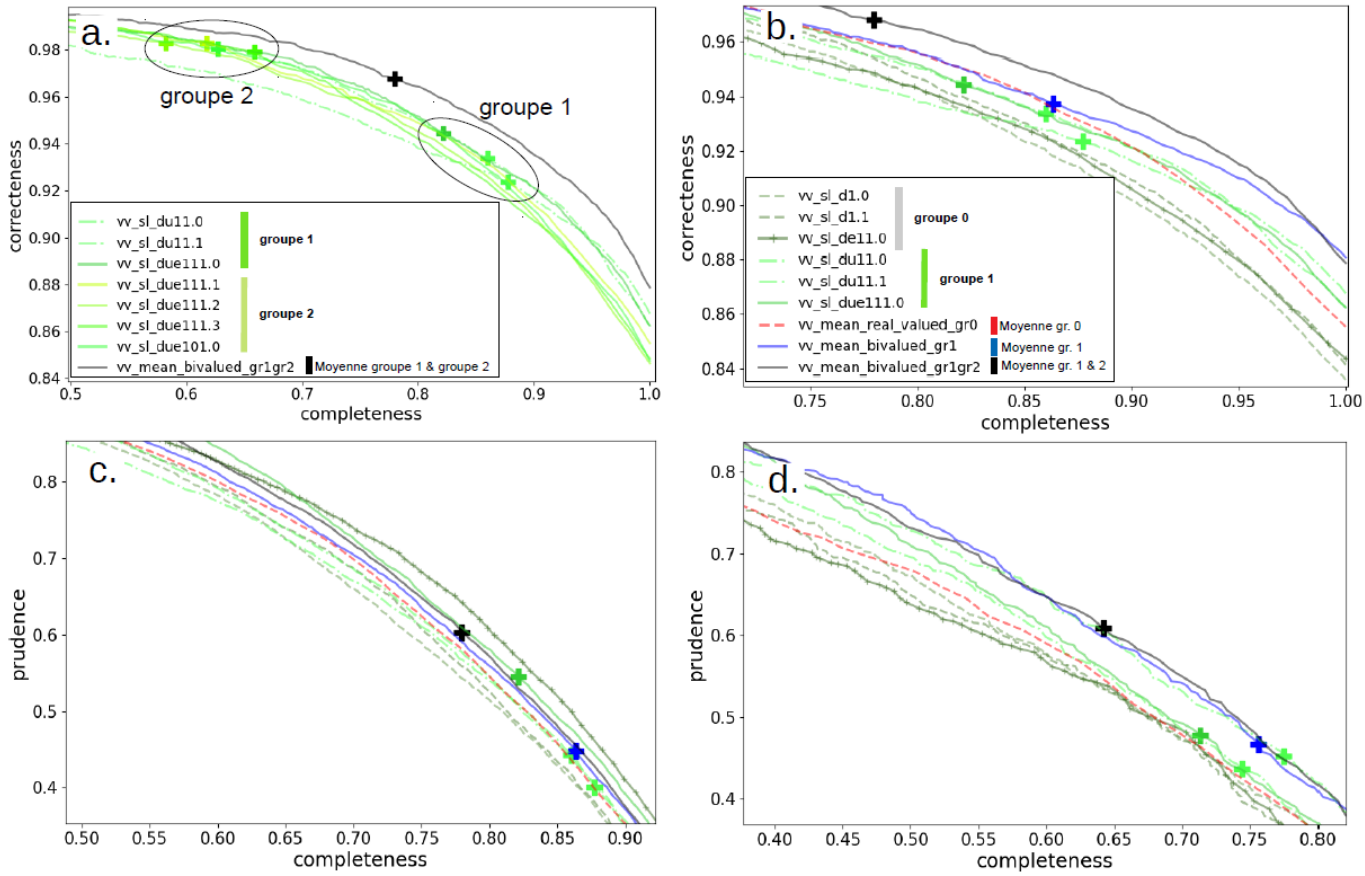


FIGURE 4.5 – Diagrammes Correctness-Completness et Prudence-Completeness sur le jeu de test d'AMOSvv. Les croix resprésentent les compromis Correctness-Completeness (ou Prudence-Completeness) atteints par les fonctions de rang bivaluées. Les courbes représentent les compromis obtenus par épaississement des fonctions de rang à valeur réelle (tiretés) et des fonctions de rang bivaluées (en traits pleins, voir encadré 4.9). a.Effet des fréquences de présentation et du moyennage. La fréquence de présentation des paires incomparables est de 2/3 pour le groupe 2 contre 1/3 pour le groupe 1. La courbe noire représente la moyenne des fonctions de rang bivaluées apprises sur AMOSvvExt. b-c. Comparaison entre fonctions d'ordre à valeurs réelles (groupe 0) et bivaluées (groupe 1). La légende vaut pour les trois graphiques b,c,d. d. Même comparaison qu'en c., mais en limitant le calcul des scores aux paires où au moins une des images est de mauvaise qualité.

4.2.6.3 Comparaison avec des fonctions de rang à valeur réelles épaissies :

Ce sont des fonctions de rang bivaluées qui offrent les meilleurs compromis Correctness-Completeness (courbes du groupe 1, figure 4.5.b). Cependant, les courbes associées aux moyennes des groupes 0 et 1 sont confondues pour des Completeness inférieures à 0.85 (courbes bleue et rouge).

Sur le diagramme Prudence-Completeness (figure 4.5.c), le bilan est mitigé : il n'y a pas de gradation claire.

Par contre, lorsque le calcul des Completeness et Prudence est limité aux paires comprenant au moins une image de mauvaise qualité [7] -paires sur lesquelles l'ordre d'intervalle est supposé apporter une plus-value- une gradation apparaît, plus marquée pour des Completeness plus petites (jusqu'à 10 points

7. 1.781 arêtes de $\mathcal{G}^v_{test}$ et 1.658 arêtes de $\mathcal{U}^v_{test}$

d'écart en prudence entre les moyennes des groupes 0 et 1). Sur les mêmes paires d'images, le diagramme Correctness-Completeness restait à l'avantage des fonctions de rang bivaluées.

4.2.6.4 Evaluation qualitative. Des intervalles cohérents :

Sur une séquence d'images, les plus larges intervalles prédits correspondent aux images de mauvaise qualité (ou aux images très atypiques). Les images a-e de la figure 4.6 sont par exemple tirées du jeu de test et sont chacunes associées au plus grand intervalle de leur séquence. Même si l'échelle de sortie est arbitraire, cela indique que les causes d'incomparabilité liées aux défauts de qualité les plus fréquents (masques météorologiques, contrejour, etc) ont bien été apprises par le modèle. Les images de nuit, qui n'ont pourtant pas été vues par le modèle, sont généralement [8] associées à des intervalles de grande taille (image f.).
Cette capacité à distinguer les images de mauvaise qualité est partagée par toutes les fonctions de rang bivaluées. Notons que ce résultat est intéressant en soi. Il permet d'envisager la détection d'images de mauvaise qualité.

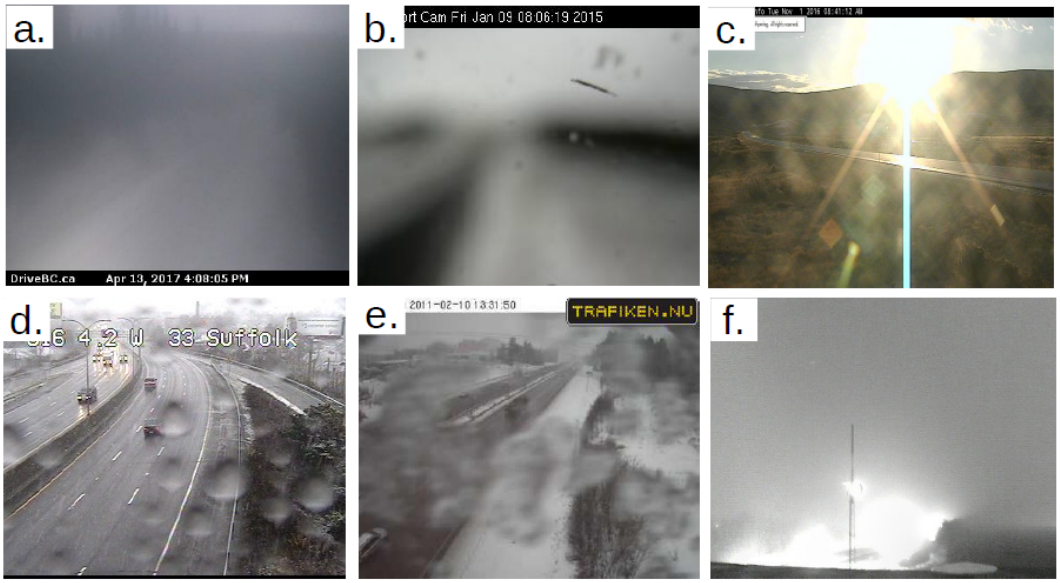


FIGURE 4.6 – a-e. Images du jeu de test d'AMOSvv associées à un intervalle de taille maximum (comparé aux images de même séquence) par le modèle vv_sl_due111.0. L'image a. est floutée par la condensation et l'image b., à cause de la focalisation sur des débris collés à la vitre. f. La dernière image est issue de la séquence entzheim3 (TENEBRE).

D'autres éléments qualitatifs sur les fonctions de rang bivaluées sont présentés dans l'annexe D-V. Nous avons par exemple cherché à savoir si, sur les scènes où la visibilité est difficile à comparer pour l'annotateur, ces modèles prédisent plus fréquemment des incomparabilités. Nous montrons que, lorsque le premier plan ou le second plan sont rognés, les chevauchements sont relativement plus nombreux.

8. Voir aussi la figure A-8.

Nous nous sommes aussi intéressés à la relation d'inclusion entre les intervalles prédits. Nous montrons que les inclusions sont trop rarement prédites, mais qu'elles sont prédites dans le bon ordre (voir la figure 4.7 et l'annexe D, section D.5.2).

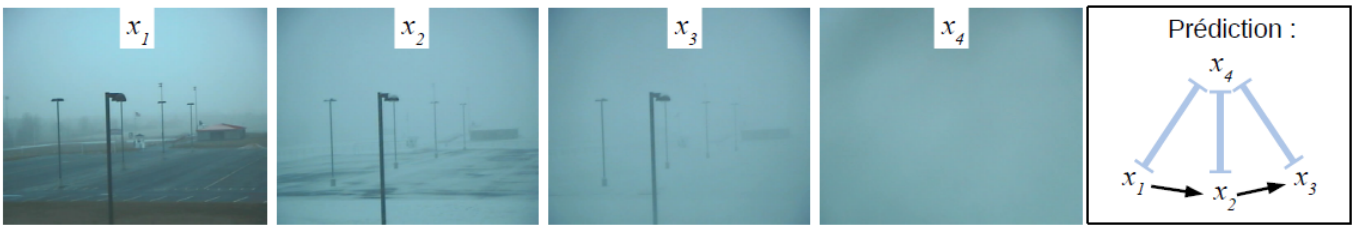


FIGURE 4.7 – Cas d'une inclusion correctement prédite. Les cinq relations prédites sont correctes : entre l'image x_1 et l'image x_3, la visibilité a décru, mais un flocon masque complètement la scène sur l'image x_4. Pour prédire correctement ces relations, il faut que l'intervalle associé à x_4 inclue l'intervalle associé à x_2. Les images viennent de la séquence AMOS n°713 du jeu de test. La dernière image est complètement masquée par un flocon.

4.2.7 Cas d'erreur :

Pour les deux modèles vv_sl_due111.0 et vv_sl_due111.3, un passage en revue des discordances ($\dot{\prec}$ au lieu de $\dot{\succ}$) a été réalisé.
Les discordances sont très rares. Elles apparaissent de manière anecdotiques en lien avec un changement important dans l'orientation de la caméra (séquence AMOS n° 12420) ou dans l'encodage (séquence AMOS n° 15905 : les dimensions de l'image ont changé en milieu de séquence).

La revue des fautes de prudence ($\dot{\prec}$ ou $\dot{\succ}$ au lieu de $\dot{\perp}$) est plus éclairante. Premièrement, les intervalles associés aux images de très mauvaise qualité, sont certes plus grands que les autres, mais ils ne le sont souvent pas assez pour éviter toutes les fautes de prudence. En particulier, des cas de scènes complètement masquées par l'accumulation de neige sur la vitre de protection générent des comparaisons strictes alors qu'elles ne comportent aucune information sur la visibilité présente. Ces comparaisons strictes se produisent surtout avec des images prises par beau temps [9].
Le cas particulier des contrejours est aussi à signaler : mal pris en compte sur certaines scènes, ils sont responsables d'une fluctuation régulière (en général une par jour) de la sortie des modèles. Sur des scènes urbaines présentant de larges surfaces de réverberation, il peut y avoir plusieurs faux épisodes par jour de beau temps.
Deuxièmement, les paires d'images ("*neige au sol*" × "*non neige au sol*") génèrent une partie importante des fautes de prudence. Globalement, ces arêtes sont d'ailleurs traitées avec une prudence de 15 à 20 points inférieure au niveau global.
Enfin, comme attendu, la taille de l'intervalle est généralement plus grande à l'aube et au crépuscule : les difficultés d'estimation liées à une faible luminosité ont été apprises. Néanmoins, ces situations

9. Ces erreurs contribuent à l'asymétrie observée lors de l'étude des inclusions, Annexe D, section D.5.2.

provoquent des fautes de prudence systématiques sur quelques séquences. Les profils par beau temps sont alors en forme de "n" et relativement facile à détecter.

4.2.8 Discussion sur les fonctions d'ordre bivaluées

Au début de la section, nous avons pointé deux limites à l'utilisation d'un ordre d'intervalle : la compatibilité avec les comparaisons manuelles et le caractère mal posé du problème.
Peut-on dire si ces limites ont joué sur les performances ?

Il est possible de quantifier la compatibilité de l'annotation avec un ordre d'intervalle en comptant le nombre de sous-ensembles de type 2+2. Notre jeu de test en contient approximativement 1.500. Ces sous-ensembles sont en partie dus à l'existence de groupes naturels au sein desquels la comparaison est plus facile. Par exemple, parmi les paires de scènes enneigées, la comparaison est plus confortable qu'entre une scène enneigée et une scène non-enneigé.
La présence de ces sous-ensembles a nécessairement un impact sur les scores : si l'on suppose que ces sous-ensembles sont tous indépendants, 1.500 erreurs sont inévitables [10], qu'elles soient commises sur les arêtes de $\mathcal{G}^v_{test}$ ou sur les arêtes de $\mathcal{U}^v_{test}$. Mais elles restent marginales devant les erreurs de prudence et les rejets (8.000 erreurs environ).
D'ailleurs, les justesses atteintes à l'entraînement sur le problème à trois classes (plus de 90 %) montrent qu'une approximation par ordre d'intervalle peut couvrir la grande majorité des comparaisons portées sur une séquence. Elles montrent aussi que les réseaux de neurones utilisés sont assez profonds pour restituer ces intervalles (propriété d'expressivité, voir chapitre 1).

Par contre, la Prudence en test est encore faible (de l'ordre de 50 % pour un taux de rejet de 20 %). L'information contenue dans une image n'est peut-être pas suffisante pour régler correctement les tailles des intervalles, mais au vu des erreurs de Prudence commises, il reste encore une marge de progression importante. Il sera sans doute nécessaire de recourir à une nouvelle étape d'annotation manuelle pour mieux couvrir les situations qui génèrent le plus d'erreurs.

4.2.9 Conclusion sur les fonctions de rang bivaluées

Les meilleurs scores sur le jeu de test d'AMOSvv sont obtenus avec des fonctions de rang bivaluées apprises sur AMOSvvExt. Sur le problème à deux classes, cette amélioration n'est pas restreinte aux scènes routières (voir la figure 4.8). En particulier, les performances sur les scènes d'infoclimat ont été largement améliorées. L'intérêt des méthodes présentées dans cette section et dans la section précédente est ainsi pleinement justifié.

10. Remarquons que si des comparaisons inter-scènes avaient été utilisées, le nombre de sous-ensemble de type 2+2 aurait été beaucoup plus important. De même si les images de nuit avaient été incluses : il est en effet plus facile de comparer sur le critère de la visibilité deux images prises de nuit qu'une image prise de jour avec une image prise de nuit

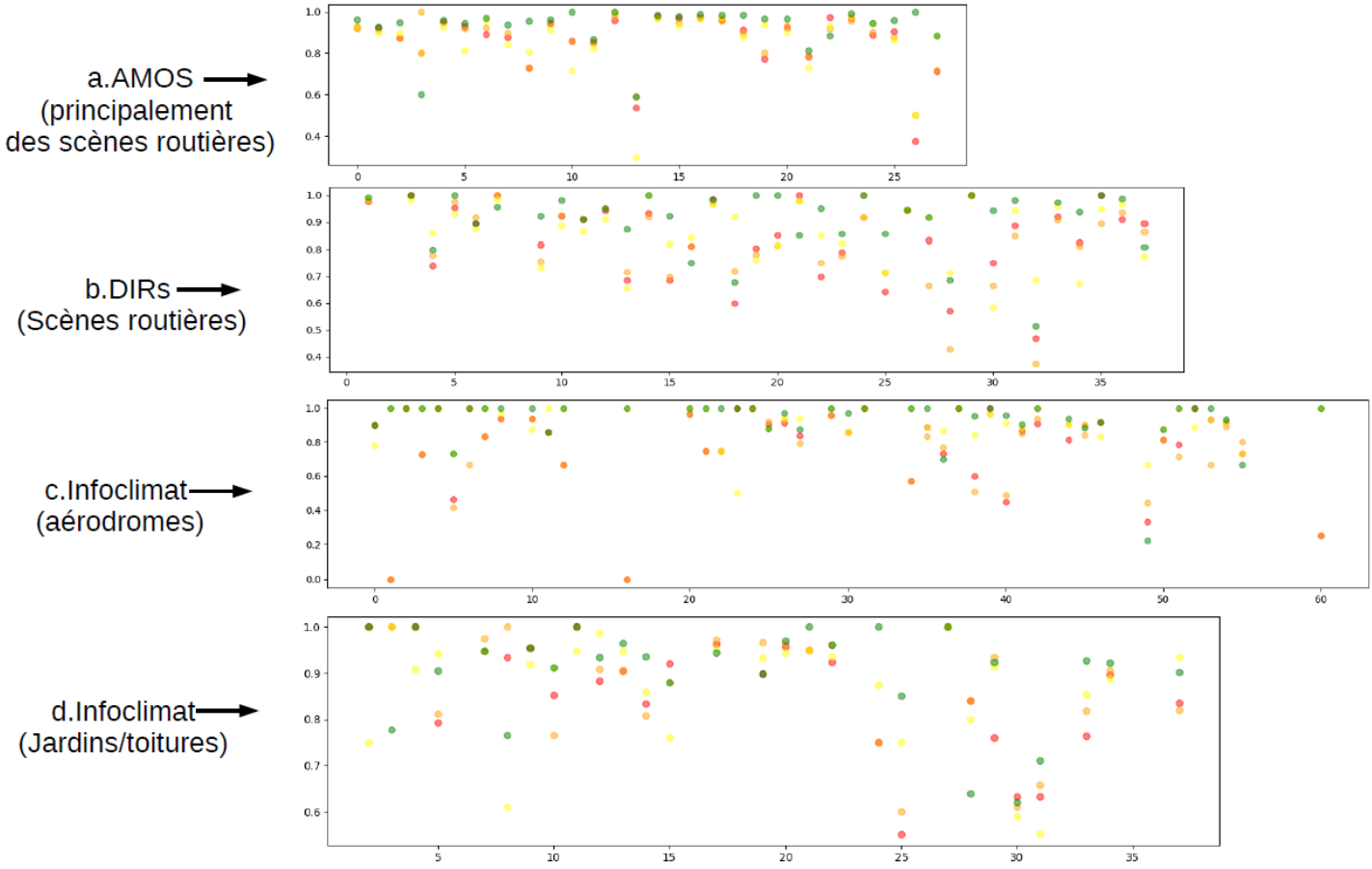


FIGURE 4.8 – Comparaison des performances en généralisation sur le problème à deux classes. Les points verts représentent les centres d'intervalles de mean_bivalued_gr1gr2. Les autres points représentent les valeurs prédites par les fonctions d'ordre entraînées sur AMOSvv (lignes 2-4 de la table 4.2). Lorsqu'une colonne est vide, c'est que les images de la séquence sont toutes incomparables deux à deux.

La restitution de la relation d'incomparabilité, tâche a priori plus difficile, a été abordée à travers une approche originale, sur laquelle le recul manque encore. Nos fonctions de rang bivaluées parviennent à saisir les causes d'incomparabilité et se montrent légèrement plus performantes qu'un simple mécanisme d'abstention dans le cas où l'incomparabilité tient à la mauvaise qualité d'une des deux images. Sur ce problème, la marge de progression semble cependant encore importante.
Après avoir pousser les performances des fonctions de rang, nous nous intéressons à leur étalonnage dans la section suivante.

4.3 Etalonnage des fonctions de rang

Dans les sections précédentes, nous avons montré comment améliorer des fonctions de rang pour le paramètre visibilité. Ces fonctions rang prédisent un réel ou un intervalle pour chaque image, sur une échelle arbitraire, propre au modèle et à la scène. Dans cette partie, nous cherchons à convertir ces prédictions de nature ordinale en des prédictions de nature quantitative.
Dans un premier temps, nous nous plaçons dans le cas idéal où des mesures colocalisées sont disponibles et où la fonction d'ordre bivaluée restitue parfaitement l'ordre contenu dans l'annotation. Nous nous interrogeons alors sur la conversion des intervalles de sortie en encadrements de la valeur de la visibilité.
Sous des hypothèses raisonnables, nous montrons qu'il est possible de passer des intervalles prédits par une fonction d'ordre bivaluée à des encadrements de la mesure à partir de l'image (section 4.3.1.1). Nous nous donnons ensuite une méthode d'étalonnage peu exigeante, basée sur la distribution locale des visibilités (sous-section 4.3.1.2). Cette méthode est appliquée à des caméras associées à des instruments de mesure (sous-section 4.3.1.4)) pour un premier bilan.

Dans le contexte de l'exploitation des données d'opportunité, on ne dispose pas des distributions locales exactes. Pour contourner ce problème, nous avons envisagé deux méthodes. La première, ébauchée à la fin de cette section, est fondée sur l'estimation de la distribution locale du paramètre à partir de données distantes. La seconde, basée sur un nouvel apprentissage, fait l'objet de la section suivante.

4.3.1 Etalonnage d'une fonction de rang bivaluée

La prédiction d'intervalles pose un problème spécifique : comment exploiter la correspondance entre les intervalles prédits et les mesures de visibilité ?
Il faut d'abord préciser la nature de cette correspondance. Dans le paragraphe qui suit, nous donnons des conditions sous lesquelles l'échelle des visibilités et celles des sorties du modèles sont liées par une bijection.
Cette propriété permet d'étalonner les intervalles prédits par les fonctions de rang bivaluées.

4.3.1.1 Encadrements de la visiblité :

Lorsque qu'une fonction de rang à valeurs réelles rend parfaitement compte de l'annotation, l'étalonner revient à trouver une correspondance (une bijection) entre le domaine $\mathcal{Z}$ des sorties du modèle et le domaine $\mathcal{D}_v = [v_{min}, v_{max}]$ des visibilités observables.

Pour nous ramener à cette situation avec des fonctions de rang bivaluées, nous nous plaçons dans un cadre idéal, et nous faisons trois hypothèses sur l'annotation. Considérons E^c l'ensemble des couples image-visibilité (x, v) observables avec une caméra fixe c. On suppose d'abord que les comparaisons émises par l'annotateur sont correctes (elles ne contredisent pas l'ordre des visibilités) et compatibles

avec un ordre d'intervalle. Nous supposons en outre que l'annotation vérifie deux hypothèses supplémentaires, cohérence et de séparabilité, fondées sur la définition suivante :

Definition 4 L'image x est dite indissociable de v si pour toute image x_v telle que $(x_v, v) \in E^c$, on a $x \perp x_v$. Sinon, x et v sont dits dissociables.

Comme l'annotation est correcte, les couples (x, v) de E^c sont des exemples de couples image-visibilité indissociables. La première hypothèse supplémentaire assure que ce sont les seuls :

Hypothèse 1 ***(cohérence)*** *Si l'image x est indissociable de v alors (x, v) est dans E^c.*

Selon la seconde hypothèse, les visibilités sont séparées par l'annotateur :

Hypothèse 2 ***(séparabilité)*** *Soient deux visibilités $v \neq v'$. Il existe une image x générée par la caméra c qui est indissociable de v et dissociable de v'.*

Sous les hypothèses supplémentaires précédentes, on montre (voir Annexe D-V) la propriété suivante :

Proposition 2 *Soit une fonction de rang bivaluée continue f^c qui conserve parfaitement l'ordre contenu dans l'annotation. Il existe une fonction strictement croissante g^c telle que, pour tout couple (x, v) de E^c, on ait $v \in [g^c(z^-), g^c(z^+)]$ où $[z^-, z^+] = f^c(x)$. Les intervalles $[g^c(z^-), g^c(z^+)]$ ne dépendent que de l'image x.*

Une telle fonction g^c est dite encadrante.
Ce résultat permet de donner un sens supplémentaire à la prédiction d'intervalles : dans des conditions idéales, les fonctions de rang bivaluées fournissent, à une bijection près, des encadrements de la visibilité.

4.3.1.2 Procédé d'étalonnage

Nous étalonnons par un procédé assez naïf, mais efficace, qui consiste à caler les quantiles des prédictions sur ceux des mesures (histogram matching [113]). L'intérêt de ce procédé est qu'il suffit de pouvoir estimer la distribution des visibilités sur le site pour pouvoir étalonner.
Cependant, avec les fonctions bivaluées, nous ne disposons pas d'une prédiction, mais d'un intervalle. Comment faire alors ? En exploitant la proposition 2, nous pouvons construire des encadrements de la visibilité plutôt que des estimations ponctuelles.
Prenons une caméra c, pour laquelle on dispose de longues séries temporelles d'images x_i (sur un hiver entier par exemple) et des prédictions $(z_i^-, z_i^+) = f(x_i, w)$ par une fonction de rang bivaluée qui conserve parfaitement l'ordre de l'annotation.
Considérons les images x_i comme les réalisations d'une variable aléatoires X. Notons $\mathcal{R}^c_{Z^-}$, $\mathcal{R}^c_{Z^+}$ et $\mathcal{R}^c_{Z_m}$ les fonctions de répartition (CDF) de la borne inférieure, de la borne supérieure et du centre de l'intervalle $f(X, w)$. Nous supposerons avoir suffisamment d'images pour pouvoir estimer ces CDF

avec une erreur négligeable. Notons aussi $\mathcal{R}_V^c$ la CDF de la visibilité V, qu'on suppose continue et strictement croissante, et notons $\mathcal{Q}_V^c$ sa réciproque (fonction quantile), définie sur $[v_{min}, v_{max}]$. On a alors :

Proposition 3 *Soit g^c, une fonction encadrante 2 associée à la fonction rang bivaluée $f(., w)$. Pour tout (x, v) de E^c, en posant $f(x, w) = [z^-, z^+]$ nous avons :*

$$\mathcal{Q}_V^c(\mathcal{R}_{Z^+}^c(z^-)) \leq g^c(z^-) \leq v \leq g^c(z^+) \leq \mathcal{Q}_V^c(\mathcal{R}_{Z^-}^c(z^+)) \tag{4.10}$$

On en déduit un intervalle « prudent » à partir des fonctions composées $\mathcal{Q}_V^c o \mathcal{R}_{Z^+}^c$ et $\mathcal{Q}_V^c o \mathcal{R}_{Z^-}^c$:

$$v_p^+ = \mathcal{Q}_V^c(\mathcal{R}_{Z^-}^c(z^+)) \tag{4.11}$$
$$v_p^- = \mathcal{Q}_V^c(\mathcal{R}_{Z^+}^c(z^-)) \tag{4.12}$$

Nous estimons aussi empiriquement la fonction encadrante à partir de la fonction de répartition $\mathcal{R}_{Z^m}^s$ des centres d'intervalle $z^m = \dfrac{z^- + z^+}{2}$:

$$v_f^+ = \mathcal{Q}_V^s(\mathcal{R}_{Z^m}^s(z^+)) \tag{4.13}$$
$$v_f^- = \mathcal{Q}_V^s(\mathcal{R}_{Z^m}^s(z^-)) \tag{4.14}$$

Cette seconde option fournira des intervalles plus fins, mais moins sûrs. La même fonction est utilisée pour donner une estimation ponctuelle de la visibilité :

$$v_m = \mathcal{Q}_V^s(\mathcal{R}_{Z^m}^s(z^m)) \tag{4.15}$$

Les fonctions composées des équations (4.11-4.15) apparaissent sur des diagrammes quantile-quantile. La figure 4.9 présente ces diagrammes pour quatre caméras du réseau TENEBRE. Pour les estimer, nous interpolons simplement les diagrammes quantile-quantile par des splines de degré 1. Les estimations résultantes seront notées, dans l'ordre, $\hat{v}_p^+$, $\hat{v}_p^-$, $\hat{v}_f^+$, $\hat{v}_f^-$ et $\hat{v}_m$.

Faute de temps, nous n'avons pas développé le cadre probabiliste dans lequel étudier les propriétés de ces estimations. Pour le faire, plusieurs pistes se présentaient dans la littérature.
Les intervalles de valeurs possibles peuvent, par exemple, être vus comme des « nombres flous ». La théorie des nombres flous aléatoires aborde la question de la définition et de l'estimation des CRF. Le lecteur intéressé pourra consulter les références [114], [115].
Une alternative consisterait à considérer un intervalle de valeurs possibles comme un intervalle de confiance associé à la distribution du paramètre conditionnellement à l'image. Cela peut par exemple

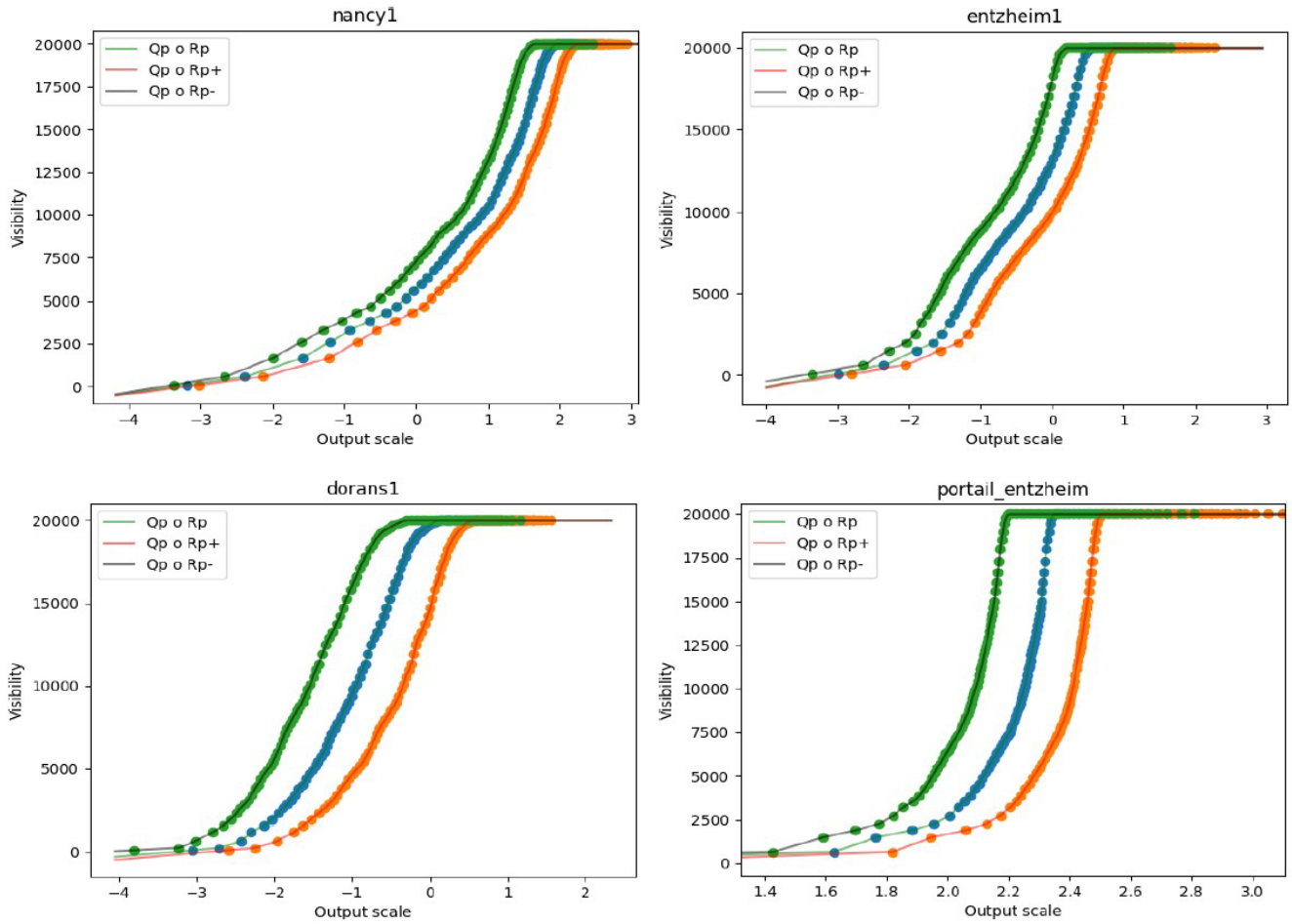


FIGURE 4.9 – Etalonnage des prédiction sur quatre caméras de TENEBRE. Les fonctions utilisées pour l'étalonnage des fonctions bivaluées sont estimées à partir de diagrammes quantile-quantile. Les coordonnées des points sont les centiles des séries de prédictions (abscisses) et les centiles des séries de visibilités (ordonnées) mesurées sur site pendant l'hiver 2012-2013 (octobre 2012 - mars 2013), de jour, à un pas de temps de dix minutes. Les prédictions ont été obtenues avec vv_sl_mean_gr1_gr2.

être fait à travers une modélisation de l'annotation avec un modèle de détection du signal (voir [116]).

4.3.1.3 Etalonnage avec les mesures colocalisées : l'exemple des caméras TENEBRE

Nous avons appliqué la méthode aux caméras du réseau TENEBRE implantées dans des stations météorologiques (voir chapitre 2).

Dans cette section, nous utiliserons la fonction de rang bivaluée vv_sl_mean_gr1_gr2, qui présente les meilleures performances en généralisation sur le jeu de test d'AMOSvv.

Sur les séquences de TENEBRE, les performances de cette fonction de rang sont irrégulières. Sur le jeu TENEBRE_IH (annoté à la main), la justesse sur le problème à deux classes varie de 80% pour la scène portail_entzheim, très atypique, à plus de 90% pour les scènes les moins atypiques (nancy1, nancy2, entzheim1, entzheim3, dorans1 et roissy). Nous en tiendrons compte pour juger du potentiel de la méthode.

Nous allons d'abord regarder les prédictions lorsque les quantiles de la visibilité sont obtenus à partir de

mesures colocalisées à l'instrument. L'utilisation de mesures distantes sera discutée dans les sections suivantes.

Etude qualitative :

Nous présentons d'abord une analyse qualitative des prédictions étalonnées sur quelques épisodes de brouillard et de chutes de neige, à partir des distributions locales du paramètre. Nous avons prêté attention à la cohérence temporelle des prédictions, à l'ajustement aux mesures, aux éventuels biais systématiques, à la taille des intervalles et aux cas d'erreur que le test sur AMOSvv n'avait pas révélés. Ces paragraphes qui suivent synthétisent ces observations. Le lecteur intéressé trouvera des éléments complémentaires dans l'annexe D, section D.5.4.

Pour cette analyse qualitative, l'étalonnage a été réalisé à partir des images prises de jour (9h - 16h) au cours des six premiers mois disponibles (octobre 2012 - mars 2013). La visibilité a été seuillée à 20 km.

Pour estimer les distributions, nous avons retiré les données associées aux prédictions suspectes et aux images de très mauvaise qualité. Les prédictions sont considérées comme suspectes en cas de « sursaut » : lorsque pour trois images consécutives (x_{k-1}, x_k, x_{k+1}) les prédictions impliquent une augmentation ($z^+_{k-1} < z^-_k$) puis une baisse ($z^-_k > z^+_{k+1}$), on retire les (z^-_k, z^+_k) de la série. Idem dans le cas d'un sursaut négatif. Ces sursauts ne sont pas systématiquement associés à des prédictions fausses. Mais sur les séries TENEBRE, le retrait des sursauts prévient l'effet des images parasites intercalées dans les séries (voir chapitre 3, section 3.3.5).

Pour déterminer les images de très mauvaise qualité, nous considérons les tailles des intervalles. Lorsque la taille de l'intervalle est supérieure au quantile d'ordre 0.9 (voir encadré 4.2), l'image est suspectée d'être de mauvaise qualité, et les données correspondantes sont retirées de la série.

Les courbes d'étalonnage sont illustrées figure 4.9. Pour les autres figures (figure A-8 et figures de l'annexe V-D), nous représentons les intervalles fins (équations 4.14-4.13).

La figure A-8 présente plusieurs cas de brouillard (18/11/2012) à Entzheim (aéroport de Strasbourg), Essey (station météorologique de Nancy) et à l'aéroport de Roissy. En dehors de quelques cas de surestimation, comme le 18/11 entre 14h et 16h sur la scène entzheim3, la cohérence temporelle des prédictions et l'ajustement sont très bons sur les deux premières scènes.

Sur ces caméras, on observe aussi souvent (mais pas systématiquement) de larges intervalles pendant et après le coucher du soleil, qui a lieu vers 15 h 45 UTC à Strasbourg à la mi-novembre. Ces larges intervalles sont donc liés à des conditions d'éclairage dégradées. Ils génèrent ici des inclusions légitimes.

Nous faisons les même constats (qualité des ajustements et élargissement au crépuscule) sur les scènes entzheim1 et nancy2.

ENCADRÉ 4.2 – Sélection et représentation des intervalles

Nous le disions dans la partie précédente, l'échelle des sorties d'une fonction bivaluée est arbitraire. Cependant, lorsque deux intervalles prédits ont le même centre, l'ordre dans lequel sont rangés les rayons n'est -idéalement- pas arbitraire.
Notons $\mathcal{F}_z$ l'ensemble des intervalles prédits dont le centre est proche de z à $epsilon$ près [11]. Dans $\mathcal{F}_z$, nous distinguerons les intervalles selon le rang de leur rayon.
Nous dirons qu'un intervalle $[z_k^m - r_k, z_k^m + r_k]$ de $\mathcal{F}_z$ est « petit » lorsque son rayon r_k est inférieur à la médiane des rayons dans ($r_k \leq q_{50}(\mathcal{F}_z)$). Ils sont représentés en jaune sur la figure A-8 et les figures de l'Annexe D, section D.5.4.
Les intervalles représentés en orange vérifient :

$$q_{50}(\mathcal{F}_z) < r_k \leq q_{90}(\mathcal{F}_z)$$

Ceux représentés en rouge vérifient :

$$q_{90}(\mathcal{F}_z) < r_k \leq q_{99}(\mathcal{F}_z)$$

Et ceux représentés en noir vérifient :

$$q_{99}(\mathcal{F}_z) < r_k$$

Les intervalles rouges et noirs ne sont pas pris en compte pendant l'étalonnage.

Sur les trois caméras neige_nancy, roissy et parc_entzheim, l'ajustement est moins bon et les estimations ponctuelles sont moins régulières. Les intervalles prédits contiennent plus rarement la vérité terrain. L'élargissement des intervalles en fin de journée est moins marqué. Cela n'évite pas les fautes de prudence.
L'estimation ponctuelle pour dorans1 est plutôt régulière et les fluctuations pour portail_entzheim sont très marquées.
Pour les épisodes de neige, un bilan analogue peut être tiré. Les variations de la prédiction et l'ajustement aux données sont globalement bons pour les caméras les moins atypiques.
Par exemple, pour les scènes nancy1 et nancy2, les seuls gros écarts sont observés lorsque la visibilité varie rapidement, comme lors d'averses de neige. Ces écarts peuvent s'expliquer par un défaut de d'horodatage (incertitude de $\pm 5mn$) ou par un défaut de représentativité de la mesure, mais il ne s'agit pas d'un défaut de prédiction.

Sur toutes les scènes, les masques météorologiques (flocons et gouttelettes contre la lentilles) sont fréquents pendant les épisodes de précipitations sur toutes les caméras. Sur la plupart des scènes, ces images de mauvaise qualité correspondent aux plus larges intervalles (encadré 4.2), comme sur le jeu de test d'AMOSvv. Mais sur les scènes neige_nancy et parc_entzheim, les masques, qui sur ces caméras se présentent généralement sous la forme de tâches floues, ne sont que rarement associés à de larges intervalles, couvrant les valeurs de visibilité parmi les plus basses. Ces prédictions persistent assez longtemps pour biaiser l'étalonnage.

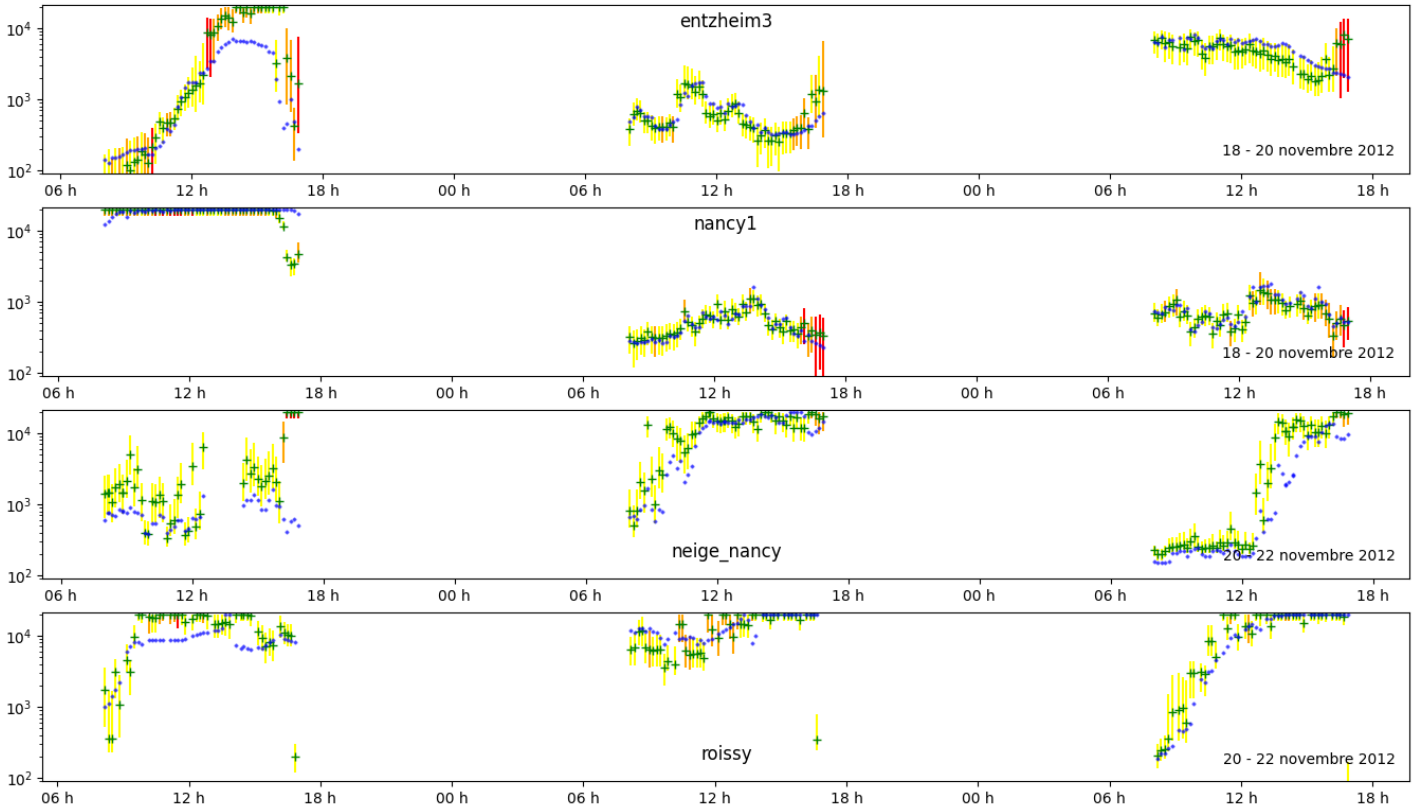


FIGURE 4.10 – Comparaison entre prédictions et mesures sur quatre scènes du réseau TENEBRE. La visibilité (en mètres) est en ordonnée, l'heure UTC en abscisse. Les graphiques couvrent des périodes de trois jours, entre 8 h UTC et 17 h UTC (alors que l'étalonnage ne fait intervenir que les images prises entre 9 h et 16 h) ; ils couvrent au moins une situation de brouillard (visibilité < 1000 m) par scène. Les estimations ponctuelles (croix vertes) sont obtenues par l'équation (4.15), les bornes des intervalles par les équations (4.14 - 4.13). Les grands intervalles sont représentés par les couleurs orange et rouge (voir encadrés 4.2). Les croix bleues représentent les mesures instrumentales

La taille des intervalles après étalonnage dépend aussi du type de scène. Sur la scène dorans1, sans second plan, ce sont des intervalles associés aux basses visibilités (inférieures à 1000 m) qui sont les plus larges, alors que sur la scène portail_entzheim, sans second plan, les intervalles sont très larges pour des visibilités supérieures à 5000 m). Ces observations sont cohérentes avec le comportement des modèles sur AMOSvv (voir section précédente et annexe D-V).

Des erreurs systématiques ont été repérées sur les caméras les mieux traitées. Des images de temps brumeux avec ciel bleu visible ont été associés à des visibilité supérieures à 10 km sur les scènes entzheim1 et entzheim3. De façon générale, les modèles semblent systématiquement associer aux ciels bleus des visibilités les plus élevées, commettant ainsi des erreurs potentiellement importantes pour des visibilités supérieures à 10 km.

Pour terminer, disons enfin que les erreurs lié à un défaut de représentativité de la mesure varient, en proportion, d'une scène à l'autre. Ils sont particlièrement fréquents pour la caméra de Roissy.

Etude quantitative :

Pour compléter cette analyse qualitative, les performances sont chiffrées dans le tableau 4.4.

Pour cette analyse, les paramètres de l'étalonnage (sélection des données, période, etc) sont les mêmes, sauf le seuil, passé à 10 km. Les images prises avant 9h et après 16h ne sont pas prises en compte dans l'évaluation ainsi que toutes les images associées à des sursauts, pour éviter l'effet des images parasites et celui des imprécisions d'horodatage.
Pour évaluer la prédiction d'intervalle, nous tenons compte d'une incertitude de $\pm 20\%$ sur la mesure. Le taux d'appartenance p est alors définit par la proportion d'intervalles $\hat{\mathcal{I}} = [\hat{v}^-, \hat{v}^+]$ tels que $\hat{\mathcal{I}} \cap [0.8 \times v, 1.2 \times v] \neq \emptyset$. On note p_p (resp. p_f) le taux d'appartenance aux intervalles des équations (4.12-4.11) (resp.des équations (4.14-4.13)).
Pour indiquer la taille des intervalles, nous utilisons l'écart relatif médian entre la borne inférieure et la borne supérieure. Nous les notons t_p et t_f, suivant le jeu d'équation utilisé. Rappelons qu'une incertitude de $\pm 20\%$ sur la mesure correspond à un écart relatif de $(1.2 - 0.8)/0.8 = 50\%$. Enfin, pour juger de la qualité des estimations ponctuelles, conformément à la littérature [37], nous utilisons l'erreur relative moyenne ($e_{moy.} = \overline{\frac{\hat{v}^m - v}{v}}$). Nous précisons aussi l'erreur relative médiane ($e_{med.}$), moins sensible aux valeurs extrêmes. Comme ces valeurs extrêmes peuvent être causées par les mesures non représentatives, la médiane nous semble mieux adaptée.

Les taux d'appartenance des intervalles prudents sont élevés (96 % en moyenne, ligne 2 de la table 4.4), mais la borne supérieure des intervalles est en moyenne dix fois supérieure à la borne inférieure. Sur l'intervalle 100 m - 10.000 m, l'encadrement « prudent » ne permet donc pas de distinguer plus de trois classes de temps sensible sur la majorité de ces scènes.
L'encadrement fin est moins juste avec un taux d'appartenance moyen de 0.91 % (ligne 4), mais nettement plus resseré (écart relatif de 225 %).
Notons que les taux d'appartenance sont sensibles au seuil considéré : les taux d'appartenance sont jusqu'à 15 points plus faibles pour des images associées à une visibilité inférieure à 5000 m (lignes 3 et 6). Les taux les plus bas sont atteints sur les scènes particulièrement atypiques (neige_nancy et portail_entzheim), alors que les les tailles d'intervalles sont parmi les plus élevées.
Quant à l'estimation ponctuelle (lignes 8-13), l'écart moyen avec la mesure est de l'ordre de 60 %, et l'écart médian est de 30 %. Ces écarts sont plus importants lorsqu'on ne considère que les visibilités mesurées inférieures à 1000 m [12] (lignes 10 et 13), mais la hausse est plus nette pour l'écart moyen (ligne 10) en partie à cause des mesures non représentatives.

Enfin, les performances sont légèrement dégradées lorsque l'on étalonne les fonctions de rang sur la période du octobre 2017 - mars 2018 avant des les appliquer à la période d'intérêt (ligne 14). Cela tient à l'écart temporel important entre les deux sous-périodes d'échantillonnage : en quatre ans, les caméras ont légèrement bougé.

12. Ces chiffres sont calculés sur 1048 cas de mesure inférieure à 1000 m.

Scène	nancy1	nancy2	neige_nancy	entzheim1	entzheim3	parc_entzheim	portail_entzheim	roissy	dorans1	Moyenne
p_p(10km)	0.97	0,98	0,95	0,98	0,98	0,93	0,93	0,94	0,99	0,96
p_p(5km)	0.92	0,92	0,69	0,93	0,95	0,73	0,7	0,84	0,96	0,85
t_p(10km)	1,74	5,44	8,6	5,81	5,8	2,8	10,8	7,5	37	9,5
p_f(10km)	0,95	0,95	0,91	0,94	0,95	0,87	0,85	0,85	0,96	0,91
p_f(5km)	0,81	0,83	0,66	0,84	0,85	0,65	0,62	0,68	0,89	0,76
t_f(10km)	0,67	1,47	2,79	1,77	1,52	0,72	2,19	1,3	3,73	1,8
$e_{moy.}$(10km)	0,26	0,32	0,74	0,45	0,45	0,59	1,05	0,64	0,68	0,58
$e_{moy.}$(5 km)	0,37	0,43	1,65	0,91	0,92	1,21	2,42	1,04	1,15	1,12
$e_{moy.}$(1 km)	0,35	0,24	2,35	3,99	3,97	4,25	8,23	3,2	3,26	3,32
$e_{med.}$(10km)	0,18	0,22	0,37	0,19	0,2	0,27	0,44	0,34	0,29	0,28
$e_{med.}$(5 km)	0,21	0,26	1,06	0,27	0,29	0,47	1,31	0,5	0,5	0,54
$e_{med.}$(1 km)	0,19	0,14	1,01	0,37	0,29	0,46	3,57	1,2	2,36	1,07
$e_{med.}^{17-18}$(10km)	0,19	0,26	0,42	0,23	0,2	0,31	0,38	0,34	0,32	0,29

TABLE 4.4 – Performances sur le jeu TENEBRE après étalonnage. Pour les prédictions, la fonction de rang bivaluée a été étalonnée avec les mesures colocalisées. Les lignes 2 et 5 indiquent les taux d'appartenance et les tailles médianes des intervalles « prudents » et « fins » (équations (4.12-4.13)). Pour les lignes 3-6, ce taux est calculé après restrictions aux visibilités mesurées inférieures à 5000m. Les lignes 8-13 contiennent les erreurs relatives moyennes associées aux prédictions $\hat{v}^m$ (équation (4.15)), avec ou sans restriction sur la visibilité mesurée.

4.3.1.4 Etalonnage avec des mesures distantes

Pour pouvoir appliquer notre méthode sur des caméras géolocalisées quelconques, il faut pouvoir estimer la fonction quantile des visibilités locales.

Or, si d'un site à l'autre la dispersion des mesures est très importante, les quantiles peuvent être en revanche très proches. C'est par exemple le cas des quantiles estimés à Nancy et Entzheim (figure 4.11.a-b) sur la même période (10/2012 - 03/2013).

En estimant la fonction quantile au site de Nancy à partir des mesures faites à Entzheim, nous induisons un biais (figure 4.11.b) relativement faible, de l'ordre de quelques points sur l'erreur médiane (table 4.5).

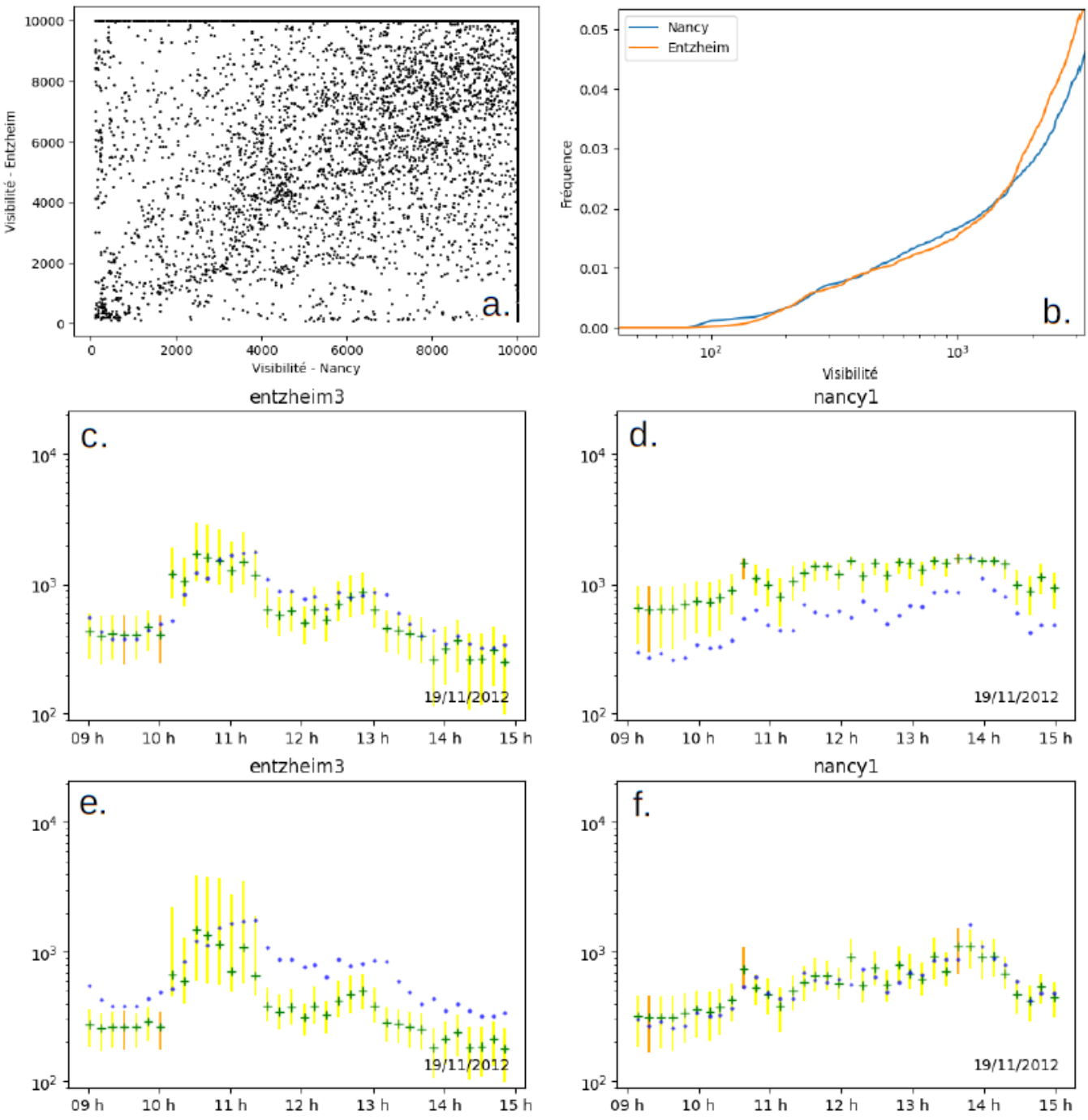


FIGURE 4.11 – a.Dispersion des visibilités mesurées à Entzheim et Nancy.b.CDF estimées sur la même période (oct 2012 - mars 2013), pour Enzheim et Nancy.c.Prédictions du 19/11 sur la scène entzheim3 après calage sur les quantiles de visibilité mesurées à Entzheim.d. Série des prédictions sur la scène nancy1, le même jour, après calage sur les quantiles mesurés à d'Entzheim.

Nous nous sommes assurés que ce qui vaut pour les sites de Nancy et d'Entzheim, séparés de 100 km, vaut encore sur l'ensemble du territoire métropolitain à l'exception des zones de montagne. Pour le savoir, nous nous nous sommes intéressés à la variabilité des quantiles de visibilité en fonction de la distance entre les points de mesure. Pour cette étude, très préliminaire, nous avons sélectionné des stations météorologiques du réseaux Radome. Pour éviter les effets de l'altitude sur la distribution, nous nous sommes restreints à des stations situées en plaine. Sur 112 de ces stations, nous avons extrait des

étalonnage	Métrique (seuil : 10 km)	nancy1	nancy2	neige_nancy	entzheim1	entzheim3	parc_entzheim	portail_entzheim	dorans1	roissy	Moyenne caméras de Nancy	Moyenne caméras d'Entzheim
local	p_f	0,95	0,95	0,91	0,94	0,95	0,87	0,85	0,85	0,96	0,94	0,90
	$e_{med.}$	0,18	0,22	0,37	0,19	0,2	0,27	0,44	0,34	0,29	0,26	0,28
site de Nancy	p_f	/	/	/	0,93	0,93	0,87	0,84	0,79	0,92	/	0,89
	$e_{med.}$	/	/	/	0,22	0,24	0,33	0,46	0,58	0,37	/	0,31
site d' Entzheim	p_f	0,93	0,94	0,9	/	/	/	/	0,78	0,95	0,92	/
	$e_{med.}$	0,23	0,24	0,4	/	/	/	/	0,45	0,29	0,29	/

TABLE 4.5 – Effets d'un étalonnage sur des mesures distantes sur les scores (taux d'appartenance p_f et erreur relative médiane). Pour les lignes 2,3, l'étalonnage utilise les mesures colocalisées (ce sont les lignes 5 et 11 de la table 4.4). Lignes 4,5, l'étalonnage utilise les visibilités mesurées à Nancy. Lignes 6,7, l'étalonnage utilise les visibilités mesurées à Entzheim.

séries de mesures de visibilité expertisées sur une période de trois mois (hiver 2012-2013), au pas de temps horaire, acquises de jour. Ce jeu de données est nommé RADOMEvv1213.

Pour représenter la variabilité des fonctions quantiles, nous nous limitons à quelques points de contrôle. Pour chaque série de mesure, nous calculons ainsi une liste de *quantiles de référence* fixée. Cette liste commence par le quantile d'ordre 0.011 qui est associé en moyenne à une visibilité de 300 m (lignes grises et noires sur la figure 4.12). Nous déterminons ensuite les écarts relatifs entre ces quantiles pour des paires de stations distantes de d km, où $d \in [k-25, k]$. Pour construire la figure 4.12, nous faisons varier k de 50 km à 150 km, avec un pas de 25 km.

En ordonnée, nous indiquons la médiane, le quantile d'ordre 0.75 et le quantile d'ordre 0.9 des séries d'écart relatifs pour chaque choix de k. Par exemple, pour l'ensemble des 166 paires de stations distantes de 25 km à 50 km, nous indiquons la médiane, le quantile d'ordre 0.75 et celui d'ordre 0.9 des 166 écarts relatifs pour chaque *quantile de référence* choisi.

Au delà du quantile associé à une visibilité moyenne de 1500 m (lignes vertes), la distribution des écarts se resserre. C'est donc pour les plus faibles visibilités que le biais sur la prédiction risque d'être le plus élevé : dans un cas sur dix l'utilisation d'une station distante de plus de 75 km peut induire un biais systématique supérieur à 300 % sur les faibles valeurs de visibilité. Le biais médian est, lui, sensiblement plus faible. A bien choisir la ou les stations distantes à partir desquelles on estime les quantiles, il semble raisonnable d'espérer de réduire le biais systématique à moins de $\pm 100\%$ d'erreur relative et de le contrôler.

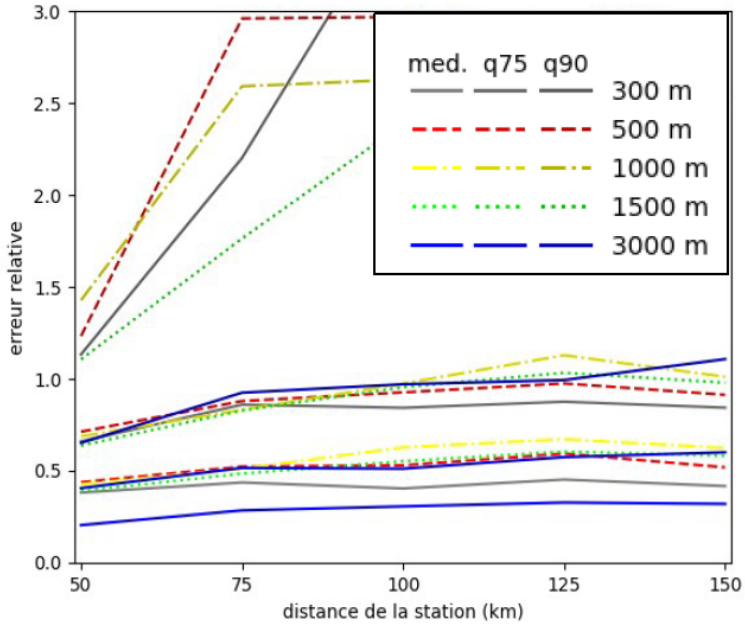


FIGURE 4.12 – Distribution des erreurs relatives commises sur la prédiction en utilisant une station distante. La distribution est représentée par la donnée de la médiane (courbes claires, en bas), du troisième quartile et du neuvième décile (courbes foncées, en haut). Ces trois paramètres sont donnés pour des visibilités de l'ordre de 300 m (courbes noires) à 3000 m (courbes bleues). Par exemple, l'erreur relative médiane est la plus faible (20 %) pour une visibilité de l'ordre de 3000 m et une station à moins de 50 km.

Le choix des stations les mieux adaptées, le choix de la méthode d'estimation des CDF et l'encadrement du biais relèvent des géostatistiques et dépassent le cadre de cette thèse. A l'adresse d'un lecteur qui souhaiterait prendre le relais, nous laissons à disposition des séries de prédictions sur plus d'une dizaine de webcams (annexe numérique) françaises échantillonnées pendant l'hiver 2020-2021.

4.3.1.5 Conclusion

Dans cette section, nous avons cherché à étalonner des fonctions de rang bivaluées. Nous avons d'abord vu que, sous quelques hypthèses portant sur les qualités de l'annotation et de la fonction de rang utilisée, passer des prédictions à des encadrements de la visibilité avait du sens. C'est ce passage que nous appelons « étalonnage » des fonctions de rang bivaluées. Nous nous sommes placés dans ce cadre pour étalonner notre meilleure fonction de rang bivaluée sur les scènes du jeu TENEBRE_1218. L'étalonnage est réalisé par histogram matching, et n'utilise que de la distribution locale des visibilités. Sur les caméras les mieux traitées, ce procédé simple nous permet d'atteindre des encadrements relativement fins (écart relatif médian entre les deux bornes de moins de 200 %) qui contiennent la visibilité dans au moins 90% cas.

Nous nous sommes aussi servis des centres d'intervalles pour obtenir des estimations ponctuelles. L'erreur relative médiane (par rapport à la mesure) associé à l'estimation ponctuelle est de l'ordre de

20% sur les caméras les mieux traitées, et passe à environ 50% lorsque la visibilité est inférieure à 5000 m. Mais sur les autres caméras, ou lorsque l'on considère le quantile d'ordre 0.9, elle avoisine ou dépasse 100 %.

Dans le cadre d'une application à des caméras d'opportunité, aucune mesure locale n'est disponible. Nous avons évalué dans cette section une première approche pour un étalonnage des fonctions de rang à partir de mesures distantes. Cette approche consiste à estimer la distribution locale du paramètre à partir de mesures distantes.

Cette estimation introduit un biais systématique dans les prédictions, mais ce biais pourrait être contrôlé, sinon réduit par des méthodes de géostatistiques adaptées.

Par contre, pour appliquer cette approche, il faut disposer de séries assez longue pour construire des courbes d'étalonnage. Une deuxième contrainte vient de la dérive des prédictions due aux mouvements de la caméra sur le long terme. Pour la prévenir, il serait nécessaire de renouveler l'étalonnage régulièrement.

Dans la section suivante, nous présentons une approche alternative de l'étalonnage sur des mesures distantes.

4.4 Intercalibration par apprentissage

Dans la section précédente, nous avons présenté une première approche réaliste de la prédiction quantitative de la visibilité. Suivant cette approche, l'étalonnage des modèles est basé sur une estimation de la distribution locale des visibilités à partir de mesures distantes.
Une approche alternative consiste à étalonner un modèle sur une caméra de référence une bonne fois pour toutes, et à appliquer ce modèle sur les caméras d'intérêt.
Mais il y a un obstacle : le caractère relatif à la scène des prédictions fournies par nos fonctions de rang. En effet, faute d'avoir appris des comparaisons inter-scènes, les valeurs (ou les intervalles) en sortie de modèle ne sont comparables que si les images d'entrée proviennent de la même scène.

Dans cette dernière section, nous montrons qu'il est possible d'apprendre une échelle ordinale indépendante de la scène, sans passer par une nouvelle phase d'annotation. L'approche consiste en une régression ordinale faiblement supervisée, que nous appellerons intercalibration. Dans cette section, nous en expliquons le principe et nous présentons des résultats sur les scènes du jeu TENEBRE.

4.4.1 Méthode

L'idée principale est d'utiliser le rang d'une image dans sa séquence comme cible de l'apprentissage. Précisons d'abord le problème d'apprentissage dans le cadre idéal où l'annotation est compatible avec un ordre total et où l'on dispose pour chaque caméra c d'une fonction de rang à valeurs réelles idéale $f^c(.;w)$ qui trie parfaitement les images dans l'ordre des visibilités croissantes. On suppose aussi que le problème de l'estimation quantitative à partir de l'image est bien posé : l'image contient assez d'information pour déterminer la valeur exacte de la visibilité.

Supposons enfin qu'une série temporelle d'images webcam peut être vue comme l'échantillonnage de l'image aléatoire X définie par :

$$X = \mathcal{G}(V, x^c, X^t) \tag{4.16}$$

où V est la visibilité, x^c représente la partie « permanente » de l'image (paramètres de la caméra, géométrie de la scène, route, maisons, reliefs, etc) et X^t représente la partie transiente, aléatoire de l'image (autres variables météo, véhicules, éclairage, etc). La variable V a une fonction de répartition qui dépend du site sur lequel la webcam est installée, notée $\mathcal{R}_V$, et la quantité $\mathcal{R}_V(V)$ définit le rang de V dans la séquence. Ce rang est aussi celui de l'image dans la séquence triée par f^c. Avec des séries temporelles arbitrairement longues, il peut être estimé à travers l'équation :

$$\mathcal{R}^c_Z(f^c(X, w)) = \mathcal{R}_V(V) \tag{4.17}$$

Où $\mathcal{R}_Z^c$ est la fonction de répartition de $Z = f^c(X, w)$ sur la caméra c.
Nous pouvons maintenant définir un problème d'apprentissage à partir des couples $(x_i, \mathcal{R}_V(v_i))$. Pour comprendre pourquoi ce problème est intéressant, complétons la modélisation probabiliste.

Considérons que choisir une image x dans un ensemble de séquences webcam revient à choisir une scène au hasard, à travers le choix de x^p, et une fonction de répartition $\mathcal{R}_V$, puis à échantillonner suivant l'équation (4.16). Notons X^p et R_V les variables aléatoires correspondantes. Au cours d'un apprentissage, le modèle serait entraîné à prédire y^*, défini par :

$$y^* = \underset{\{y \in \mathbb{R}\}}{\arg\min}\, \mathbb{E}(\mathcal{L}(y - R_V(V)) \mid X = x \ ) \tag{4.18}$$

Où $X = \mathcal{G}(V, X^c, X^t)$ et où $\mathcal{L}$ représente la fonction de coût.

Si l'on exclut les scènes qui apportent une information sur la distribution locale de la visibilité -les scènes de montagne par exemple- on peut considérer que la fonction aléatoire R_V est indépendante de X^p et X^t. Comme le problème est bien posé, l'équation 4.18 se ramène à :

$$y^*(v) = \underset{y}{\arg\min}\, \mathbb{E}(\ \mathcal{L}(y - R_V(V)) \mid V = v \ ) \tag{4.19}$$

Pour une fonction de coût quadratique, le modèle est donc entraîné à restituer la quantité y^* :

$$y^*(v) = \mathbb{E}(\ \mathcal{R}_V(V) \mid V = v) \tag{4.20}$$

Noter que y^* n'est pas connue. Ce type d'approche, entraîner sur des cibles centrées sur une quantité d'intérêt inconnue sous une hypothèse d'indépendance, a déjà été exploitée avec succès pour le débruitage d'image [117]. C'est l'illustration d'une stratégie d'apprentissage faiblement supervisée [82].
Ce qui, dans notre cas, fait l'intérêt de la quantité $y^*(v)$, c'est qu'elle est théoriquement **indépendante** de la scène et **croissante** par rapport à la visibilité v. Cette deuxième affirmation n'a rien d'évident. Elle est basée sur l'expérience qui suit.

4.4.1.1 Monotonie de $y^*(v)$:

Reprenons les 112 séries temporelles de visibilité du jeu RADOMEvv1213 contenant des séries de visibilités mesurées. Pour chacune de ces séries, on estime les fonctions de répartition par une méthode d'estimation non-paramétrique standard [118]. Les solutions au problème d'optimisation (eq. (4.19)) pour des fonctions de coût quadratiques (MSE) et linéaires (MAE), c'est à dire les moyennes et médianes conditionnelles, sont croissantes sur l'intervalle [100 m - 10.000 m] (figure 4.13).

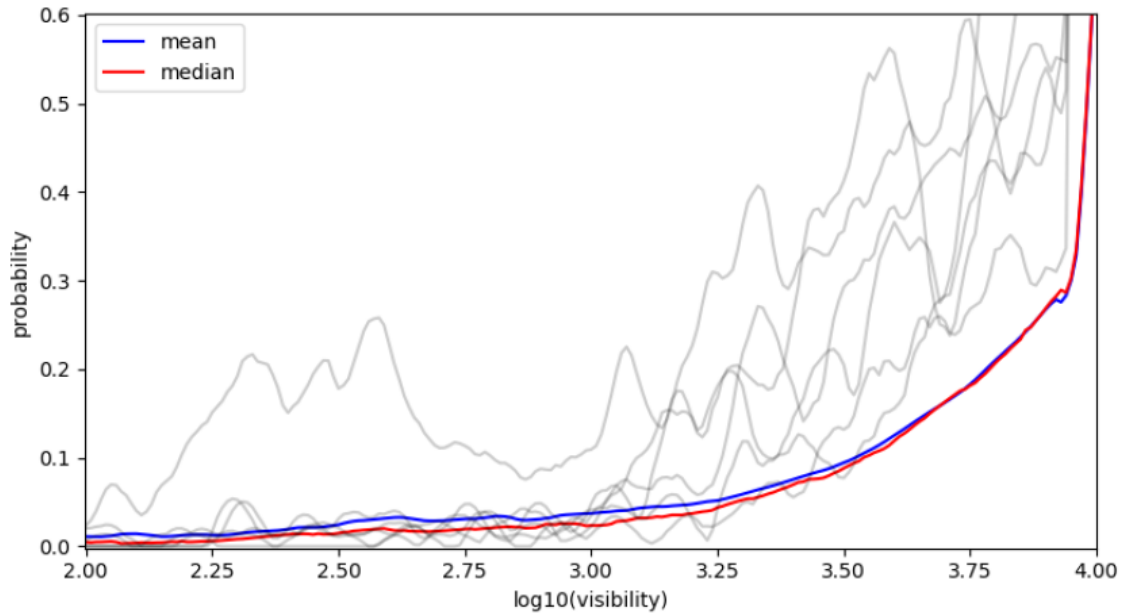


FIGURE 4.13 – Variations de $y^*(v)$, solution du problème d'optimisation défini par l'équation (4.19) dans le cas où la fonction de coût est la MSE (courbe bleue) ou la MAE (courbe rouge). à partir des mesures de RADOMEvv1213. Les courbes grises de la figure correspondent aux distributions de la visibilité associées à dix séries prises au hasard.

4.4.1.2 Généralisation au cas des ordres d'intervalle :

Nous avons cherché à adapter cette approche à une fonction d'ordre bivaluée. Replaçons-nous donc dans le cadre idéal de la section précédente. L'existence d'une fonction encadrante g et la conservation de l'ordre d'intervalle par $f^c(.;w)$ sur la caméra c impliquent [13] que :

$$\mathcal{R}^c_Z(Z^+) = \mathcal{R}_V(V^+) \qquad \mathcal{R}^c_Z(Z^-) = \mathcal{R}_V(V^-) \tag{4.21}$$

Où $[Z^-, Z^+] = f^c(X;w)$, $V^+ = g(Z^+)$ et $V^- = g(Z^-)$ et $\mathcal{R}^c_Z$ désigne la fonction de répartition de $g^{-1}(V)$ et $\mathcal{R}_V$ désigne la fonction de répartition de V sur le site d'installation.

La difficulté ici, c'est que la fonction $\mathcal{R}^c_Z$ n'est pas connue. Pour la contourner, nous supposerons qu'il est possible de l'estimer par ${\mathcal{R}^c}_{Z^\pm}$, définie par :

$${\mathcal{R}^c}_{Z^\pm} = \frac{{\mathcal{R}^c}_{Z^-} + {\mathcal{R}^c}_{Z^+}}{2} \tag{4.22}$$

Sous cette hypothèse, nous pouvons entraîner un modèle à résoudre :

$$(y^{*,+}, y^{*,-}) = \underset{\{(y^+,y^-)\in\mathbb{R}^2\}}{\arg\min} \; \mathbb{E}(\; \mathcal{L}(y^- - R_V(V^-), y^+ - R_V(V^+)) \;|\; X = x \;) \tag{4.23}$$

13. En réalité, ces égalités ne sont pas nécessairement vérifiées aux bords de l'échelle des visibilités. Ce problème n'a pas été pris en compte.

où $X = \mathcal{G}(V, X^p, X^t)$ et R_V est la fonction aléatoire dont les réalisations sont les $\mathcal{R}_V$.

De nouveau, on suppose que le problème est bien posé, c'est à dire que pour chaque image x, l'intervalle $[v^-, v^+]$ fourni par la proposition 2 contenant la vraie valeur v est identifiable [14]. En d'autres termes, on suppose que l'image contient une information parfaite sur sa propre ambiguïté. Toujours sous l'hypothèse que la fonction R_V est indépendante de X^p et X^t, on a pour la MSE :

$$y^{*,-} = \mathbb{E}(\ \mathcal{R}_V(v^-)\ \mid\ v^-, v^+) \tag{4.24}$$

$$y^{*,+} = \mathbb{E}(\ \mathcal{R}_V(v^+)\ \mid\ v^-, v^+) \tag{4.25}$$

Il s'agit de savoir si ces quantités induisent un ordre d'intervalle plus grossier, sinon égal, à celui défini par les v^-, v^+. Comme ces bornes ne sont pas connues, nous les avons simulées à partir des valeurs de visibilités des séries de RADOMEvv1213 présentées plus haut et pour différentes statistiques de bruits.

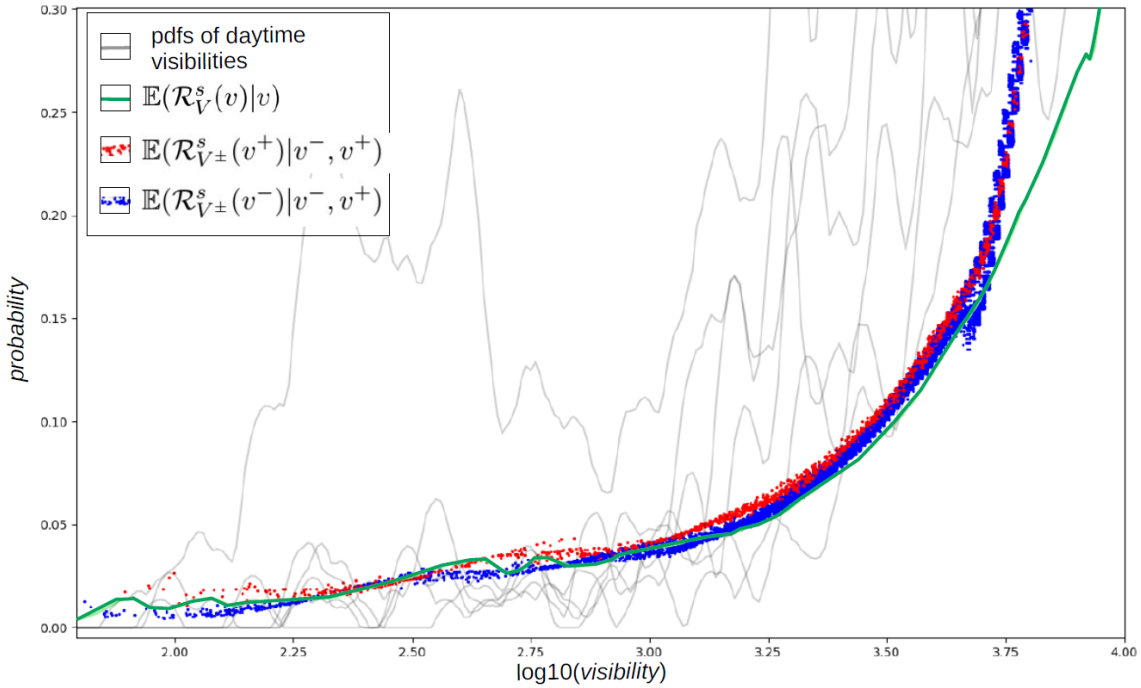


FIGURE 4.14 – Variations de $y^{*,-}(v)$ (courbe rouge) et $y^{*,+}(v)$ (courbe bleue), solutions du problème d'optimisation définies par l'équation (4.23) défini à partir des 112 séries de mesures de RADOMEvv1213.

Par exemple, pour obtenir la figure 4.14, les écarts $v - v^-$ et $v^+ - v$ suivent des lois uniformes dans $[0, 0.3]$ (après passage à l'échelle logarithmique). Les points rouges et bleus correspondent alors aux $y^{*,-}$ et $y^{*,+}$. Sur certaines plages, les y^+ sont manifestement plus grands (et les y^- plus petits) que ce qu'ils devraient être pour que l'ordre d'intervalle soit conservé. Néanmoins, même s'il est plus grossier, il n'implique pas de discordance ou d'erreur de prudence supplémentaires.

14. Sauf, éventuellement, en marge du support de distribution des visibilités, mais voir la note précédente.

Le même constat peut être fait pour des amplitudes plus importantes ou des intervalles non centrés.

Ces obervations nous ont conduit à faire un essai sur un ensemble de séries temporelles d'images issues d'AMOS et des DIRs. Les scènes du jeu de validation n'ont pas été utilisées de façon à pouvoir contrôler l'apprentissage. La construction des séquences, la construction des cibles avec, en particulier, les étapes de détection et de correction des erreurs systématiques sont détaillées dans l'annexe D, section D.6. De même pour l'augmentation de la donnée et la formation des mini-lots.

4.4.1.3 Résultats sur des scènes de TENEBRE

Sur ce jeu de séries temporelles pré-triées, nous apprenons de nouveaux réseaux de neurones à couches de convolution. Pour chaque image x_i de la série associée à la caméra c, les cibles sont définies par les couples $(\mathcal{R}^c{}_{Z^{\pm}}(z_i^-), \mathcal{R}^c{}_{Z^{\pm}}(z_i^+))$. Faute de disposer de comparaisons inter-scènes, l'étape de validation est faite sur les comparaisons intra-séquence du jeu de validation d'AMOSvv. Comme dans les sections précédentes, nous sélectionnons l'architecture associée aux meilleures performances en validation. Nous présenterons ici les résultats obtenus avec un ResNet50 noté cal0103.

La comparaison entre cal0103, et le modèle d'où l'annotation a été tirée (vv_sl_due111.0) a été conduite sur le sous-ensemble de la base TENEBRE_I (voir annexe B) comportant environ 5.000 images par scène. Ce jeu concentre les situations météorologiques qui nous intéressent le plus (brouillard, précipitation et neige au sol).
Connaissant la fragilité des résultats sur les caméras neige_nancy, roissy et parc_entzheim, nous avons limité l'analyse aux cinq scènes nancy1, nancy2, entzheim1, entzheim3 et dorans1. Par contre, ces scènes ont été recadrées suivant deux ou trois modalités différentes. Par exemple, deux nouvelles séquences sont formées à partir de nancy1, la première en rognant les 30 % du haut de l'image, la seconde en rognant les 30 % du bas. Pour les scènes d'Entzheim et Dorans, à horizon bas, seuls le rognage haut est pratiqué.
Pour comparer les modèles, nous les avons étalonnés selon deux modes différents. Suivant le premier mode, il est réalisé à partir des mesures colocalisées. Suivant le second mode, il est réalisé sur une scène, et les courbes d'étalonnage obtenues sont appliquées pour toutes les autres scènes.
Dans les deux cas ce sont les équations (4.14 - 4.15) qui sont appliquées après filtrage des images de mauvaise qualité, comme dans la section précédente.

Nous nous sommes intéressés en priorité à la dérive des estimations ponctuelles. Les deuxième et troisième lignes de la table 4.6 permettent de comparer les erreurs relatives médianes en cas d'étalonnage local. Les prédictions du modèle appris par intercalibration sont généralement moins bonnes. On en trouve la confirmation avec des taux d'appartenance diffèrent peu (3 points d'écart en moyenne) alors que les écarts relatifs médian (t_f(10km), lignes 14-15) du modèle appris par intercalibration sont supérieures de plus de 30 points.
Par contre, le modèle cal0103 peut être étalonné sur une scène distante sans que les prédictions varient

étalonnage	modèle	nancy1 0%	nancy1 30% haut	nancy1 30% bas	nancy2 0%	nancy2 30% haut	nancy2 30 % bas	entzheim1 0%	entzheim1 30% haut	entzheim3 0%	entzheim3 30% haut	dorans1 0%	dorans1 30% haut	Moyenne
local	sl	0.20	0.2	0.18	0.15	0.16	0.16	0.23	0.23	0.23	0.23	0.33	0.26	0.21
	cal	0.28	0.25	0.25	0.25	0.22	0.22	0.22	0.2	0.32	0.27	0.36	0.42	0.27
nancy2 0 %	sl	0.20	0.2	0.34	0.37	0.34	0.54	0.59	0.60	0.75	0.8	0.74	0.77	0.52
	cal	0.28	0.22	0.25	0.27	0.26	0.31	0.33	0.27	0.43	0.34	0.34	0.43	0.31
entz.1 0%	sl	0.3	0.39	0.27	0.2	0.2	0.36	0.23	0.27	0.27	0.72	0.51	0.67	0.39
	cal	0.27	0.31	0.24	0.31	0.29	0.31	0.22	0.19	0.32	0.25	0.35	0.4	0.29
$\overline{e_{med}}$(10km)	sl	0.52	0.58	0.44	0.36	0.36	0.33	0.39	0.37	0.37	0.46	0.34	0.42	0.41
	cal	0.31	0.34	0.3	0.29	0.28	0.28	0.29	0.28	0.32	0.3	0.33	0.36	0.31
$\overline{e_{med}}$(5km)	sl	0.57	0.62	0.54	0.42	0.42	0.42	0.58	0.56	0.84	1.12	0.83	1.09	0.67
	cal	0.49	0.48	0.54	0.31	0.33	0.32	0.6	0.52	0.68	0.56	0.53	0.67	0.5
$\overline{e_{q90}}$(5km)	sl	0.92	0.87	1.04	0.97	0.96	1.18	1.4	1.37	2.2	2.96	2.06	2.75	1.6
	cal	1.76	1.71	1.81	1.41	1.66	1.72	2.32	2.15	2.34	1.84	1,3	1,8	1,8
local (t_f(10km))	sl	0,49	0,44	0,51	0,3	0,23	0,23	0,64	0,97	0,48	0,98	1,7	1,85	0,74
	cal	1,06	0,76	1,01	0,39	0,32	0,4	1,39	1,01	1,31	0,96	2,48	1,85	1,08

TABLE 4.6 – Table des erreurs relatives après seuillage à 5000 m. Les lignes paires (2-10) contiennent les résultats du modèle entraîné sur des paires intra-séquence à partir desquelles les cibles ont été obtenues (vv_sl_due111.0), les lignes impaires (3-11), ceux du modèle entraîné par intercalibration (cal0103). Les lignes 2-3 contiennent les erreurs relatives médianes après étalonnage sur les mesures locales. Pour les lignes 4-5 (resp. 6-7) les modèles ont été étalonnés sur la scène nancy1 (resp. entzheim1). Dans les ligne 8-9, on indique la moyenne des erreurs relatives médianes sur l'ensemble des scènes après étalonnage sur la scène associée à la colonne. Les lignes 10-13, on donne les erreurs relatives (médiane ou neuvième décile) associées à des visibilités inférieures à 5 km. Les deux dernières lignes de la table indiquent la taille relative des intervalles après étalonnage local.

beaucoup (figure 4.15, deuxième ligne). Par comparaison, lorsque sl_vv_due111.0 est étalonné sur nancy1 plutôt que sur la scène d'intérêt, les prédictions sont sytématiquement biaisées.

Cette relative stabilité des prédictions de cal0103 explique celle des erreurs relatives médianes (première ligne de la figure 4.15). Cette stabilité est observable quelle que soit la scène utilisée pour l'étalonnage (comparer les lignes 8 et 9 de la table 4.6). De ce fait, cal0103 apparaît nettement plus avantageux lorsque l'étalonnage a lieu sur une scène de référence, et ce même si le modèle sl_vv_due111.0 est meilleur sur une tâche de comparaison intra-séquence.

Cet avantage se conserve lorsque le calcul de l'erreur médiane est restreint aux visibilités inférieures à 5000 m (lignes 9-10). Par contre, si l'on considère le quantile d'ordre 0,9, le résultat s'inverse (lignes 11-12), probablement du fait de la plus grande dispersion du nouveau modèle.

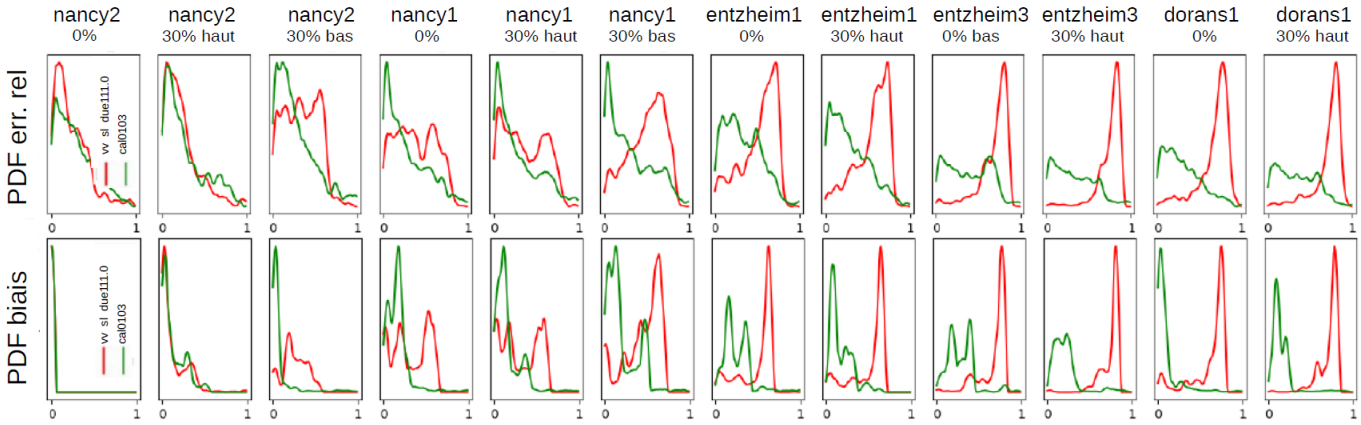


FIGURE 4.15 – Les courbes rouges sont relatives au modèle entraîné sur les paires intra-séquence (vv_sl_due111.0), les vertes sont associées au relatives au modèle entraîné par intercalibration (cal0103). Ligne 1 : distribution des écarts relatifs à la mesure dans le cas où le modèle est étalonné sur nancy2 - 0%. Ligne 2 : distributions des écarts relatifs entre les prédictions selon la modalité d'étalonnage (étalonnage local ou étalonnage sur nancy1).

4.4.1.4 Discussion sur l'intercalibration :

Ce résultat en demi-teinte peut tenir en partie à des défauts de construction du jeu d'apprentissage. En particulier, la procédure d'échantillonnage des séquences d'AMOS [15] a dû modifier sensiblement la distribution « naturelle » des fonctions de répartition. Il serait donc intéressant de reprendre ce travail avec des séquences de tailles similaires, cas dans lequel nous nous sommes placés pour justifier la méthode (section 4.4.1.1).

De plus, même si un post-traitement a été réalisé, les erreurs systématiques commises par les fonctions de rang bivaluées (voir sections 4.2.7 et 4.3.1.3) sur certaines scènes ont dû fausser les fonctions de répartition.

A cet égard, il aurait peut-être été plus judicieux de reconstruire les ordres d'intervalle à partir de la prédiction par paires, en suivant la méthode présentée dans l'annexe D, section D.4.2. Néanmoins, nous savons pour l'avoir observé que les fonctions de comparaison présentent encore des cas d'erreur de prudence (voir section 4.2). Aussi, nous suggérons de ne reprendre cet essai que lorsque les erreurs de prudence auront été corrigées par l'annotation de séquences supplémentaires.

Enfin, la méthode d'apprentissage pourrait, elle aussi être améliorée. En particulier, nous n'avons pas utilisé de fonction d'activation pour la dernière couche. A posteriori, l'utilisation d'une fonction logistique paraît mieux adaptée [119]. L'idée d'aborder les deux tâches, estimation relative par paires et régression ordinale, à travers un apprentissage séquentiel, nous paraît aussi intéressante.

15. Cette procédure, définie au chapitre 2, consistait à concentrer l'échantillonnage au cours des périodes de neige pour faciliter la construction de jeux équilibrés. Cet avantage se transforme en inconvénient si l'on mise sur la proximité entre les fonctions de répartition.

4.4.2 Conclusion sur l'approche par intercalibration

Dans cette section, nous avons présenté une approche alternative de l'étalonnage sur les données distantes. Nous avons cherché à dépasser la principale limite des fonctions de rang bivaluées apprises sur les paires intra-séquence : a priori, on ne peut pas étalonner ces fonctions sur une caméra de référence pour les appliquer à de nouvelles caméras.
Nous avons donc cherché à apprendre un modèle qui le puisse, sans effort d'annotation supplémentaire.
L'approche présentée ici consiste à apprendre un tel modèle par une régression ordinale faiblement supervisée. Cet apprentissage exploite les prédictions de fonctions de rang bivaluées préalablement entraînées. Le rang de l'image dans une séquence est vue comme la version bruitée d'une quantité qui croît avec la visibilité. Cette forme d'apprentissage a débouché sur des modèles moins performants sur les paires intra-séquence. Mais en contrepartie, ces modèles peuvent être étalonnés sur une scène de référence et appliqué sur d'autres scènes pour un coût limité en terme d'erreur relative médiane.
Comme l'approche présentée dans la section précédente, cette intercalibration par apprentissage présente un potentiel intéressant. Mais comme nous n'étions ni en mesure de pousser chacune de ces approches à leur maximum, ni en mesure de les tester sur un jeu de caméras plus large que TENEBRE, nous n'avons pas cherché à les comparer. Nos résultats suggèrent simplement que, dans un cas comme dans l'autre, qu'une erreur relative médiane de moins de 50 % est un objectif réaliste.

4.5 Conclusion et perspectives

Dans ce chapitre, nous avons cherché à améliorer et à étalonner des fonctions de rang, dans l'idée de produire des estimations quantitatives de la visibilité. Quatre méthodes ont été proposées. Les deux premières sont génériques et visent à améliorer les fonctions de rang, tant au plan quantitatif que qualitatif. Leur apport a été évalué sur un grand nombre de scènes (jeu de test d'AMOSvv). Les deux dernières sont plus spécifiques à la visibilité et ne sont pas encore tout à fait abouties. Elles visent à étalonner les prédictions à partir de données disponibles en pratique. Les résultats sont encourageants, mais ils n'ont pu être évalués que sur les quelques scènes du réseau TENEBRE.
L'amélioration des fonctions de rang a été présentée en première partie. La première méthode peut être vue comme un exemple de distillation semi-supervisée [81]. Elle a consisté à exploiter la prédiction par paire, plus performante, pour annoter automatiquement des séquences AMOS supplémentaires. Apprises sur un jeu ainsi étendu (AMOSvvExt), les fonctions de rang se montrent plus performantes pour la restitution des comparaisons strictes.
La seconde méthode est fondée sur une extension simple du concept de fonction de rang. A l'origine, ces dernières induisent une relation d'ordre total sur les images. Pour pouvoir apprendre et restituer la relation d'incomparabilité contenue dans l'annotation, nous avons introduit les fonctions de rang bivaluées, qui prédisent des intervalles et induisent sur les images d'une séquence un ordre d'intervalle.
Après apprentissage sur AMOSvvExt, la taille de l'intervalle reflète à la fois la qualité de l'image et la

difficulté de la scène : les images dégradées par des gouttelettes, des flocons collés à la lentille, les cas de contrejour et les images floues, les scènes sans second plan ou, au contraire, sans premier plan sont ainsi associées à des intervalles de plus grande taille. Au plan quantitatif, l'utilisation de fonctions de rang bivaluées permet un léger gain en prudence (meilleur restitution de l'incomparabilité) par rapport à l'utilisation d'un mécanisme de réjection simple. Elles se montrent aussi plus performantes pour la restitution des comparaisons strictes.

Dans les deux dernières sections, nous avons abordé le problème de l'estimation quantitative de la visibilité sur des scènes indépendantes. Nous l'avons abordé à partir de fonctions de rang apprises sur les comparaisons intra-séquence.
Dans la troisième section, nous avons précisé un cadre dans lequel étalonner des fonctions de rang bivaluées faisait sens et nous nous sommes donnés un procédé d'étalonnage simple (histogram matching), basé sur la distribution locale des visibilités. Ce procédé débouche sur des encadrements plus ou moins fins de la visiblité et sur une estimation ponctuelle.
Ce procédé a d'abord été évalué en utilisant les distributions de mesures colocalisées aux images. En l'absence de base météo-image publique, l'évaluation a été réalisée sur neuf des scènes du jeu TENEBRE, assez différentes des scènes routières du jeu d'entraînement initial. Sur les scènes les mieux traitées, les séries temporelles des estimations ponctuelles sont proches de la mesure (erreur relative médiane inférieure à 50%) et les encadrements proposés affichent des taux d'appartenance élevés (plus de 80 %).
Mais dans la pratique, nous ne disposons pas de mesures colocalisées. Pour contourner le problème, nous avons considéré deux approches alternatives. La première consiste à estimer la distribution locale à partir de mesures distantes. Une étude réalisée sur une centaine de stations métropolitaines suggère qu'il est possible de limiter l'erreur sur l'estimation des quantiles locaux à $\pm 50\%$ pour des caméras situées en plaine.
Une alternative consiste à étalonner le modèle une bonne fois pour toutes sur une caméra associée à de la donnée colocalisée. Nos fonctions de rang, apprises sur des comparaisons intra-paires, ne s'y prêtaient pas a priori. Nous avons donc réappris un modèle capable de prédire des comparaisons inter-scènes sans annotation supplémentaire, par une régression ordinale faiblement supervisée. Cette méthode débouche sur des modèles moins précis en terme de comparaison intra-paires. Mais comparés aux fonction de rangs précédentes, ils sont relativement plus robustes face à un changement de scène : sur les scènes qui ressemblent le plus aux scènes d'AMOSvv, l'erreur relative médiane sur une scène indépendante atteint 30 % (contre 41 %) pour des visibilités seuillées à 10 km et 50 % (contre 67 %) pour des visibilités inférieures à 5 km. Par contre, probablement à cause des erreurs plus fréquentes, le quantile d'ordre 0.9 s'est dégradé (180 % en moyenne contre 160 %).

Que tirer de ces résultats pour le suivi d'épisodes de neige en plaine ? Peut-on par exemple envisager d'estimer un taux de précipitations neigeuses [16] qui soit utile au prévisionniste ? Disons d'abord

16. snowfall rate. Selon l'instrument de mesure, il est exprimé en mm/h (pluviomètre chauffé) ou en g/dm/h (pesée)

que le lien entre la visibilité et l'intensité des chutes de neige est complexe. En particulier, la température joue sur la microphysique et, en définitive, sur le coefficient d'extinction du milieu. Même à température et visibilité fixées, la dispersion des taux de précipitations neigeuses reste importante [88]. Néanmoins, il reste possible de décider d'une classe d'intensité parmi les trois ou quatre classes en usage : chutes de neige « fortes », « modérées », « légères » et « très légères » [17]. Pour distinguer ces classes de neige, il faut au minimum pouvoir séparer les visibilités moyennes associées à chacune de ces classes. Or, dans la littérature, ces visiblités moyennes sont dans un rapport compris en 3/2 et 2, pour des visibilités inférieures à 5 km. On doit ainsi commencer à apporter de l'information utile à partir du moment où l'erreur relative descend sous les 100 % pour des visibilités de moins de 5 km.
Avons-nous atteint ce seuil ? Cela dépend de la métrique utilisée et des scènes considérées. La vision optimiste revient à considérer l'erreur relative médiane sur les scènes qui ressemblent le plus aux scènes d'AMOSvv comme représentatives des performances générales. Mais à considérer l'ensemble des scènes avec le quantile d'ordre 0.9, le diagnostic est tout différent.
A notre avis, compte tenu de la taille modeste du jeu de départ, il reste une importante marge de progression. En particulier, une nouvelle campagne d'annotation devrait permettre :
- d'améliorer la prudence sur plusieurs phénomènes assez mal représentés dans notre jeu comme les basses visibilités associées à un ciel bleu, le passage du soleil à faible distance angulaire de la ligne de visée et les masques défocalisés.
- d'améliorer les performances sur des scènes atypiques comme celles de parc_entzheim, nancy_neige et portail_entzheim. Ajoutons que la construction d'AMOSvvExt et celle des séquences pour la régression ordinale faiblement supervisée pourront être réalisées sur une part plus grande du jeu AMOS.
De cette manière, nous pouvons espérer améliorer significativement ces premiers résultats et reprendre la question d'une classification du taux de précipitations neigeuses mieux armés. Mais si les applications directes sont encore prématurées, la connaissance même approximative de la visibilité peut nous être utile indirectement : dans le chapitre suivant, nous l'utiliserons pour augmenter l'annotation sur les deux autres paramètres d'intérêt, l'étendue et l'épaisseur du manteau neigeux.

17. c'est une classification grossière du taux de neige qui correspond aux besoins de la météorologie opérationnelle, en particulier dans le domaine de la prévision aérienne. On distingue en général les fortes chutes de neige (> 2.5 mm/h) des chutes modérées (2.5 - 1 mm/h) des chutes légères (< 1mm/h). On peut ajouter une denière classe [91] à 0.4 mm/h.

Chapitre 5

Application à la caractérisation de la neige en plaine

Dans ce chapitre, nous nous intéressons à la caractérisation du manteau neigeux. Deux paramètres ont été annotés (voir chapitre 2) : l'étendue et l'épaisseur du manteau. Nous abordons la caractérisation de ces paramètres sous un angle moins méthodologique. Il s'agit plutôt d'appliquer les méthodes présentées dans les chapitres précédents, d'en montrer les apports et d'en évaluer l'intérêt dans une perspective opérationnelle.
C'est par ailleurs sur ces paramètres que les premiers essais d'apprentissage par paires ont été réalisés. Nos premières pédictions sur des scènes indépendantes étaient relativement bien corrélées à l'épaisseur de neige mesurée mais les cas d'erreur restaient nombreux. Le jeu de test n'était pas suffisamment riche pour en rendre compte de manière satisfaisante.
Après l'annotation par paires non-consécutives, la prédiction par paire et la prédiction par image se sont améliorées, mais les fausses détections nombreuses, en particulier en dehors des épisodes de neige. Pour cette raison, nous avons décidé d'une nouvelle étape d'annotation semi-automatique. Cette troisième étape d'annotation, évoquée au chapitre 2, est détaillée dans l'annexe I-A.3. Elle a permis d'agrandir le jeu relatif à l'étendue.
Dans ce chapitre, nous considérerons directement les jeux disponibles à l'issue de cette troisième étape, i.e. AMOSss.1 (pour l'étendue) et AMOSsd.0 pour la hauteur. A partir de ces jeux les méthodes de base présentées au chapitre 3 peuvent être mises en oeuvre. Comme pour la visibilité, nous comparons les approches entre elles, et avec l'existant.

Dans la littérature, les approches par apprentissage relatives à la neige sont tournées vers la classification neige au sol/non neige au sol [35], [34]. Mais dans le domaine des sciences de l'environnement, des descripteurs d'image simples et efficaces ont été développés pour suivre l'évolution quotidienne du manteau neigeux par caméra fixe [5], [120], [109]. Nous utilisons ces descripteurs pour obtenir un score de base.
La comparaison entre les méthodes du chapitre 3 donne aussi le moyen d'évaluer la qualité des apprentissages : un modèle bien entraîné à la prédiction par paires doit a priori être plus performant qu'une

fonction de rang, même pré-entraînée, parce qu'il bénéficie d'une information plus complète. Nous vérifierons que, comme au chapitre précédent, nous nous retrouvons bien dans cette situation.
Nous poserons aussi la question de la séparation entre les deux paramètres, étendue et épaisseur. D'abord, parce que c'est un problème a priori difficile. En effet, l'étendue et l'épaisseur de la couverture neigeuse sont généralement corrélées. Ce critère permet donc de juger de la qualité de l'apprentissage. Cette capacité à séparer peut aussi être recherchée pour des raisons pratiques. Si l'on veut pouvoir estimer une épaisseur de neige indépendamment des fluctuations en étendue causées par le passage des véhicules, le dégagement et le salage de la voirie pendant les épisodes de neige, la bonne séparation des deux paramètres est essentielle.

Après cette évaluation, la méthode semi-supervisée développée au chapitre 4 est appliquée à l'étendue du manteau neigeux.
Les fonctions de rang résultantes sont à leur tour évaluées, là encore selon plusieurs modalités : il s'agit non seulement de savoir si une plus-value existe, mais aussi de pouvoir faire un bilan plus global sur la progression effectuée depuis les premiers essais en classification et de statuer sur l'intérêt des prédictions pour le suivi des épisodes de neige en plaine.

Le chapitre est organisé comme suit. Dans la première section, les jeux de données sont présentés et la méthode de scoring est légèrement remaniée. Dans la deuxième section, nous reprenons les méthodes du chapitre 3 et du chapitre 4 (fonctions de rang bivaluées). La procédure d'apprentissage est décrite dans ce qu'elle a de spécifique. Les différents modèles sont comparés et nous évaluons la capacité de la prédiction par paire à séparer l'étendue et l'épaisseur.
Dans la troisième section, nous décrivons comment l'approche semi-supervisée développée dans le cas de la visibilité est remaniée et mise en oeuvre pour l'estimation de l'étendue. Nous analysons ensuite la plus-value sur le jeu de test d'AMOSss.
Dans la quatrième section, les fonctions de rang sont évaluées sur la base TENEBRE_1218. Nous profitons de ce jeu resté à l'écart des apprentissages pour comparer les fonctions de rang aux classifieurs entraînés au problème binaire neige au sol/non neige au sol présenté au début du chapitre 3.
Dans la cinquième section, ces modèles sont appliqués à des séquences échantillonnées pendant l'hiver 2020/2021 à partir des caméras d'infoclimat et des DIRs. L'analyse est centrée sur les délais des prédictions de tenue de neige au sol.
Dans la dernière section nous donnons un bref aperçu des travaux encore en cours.

5.1 Jeux d'apprentissage et problèmes d'apprentissage associés

Dans cette section nous présentons les jeux de données (paires d'images comparées à la main) relatifs à l'étendue et à la profondeur du manteau neigeux. Dans un second temps nous précisons la procédure d'évaluation sur les jeux de test.

5.1.0.1 Jeux d'apprentissage

Le jeu de données relatif à l'étendue du manteau neigeux est nommé AMOSss.1 (« ss » pour « snow surface »). La troisième étape d'annotation (cf. chapitre 2) a permis d'étendre considérablement le jeu d'entraînement. Pour la comparaison, nous précisons la taille du jeu d'entraînement avant cette étape (ligne 2). En particulier, le nombre de scènes en jeu a triplé (table 5.1).
Sans revenir sur le détail de la troisième étape, donné dans l'annexe A-I, il est utile de préciser la nature de ces données supplémentaires. Il s'agit principalement de paires d'images incomparables ou équivalentes. Le nombre de scènes a certes triplé, mais sur une grosse partie des scènes supplémentaires, il n'y a aucune comparaison stricte.

Le dédoublement des jeux de validation n'a pas été réalisé dans le cas de la neige. Les 360 caméras du jeu d'entraînement d'AMOSss.0, qui se retrouvent dans AMOSss.1, rassemblent les caméras du jeu d'entraînement et celles du jeu VAL_{same} d'AMOSvv (voir chapitre 3, table 3.5), et comptent aussi quelques séquences AMOS supplémentaires. Celles du jeu de validation sont principalement issues du jeu VAL_{indep} d'AMOSvv. Pour le jeu de test, les caméras sont les mêmes -ce qui permet des analyses croisées. Le jeu de test pour l'étendue contient davantage de comparaisons strictes qu'AMOSvv (24.765 contre 15.007), mais si l'on exclut les paires contenant une image sans neige, les effectifs sont comparables (14.507, dernière ligne de la table 5.1).

	jeu	séquences	images	comparaisons strictes		incomparabilités		équivalences	
				graphe	arêtes	graphe	arêtes	graphe	arêtes
	$TRAIN_0$	360	9.588	$\mathcal{G}_0^s$	255.838	$\mathcal{U}_0^s$	43.472	$\mathcal{E}_0^s$	20.647
AMOSss.1	$TRAIN$	1.331	39.769	$\mathcal{G}_1^s$	268.144	$\mathcal{U}_1^s$	56.756	$\mathcal{E}_1^s$	57.702
	VAL	66	1.435	$\mathcal{G}_{val}^s$	11.457	$\mathcal{U}_{val}^s$	5.092	$\mathcal{E}_{val}^s$	2.715
	$TEST$	166	2.620	$\mathcal{G}_{test}^s$	24.765	$\mathcal{U}_{test}^s$	8.813	$\mathcal{E}_{test}^s$	4.529
	$TESTsnow$	-	-	$\mathcal{G}_{test}^{s>0}$	14.507	$\mathcal{U}_{test}^{s>0}$	5.118	$\mathcal{E}_{test}^{s>0}$	834

TABLE 5.1 – Description d'AMOSss.1, jeu de base pour l'apprentissage par paire des comparaisons relatives à l'étendue du manteau neigeux. Dans la seconde ligne, on indique les effectifs avant la troisième étape d'annotation (jeu d'entraînement de AMOSss.0). La dernière ligne est obtenue après restriction aux paires d'images sur lesquelles de la neige apparaît.

Le jeu AMOSsd.0 (« sd » pour snow depth) contient l'ensemble des relations disponibles à l'issue de la seconde étape d'annotation. Les caméras des différents jeux d'AMOSsd.0 sont les mêmes que celles d'AMOSss.0. Lorsque l'on compare les effectifs d'AMOSsd.0 (table 5.2) à ceux d'AMOSss.0, une remarque peut être faite : les équivalences et les incomparabilités représentent une plus grande proportion des paires annotées dans AMOSsd.0. Cela tient en partie au fait que les variations d'épaisseur apparentes sont plus rares.

Le jeu AMOSsd.0 n'a pas fait l'objet d'une étape d'annotation supplémentaire [1]. Comme pour le jeu précédent, les caméras utilisées dans les jeux de validation et de test sont les mêmes que dans AMOSvv.

	jeu	séquences	images	comparaisons strictes		incomparabilités		équivalences	
				graphe	arêtes	graphe	arêtes	graphe	arêtes
AMOSsd.0	$TRAIN$	360	9.588	$\mathcal{G}_0^d$	137,874	$\mathcal{U}_0^d$	53,005	$\mathcal{E}_0^d$	21,056
	VAL	66	1.435	$\mathcal{G}_{val}^d$	11.234	$\mathcal{U}_{val}^d$	5.630	$\mathcal{E}_{val}^d$	2.735
	$TEST$	166	2.620	$\mathcal{G}_{test}^d$	19.988	$\mathcal{U}_{test}^d$	10.080	$\mathcal{E}_{test}^d$	4.497
	$TESTsnow$	-	-	$\mathcal{G}_{test}^{d>0}$	6.730	$\mathcal{U}_{test}^{d>0}$	6.385	$\mathcal{E}_{test}^{d>0}$	802

TABLE 5.2 – Description d'AMOSsd.0, jeux de base pour l'apprentissage par paire des comparaisons relatives à l'épaisseur du manteau neigeux. La dernière ligne est obtenue après restriction aux paires d'images sur lesquelles de la neige apparaît.

5.1.0.2 Précision sur le scoring

Comme pour le paramètre visibilité, nous nous intéresserons à la restitutition des incomparabilités. Les causes d'incomparabilité, détaillées dans l'annexe I-A, ne sont cependant pas tout à fait les mêmes. Dans la grande majorité des cas, une incomparabilité n'exprime pas un doute de l'annotateur, mais la certitude que la paire ne peut pas être comparée, soit du fait de la mauvaise qualité de l'image, soit que la répartition de la neige sur le sol diffère sensiblement d'une image à l'autre.
De ce fait, nous pourrons compter les fautes de prudence avec les discordances. C'est fait à travers la « précision » P_{so} , définie par :

$$P_{so} = \frac{C}{C + D + F} \tag{5.1}$$

1. Cependant, un jeu un jeu AMOSsd.1 existe, dans lequel toutes les comparaisons faites à la troisième étape, pourtant relatives à l'étendue, ont été versées. Pour les équivalences et les incomparabilités, c'est justifié. Pour les comparaisons strictes, c'était sous l'hyptothèse, souvent fausse, qu'au bout de chaque séquence classée croissante, l'épaisseur apparente a cru elle aussi. Cet ajout n'a eu aucun effet positif sur les scores et nous n'en parlerons pas plus dans cette section.

où C est le nombre de paires strictement ordonnées par le modèle et par l'annotateur, D, le nombre de discordances (prédiction de $\dot{\prec}$ au lieu de $\dot{\succ}$ ou de $\dot{\succ}$ au lieu de $\dot{\prec}$) et F, le nombre de fautes de prudence (prédiction de $\dot{\prec}$ ou $\dot{\succ}$ au lieu de $\dot{\perp}$).
Nous utiliserons le diagramme Pso-completeness comme principal moyen de comparaison. Les diagrammes correctness-completeness, prudence-completeness et les performances sur le problème à deux classes seront utilisées de façon complémentaire.

5.2 Résultats de base

5.2.1 Procédures d'apprentissage et nomenclature

Les mêmes méthodes d'apprentissage qu'avec la visibilité ont été mises en oeuvre pour caractériser l'étendue et l'épaisseur du manteau neigeux. La procédure d'apprentissage est modifiée à la marge. Ce paragraphe précise ces modifications. D'abord, le rognage n'est pas pratiqué de la même façon. La zone proche du bord inférieur, qui couvre la partie de la scène la plus proche de la caméra, contient souvent des informations utiles à la comparaison. Nous avons cherché à la préserver en restreignant l'amplitude du rognage en bas de l'image.
Sur le paramètre "étendue du manteau", nous avons entraîné des fonctions de rang bivaluées avec ou sans tâche d'identification spécifique aux paires équivalences. En validation, les scores atteints étaient légèrement plus faibles avec la tâche supplémentaire. Nous avons donc abandonné ce dispositif pour l'apprentissage de l'étendue. Sur l'autre paramètre les deux tâches ont été conservées.

Contrairement au cas de la visibilité, nous n'avons pas pris le temps d'une longue étape de sélection de modèle : nous nous sommes contentés de choisir les architectures dans les catégories sélectionnées pour la visibilité : des VGG pour la prédiction par paire (VGG11, VGG13 et VGG16) et des ResNet50 pour la prédiction par image. Pour la prédiction par paires, la fréquence de présentation des arêtes incomparables (p_u) est de $1/3$. Pour l'entraînement des fonctions de rang bivaluées, nous avons choisi comme paramétrisation [2] : $p_g = 60\%$, $p_u = 20\%$ et $p_e = 20\%$.
La nomenclature des modèles est précisée dans l'annexe C. Notons seulement que lorsque deux paramètres apparaissent dans le nom du modèle (vvss ou vvsssd), c'est que le modèle a été entraîné en multitâche sur un jeu hybride entre AMOSss.1, AMOSsd.0, AMOSvv. Comme ces apprentissages ne se sont pas révélés plus fructueux, ils ne sont pas présentés ici (mais voir l'annexe D, section D.3 pour des détails).

5.2.1.1 Point de comparaison

Pour s'assurer de l'intérêt des méthodes du chapitre 3, nous utilisons des points de comparaison basés sur l'intensité du canal bleu. Ce canal est utilisé dans [5], [109], pour la segmentation sémantique à deux classes (neige au sol/non neige au sol). La méthode a été développée pour des caméras fixes couleur, haute résolution, situées en montagne. Elle est basée sur le fait que, dans des conditions d'éclairage identiques, le canal bleu d'un pixel est toujours plus intense quand l'élément de sol couvert par le pixel est enneigé [5].
La prédiction neige/non neige par pixel est obtenue par seuillage autour de l'intensité 0.5 (128/256). Cette méthode fonctionne bien sur des images prises par beau temps [5].
En nous inspirant de cette méthode simple, nous avons construit plusieurs fonctions de rang. Pour les

2. C'est un intermédiaire entre les deux paramétrisations associées aux groupes 1 et 2, de la deuxième section du chapitre 4.

premières, on compte le nombre de pixels ayant passé le seuil de 0.5 sur le canal bleu de l'image, en se restreignant au bas de l'image (entre les quinze et cinquante premiers pourcents pour la fonction blue_5015, dans les 20 % - 60 % pour la fonction blue_6020).
La troisième fonction (blue_sum) somme les intensités du canal bleu sur toute la partie basse de l'image (15%-50%), sans seuillage préalable. La dernière (blue_5015_m.sum) somme les intensités des pixels qui ont passé le seuil de 0.5.
Ce point de comparaison ne sera utilisé que pour l'étendue.

5.2.1.2 Résultats pour l'étendue du manteau

Nous avons entraîné des classifieurs à la prédiction par paire et des fonctions de rang bivaluées sur les jeux d'entraînement d'AMOSss.0 et d'AMOSss.1. Sur la figure 5.1 les diagrammes obtenus par épaississement (voir chapitres 3 et 4). Par souci de lisibilité, nous n'affichons que les deux meilleurs modèles (au sens d'une plus grande aire sous la courbe dans le diagramme Pso - completeness) par catégorie.
Les modèles entraînés sur AMOSss.0 (courbes oranges et bleues) sont mal séparés. Par contre, les classifieurs entraînés sur AMOSss.1 à la prédiction par paire prennent un ascendant net sur les autres modèles. Grâce à la troisième phase d'annotation, nous nous retrouvons dans la situation présentée au chapitre 3.
Par contre, les performances en généralisation des fonctions de rang entraînées sur AMOSss.1 sont légèrement moins bonnes que celles entraînées sur AMOSss.0. Cette moins-value apparaît aussi sur les diagrammes prudence-completeness et correctness-completeness.
Enfin, les fonctions de rang construites sur le canal des bleus sont dépassées de plus de 15 points de justesse (30 points de correctness) sur le problème à deux classes (figure 5.1).

5.2.1.3 Résultats pour l'épaisseur de neige

La prédiction par paire entraînée sur AMOSsd.0 présente aussi de meilleurs résultats que les fonctions de rangs bivaluées. La figure 5.2 présente les mêmes gradations que la figure 5.1.
Les valeurs atteintes en correctness et Pso sont cependant moins bonnes que pour l'étendue. Par exemple le meilleur classifieur atteint une Pso de 84 % pour une completeness de 77 % (contre 86% et 93% repectivement sur le paramètre étendue). Cela n'est pas seulement dû à une proportion plus importante de paires incomparables : sur le problème à deux classes, les correctness sont inférieures de trois points (points d'abscisse 1 sur les diagrammes correctness-completeness).
Notons aussi que pour des fréquences de présentation équivalentes, les completeness des modèles sont toujours plus faibles que pour le paramètre étendue.

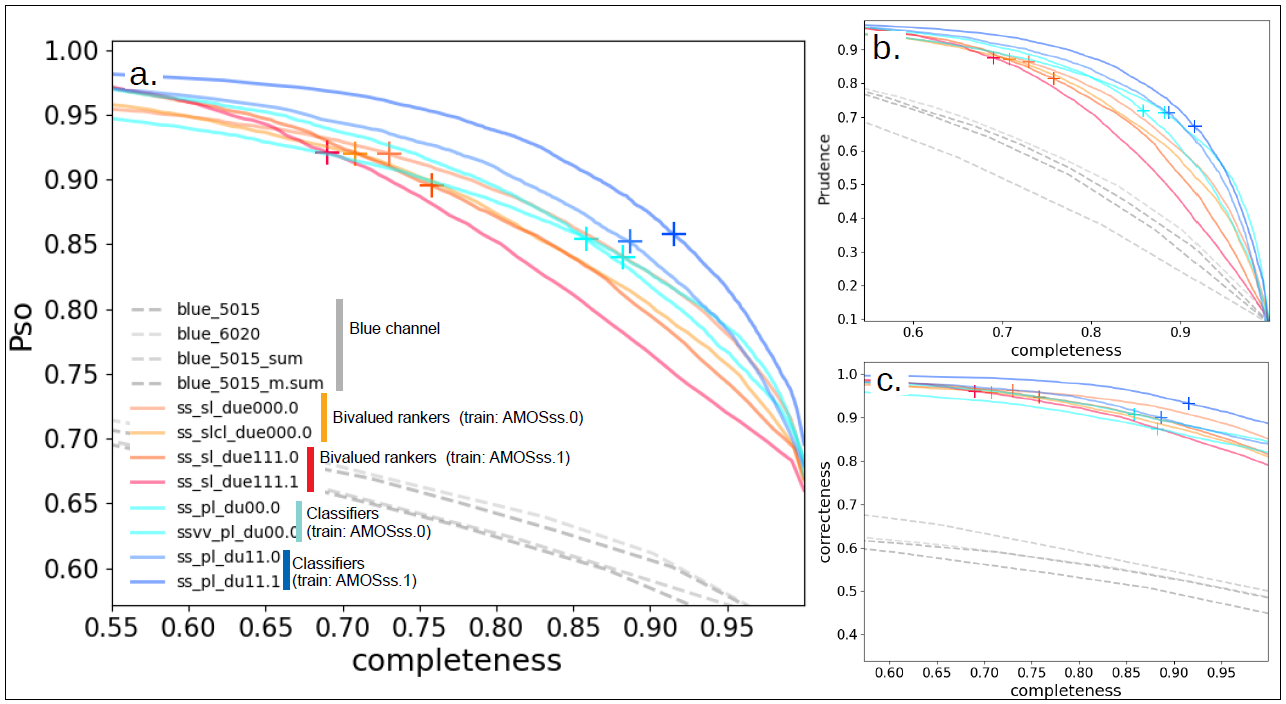


FIGURE 5.1 – Résultats des apprentissages sur les jeux relatifs au paramètre étendue du manteau. Les fonctions de rang bivaluées (prédiction par image) sont en orange et rouge, les classifieurs (prédiction par paire) en bleu. L'épaississement est obtenu selon les règles des encadrés 3.1, 3.2 et 4.1.

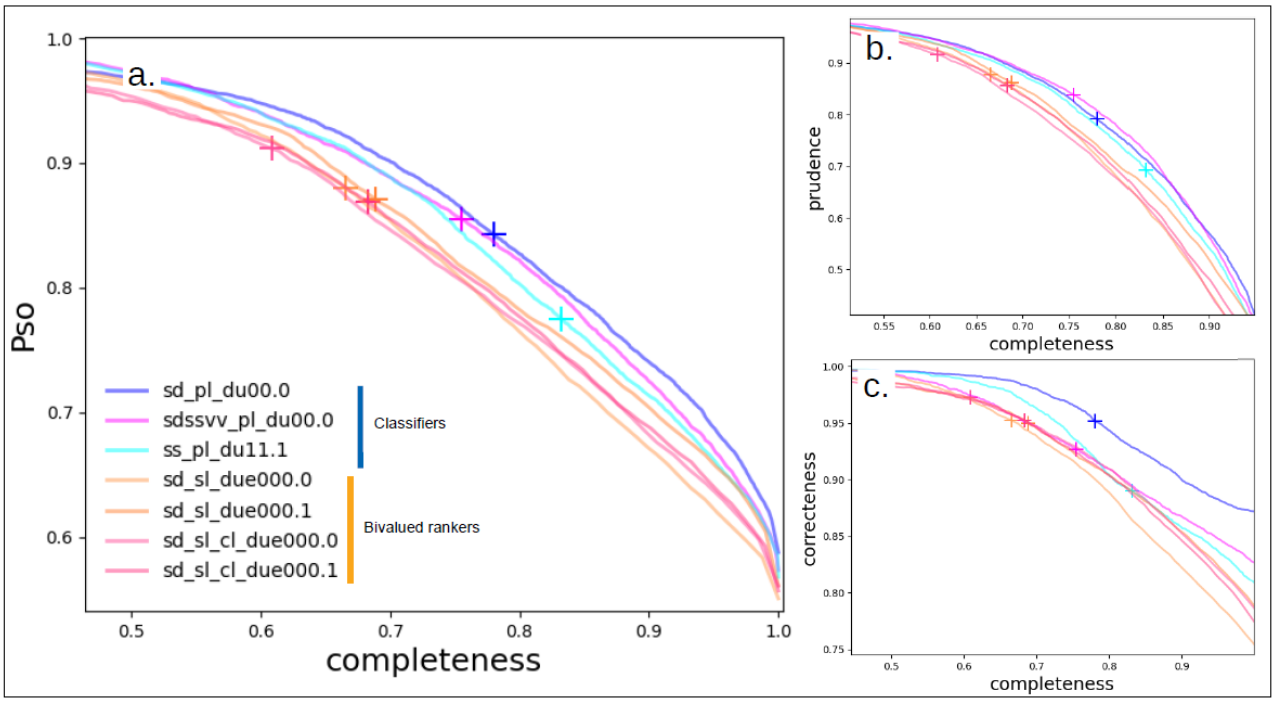


FIGURE 5.2 – Résultats des apprentissages sur les jeux relatifs au paramètre épaisseur du manteau (AMOSsd.0). Les fonctions de rang bivaluées (prédiction par image) sont en orange et rouge. Les classifieurs (prédiction par paire) sont en rose et bleu. Le classifieur ss_pl_du11.1 a aussi été entraîné sur les paires supplémentaires issues de la troisième étape d'annotation.

5.2.1.4 Séparation de l'étendue et de l'épaisseur

Nous avons cherché à savoir si les prédictions des modèles étaient bien spécifiques au paramètre ciblé. Nos modèles n'avaient peut-être pas eu besoin de faire une différence entre étendue et épaisseur -qui covarient le plus souvent- pour atteindre ces niveaux de performances.
Nous avons donc mesuré les performances de classifieurs testés sur le paramètre sur lequel ils n'ont pas été appris.
Sur la figure 5.4, on observe que les performances du meilleur classifieur entraîné sur l'épaisseur chutent lorsqu'elles sont calculées sur le jeu de test AMOSss (figure 5.4.a). C'est l'inverse pour le classifieur entraîné sur l'étendue (figure 5.4.b). De plus, c'est toujours le classifieur entraîné et testé sur le même paramètre qui donne les meilleurs résultats.
Cependant, la spécificité des prédictions, si elle renseigne sur le degré de spécialisation des modèles, ne permet pas d'évaluer la capacité à séparer les deux paramètres [3]. Pour évaluer cette capacité nous utilisons les paires « contradictoires ». L'annotation d'images non consécutives a fait apparaître, dans nos jeux de données, des paires ordonnées dans un sens différent selon qu'on considère l'étendue ou l'épaisseur (voir figure 5.3). Ces paires seront dites « contradictoires ». Dans nos jeux, elles représentent entre 2 et 5 % des paires strictement ordonnées. Dans le jeu de test, on compte ainsi 506 paires contradictoires.

FIGURE 5.3 – Exemple d'une paire contradictoire sur une caméra du jeu AMOS. Sur l'image de gauche, l'étendue est plus grande et l'épaisseur moins grande que sur l'image de droite.

Sur la figure 5.4.b, nous avons tracé le diagramme correctness-completness des meilleurs classifeurs après restriction aux paires contradictoires, orientées dans le sens des étendues. Le classifieur entraîné sur AMOSss.1 est moins performant sur ces paires, mais la correctness reste positive, de l'ordre de 0.5. Quant au classifieur entraîné sur l'épaisseur, sa correctness devrait être idéalement de -1. Dans les faits, sa correctness est presque nulle.

3. Dans le sens où les prédictions seraient indépendantes des variations du second paramètre

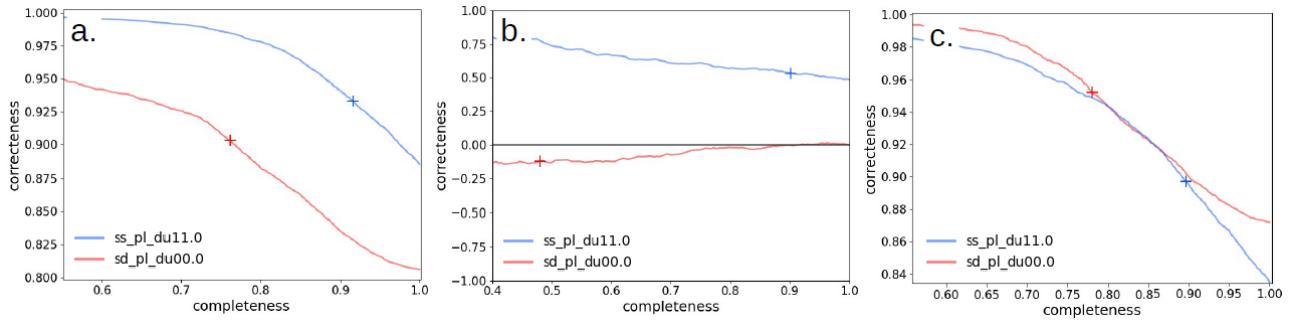


FIGURE 5.4 – a. Le meilleur classifieur sur le paramètre épaisseur est testé sur le jeu relatif à l'étendue (courbe rouge). La courbe bleue correspond au meilleur classifieur sur le paramètre étendue. b. Les mêmes courbes sont tracées après restrictions aux paires contradictoires du jeu de test AMOSss.1. Sur ces paires, le score parfait pour helvetica ss_pl_du11.0 est de +1 tandis qu'il est de -1 pour le réseau sd_pl_du00.0, entraîné à ordonner suivant l'épaissseur. Pour ce graphe, les prédictions ont été symmétrisées (voir chapitre 3). c.Les mêmes modèles sont évalués sur le jeu de test d'AMOSsd.0.

5.2.1.5 Discussion

La troisième étape d'annotation a été décidée parce que les fonctions de rang apprises sur AMOSss.0, dont les performances atteignaient celles de la prédiction par paires, présentaient de nombreux cas de fausses détections. Appliquée sur des sous-séquences d'une dizaine d'images consécutives, cette étape a permis d'augmenter rapidement le nombre de scènes annotées. Mais cette opération a surtout permis d'augmenter le nombre de paires incomparables et équivalentes.
Cet apport déséquilibré peut peut-être expliquer que les performances des fonctions de rang entraînées sur le nouveau jeu n'aient pas progressé.
Ce résultat est en tout cas compensé par le net progrès de la prédiction par paire sur le paramètre de l'étendue.
Cette amélioration nous ramène dans la situation déjà rencontrée avec le paramètre visibilité : une prédiction par paire plus performante que la prédiction par image.

La même situation est observée sur le paramètre épaisseur. Néanmoins, les scores atteints par les modèles sont globalement plus faibles que sur l'étendue. La moins bonne performance de la prédiction par paires pour l'épaisseur vient avec une incapacité à séparer l'étendue de l'épaisseur (un classifieur entraîné sur l'épaisseur n'est pas meilleur que le hasard sur les paires « contradictoires »). Cependant, il reste plus avantageux que le classifieur entraîné sur l'étendue pour l'évaluation de l'épaisseur (figure 5.4).
Cet écart de performance entre l'étendue et l'épaisseur peut s'expliquer par la rareté relative des paires strictement ordonnées dans le jeu de données. Moins bien décrit dans le jeu, le phénomène est moins bien appris.
Pour améliorer les performances sur le paramètre épaisseur, nous ne voyons pas d'autre moyen que de compléter l'annotation, en particulier en recherchant les séries temporelles d'images qui contiennent plusieurs épisodes de neige, avec de fortes fluctuations de l'épaisseur au sol. Compte tenu des échéances,

il semblait difficile de reprendre les étapes de collection d'image et d'annotation.
Au contraire, pousser les performances des fonctions de rang sur l'étendue paraissait faisable et doublement avantageux. Non à des fins d'étalonnage, ce paramètre n'étant pas une grandeur physique mesurable, mais plutôt :

— pour pouvoir signaler un début d'accumulation de neige lors de chutes de neige en plaine, et donc à répondre avant la fin du temps imparti à l'un des enjeux principaux du projet.
— pour compléter plus facilement l'annotation sur l'épaisseur, en circonscrivant les épisodes de neige sur les séquences non annotées à l'aide d'une estimation des étendues plus performante.

Pour améliorer nos fonctions de rang, nous avons repris l'approche semi-supervisée développée pour le paramètre visibilité.

5.3 Une approche semi-supervisée pour l'étendue du manteau neigeux

Dans la première section du chapitre 4, nous présentions une approche semi-supervisée basée sur les meilleures performances des classifeurs entraînés à la prédiction par paires. Cette approche consistait à utiliser les classifeurs pour étendre l'annotation sur des séquences supplémentaires. Cette extension a permis d'améliorer sensiblement la correctness de la prédiction par image.
Dans cette section, nous reprenons le même principe sur un paramètre différent : l'étendue du manteau neigeux. La procédure d'extension a été renouvelée et adaptée au paramètre d'intérêt. La section 5.3.1 présente les principaux aspects de cette procédure. Le détail est précisé dans l'annexe E.
Les fonctions de rang entraînées sur le jeu résultant, nommé AMOSssExt, sont évaluées sur le jeu de test d'AMOSss.1 (section 5.3.2). Les résultats sont discutés dans la dernière sous-section.

5.3.1 Construction d'AMOSssExt

Comme pour AMOSvvExt, la construction d'AMOSssExt est basée sur une sélection des paires annotées par le meilleur classifieur sur le jeu validation d'AMOSss.1 (ss_pl_du11.0). Dans une première étape, les prédictions du classifieur ont de nouveau été symétrisées et converties en un ordre d'intervalle sur les images des séquences AMOS non annotées. Cette fois, cependant, l'ordre d'intervalle a été obtenu par optimisation (algorithme varIO, Annexe D-IV D.1).

Dans une seconde étape, cet ordre est corrigé, en tenant compte des particularités physiques du paramètre. D'abord, les variations de l'étendue du manteau sont plus lentes que celles de la visibilité. Même pendant un épisode de neige intense, il est rare que la tendance s'inverse sur trois images consécutives (augmentation puis baisse ou l'inverse). Cela permet de détecter une partie des erreurs.
De plus, l'étendue ne peut jamais augmenter en dehors des épisodes de » basse visibilité » associées aux chutes de neige [4]. Les paires associées aux prédictions erronées sont stockées parmi les incomparabilité (graphe $\mathcal{U}_2^s$) pour être présentées lors des apprentissages.
Nous avons contrôlé les séries temporelles d'intervalles prédits à l'issue de ces deux étapes. La cohérence temporelle était bonne, mais le chevauchement était généralement trop marqué. Nous avons tenté de compenser l'effet de ces chevauchement par l'ajout d'environ 200.000 comparaisons strictes, moins sûres mais plus fines (voir annexe E).

Les détails des opérations de construction des séquences [5], de correction des prédictions, de sélection des comparaisons fines et de construction des nouveaux graphes d'apprentissage sont présentés dans la première section de l'annexe H. A l'issue de la procédure, le jeu AMOSssExt comprenait dix

4. Cette observation a d'ailleurs déjà été exploitée pour la phase automatique de la troisième étape d'annotation
5. Les séquences d'image utilisées ne sont pas exactement celles du jeu AMOSvvExt. Ces dernières ont été refondues après la troisième étape d'annotation (voir Annexe H-I).

jeu	séquences	images	comparaisons strictes		incomparabilités		équivalences	
			graphe	arêtes	graphe	arêtes	graphe	arêtes
AMOSss.1	1.331	39.769	$\mathcal{G}_1^s$	255.838	$\mathcal{U}_1^s$	56.756	$\mathcal{E}_1^s$	57.702
AMOSssExt	>3.000	579.664	$\mathcal{G}_2^s$	1.282.294	$\mathcal{U}_2^v$	892,249	$\mathcal{E}_2^v$	109.824

TABLE 5.3 – Effectifs des jeux AMOSss.1 et AMOSssEXT.

fois plus d'images qu'AMOSss.1 (table 5.2).

5.3.2 Résultats

5.3.2.1 Modèles entraînés sur AMOSssExt

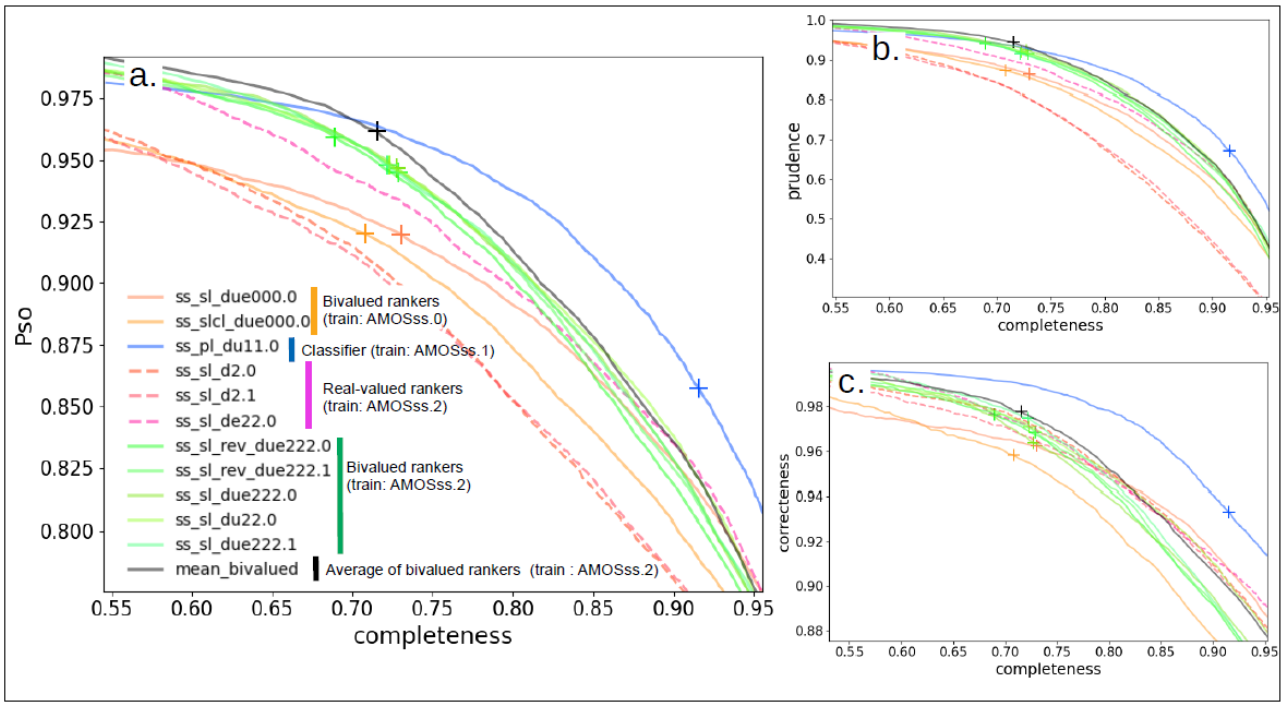


FIGURE 5.5 – Performances des fonctions de rang semi-supervisées pour l'estimation relative de l'étendue. Les Pso, prudence, completeness et correctness ont été calculées sur les paires du jeu de test d'AMOSss.1. La courbe bleue marque les performances du classifieur utilisé pour construire le jeu AMOSssExt. Les courbes vertes représentent les fonctions de rang bivaluées apprises sur AMOSssExt. Le moyennage de ces cinq fonctions donne la courbe noire. Les courbes pointillées correspondent à des fonctions de rang à valeurs réelles apprises sur AMOSssExt tandis que les courbes oranges continues représentent les meilleures fonctions de rang bivaluées apprises sur AMOSss.0.

Pour évaluer l'intérêt du nouveau jeu, plusieurs fonctions de rang ont été entraînées. Quatre fonctions de rang bivaluées (courbes vertes sur les figures 5.5-5.6) ont été entraînées sans changement notable [6] par rapport aux fonctions de rang précédentes. Une cinquième a été entraînée sans les paires

6. Les changements concernent la pondération des arêtes dans les graphes pour la formation des mini-lots (voir chapitre 3) pour ss_sl_due222.1 et le sens dans lequel les comparaisons sont apprises pour ss_sl_rev_due222.0 et

équivalentes pour des performances très proches.
Trois autres fonctions de rang à valeurs réelles ont été apprises sur AMOSssExt, dont deux sur le seul graphe $\mathcal{G}_2^s$ (ss_sl_d2.0 etss_sl_d2.1). La troisième (ss_sl_de22.0) a été entraînée sur les graphes $\mathcal{G}_2^s$ et $\mathcal{E}_2^s$ avec des fréquences de présentation de $p_g = 60\%$ et $p_e = 40\%$. Ces fonctions de rang sont représentées par les courbes en pointillé sur les figures 5.5-5.6.

Pour des niveaux de Completeness autour de 0.7, les cinq fonctions de rang bivaluées (croix vertes sur la figure 5.5) présentent de meilleurs compromis Pso-Completeness que les fonctions de rang apprises sur AMOSss.0. La moyenne des cinq fonctions de rang bivaluées, notée mean_bivalued (courbe noire), permet de gagner quelques points sur tous les diagrammes.
Sur le problème à deux classes, c'est encore une fonction rang de entraînée sur AMOSvvExt (ss_sl_de22.0) qui permet d'obtenir les meilleurs résultats mais ce n'est pas une fonction de rang bivaluée.

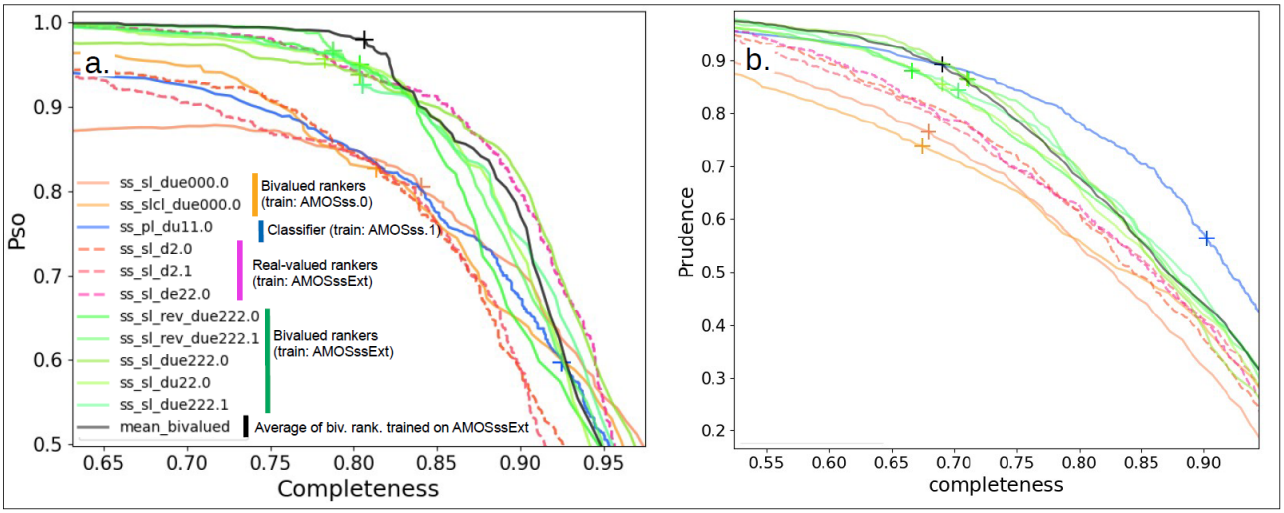


FIGURE 5.6 – Détail des performances. Les modèles comparés sont les mêmes que pour la figure 5.5. a.Diagramme Pso-Correctness après restriction aux paires contenant au moins une image sans neige au sol. b.Diagramme prudence-correctness après restriction aux paires comprenant au moins une image de mauvaise qualité.

Lorsque l'on restreint le calcul des performances aux paires d'image contenant au moins une image sans neige au sol, les écarts entre les fonctions de rang entraînées sur AMOSssExt et sur AMOSss.1 se creusent (figure 5.6.a).
Pour les niveaux de completeness atteints (autour de 0.8), la prédiction atteint la perfection ($P_{so} > 99\%$). Les fonctions de rang bivaluées dépassent le classifieur utilisé pour construire AMOSssExt.
Sur ces paires, on peut aussi noter que la fonction de rang entraînée sur $\mathcal{G}_2^s$ et $\mathcal{E}_2^s$ rejoint le groupe des fonctions bivaluées.

ss_sl_rev_due222.1. Dans le deuxième cas, les modèles sont appris sur les graphes transposés et les fonctions de rang résultantes sont multipliées par -1. Cette opération est sensée aider à corriger l'asymétrie des fautes de prudence (chapitre 4). Elle n'a pas eu d'effet apparent sur les résultats et ne sera pas détaillée dans ce manuscrit.

Si comme au chapitre au précédent, on s'intéresse à la plus-value des fonctions de rang bivaluées, la restriction aux paires comprenant au moins une image de mauvaise qualité est éclairante. Sans changement notable sur les diagrammes correctness-completeness, l'écart de prudence se creuse entre le groupe des fonctions de rang bivaluées et celui des fonctions de rang à valeurs réelles entraînées sur AMOSssEXT : sur la figure 5.6.b on observe un écart de dix points entre la moyenne des fonctions bivaluées et les fonctions d'ordre à valeurs réelles (courbes pointillées), contre moins de cinq points sur la figure 5.5.b. Comme pour la visibilité, la taille de l'intervalle prédit capte les principales causes d'incomparabilité comme les reflets sur la chaussée, les contrejours, les défauts de focalisation, et les masques météorologiques (gouttelettes, flocons), même si cette prise en compte est encore imparfaite (cas de non détection observés, voir section 5.6).

5.3.3 Discussion

Les conséquences d'un apprentissage sur ce nouveau jeu sont positives, mais assez différentes de celles observées sur la visibilité : à même Completeness, ce ne sont pas les Correctness qui ont été améliorées mais la Prudence.
En particulier, les nouvelles fonctions de rang font bien moins de fautes de détection sur les images sans neige. C'est donc l'objectif de la troisième étape d'annotation qui est atteint à travers cette approche semi-supervisée (dans le cas des fonctions de rang).
Quel est est le poids des meilleures performances du classifieur dans ce résultat ? Comme au chapitre précédent, il est difficile de conclure sans mener une analyse complète, par « ablation », de chacune des étapes de construction de AMOSssExt. Notons tout de même que sur les paires contenant au moins une image sans neige au sol, le classifieur (non symétrisé) n'était pas meilleur que les fonctions de rang apprises sur AMOSss.0 (figure 5.6). Nous pensons donc que les étapes de correction des prédictions par paire ont dû jouer un rôle important.
Par contre, en terme de compromis Correctness-Completeness, les fonctions de rang bivaluées se montrent légèrement moins bonnes que les fonctions de rang à valeur réelle. Cet écart était encore plus marqué avant que nous n'ajoutions les comparaisons fines au jeu AMOSssExt. Cela appelle une nouvelle évolution dans la méthode de construction du jeu étendu. Ce thème est discuté plus en détail dans l'annexe E, à la fin de la section consacrée à AMOSssExt.

Les progrès en matière de fausse détection sur des images non-enneigées nous ont amené à revenir sur le problème neige-non neige avec nos fonctions de rang. Cette nouvelle étude, conduite sur les séquences de la base TENEBRE_1218, est présentée dans la section suivante.

5.4 Classification neige/non neige sur les séquences TENEBRE

Dans cette section nous reprenons le problème de la classification neige - non neige à partir des fonctions de rang bivaluées. Cette analyse est réalisée sur les séquences de base TENEBRE_1218, pour lesquelles nous disposons de mesures d'enneigement fiables.

Un retour sur le problème à deux classes nous permet de confronter les fonctions de rang bivaluées aux modèles entraînés sur la tâche de classification binaire "neige au sol/non neige au sol" (section 3.1.1). A ce titre, l'utilisation des caméras TENEBRE est justifiée. Les améliorations validées sur les jeux de test issus d'AMOS faisaient perdre à ce jeu sa neutralité.
C'était aussi l'occasion de confronter nos prédictions à une donnée instrumentale complètement objective. Enfin, cela donnait la possibilité de produire des résultats comparables à ceux de la littérature, principalement tournée vers ce problème.

Cette courte section est organisée en deux parties. Dans la première, nous décrivons les classifieurs utilisés. Dans la seconde, les résultats sont présentés et discutés.

5.4.1 Modèles et données utilisées

Les modèles de la section 3.1.3 n'avaient été entraînés que sur une sous-partie des images constituant les bases d'entraînement AMOSss.0 et AMOSsd.0. Pour une comparaison plus juste, nous avons ré-entraîné des ResNet50 à la classification sur des jeux complets, en suivant la procédure décrite en section 3.1.2.
Le modèle noté ss_cl_d.0 est entraîné sur le problème à deux classes défini sur les images de jour d'AMOSss.0, à partir des labels produits lors de la première étape d'annotation.
Le modèle noté ss_cl_d.1 est entraîné sur le problème à cinq classes (table 3.1) défini sur les images de jour d'AMOSss.0. Le modèle est ramené à une prédiction à deux classes par sommation après la couche softmax, comme défini au chapitre 3.
Enfin, le modèle noté ss_cl_dn.0 est entraîné sur le problème à cinq classes sur toutes les images de jour et de nuit pour lesquelles des labels sont disponibles (soit 15.727 images -voir la table 1 annexe A-II). Le modèle est ramené à une prédiction à deux classes comme précédemment.
Les prédictions peuvent être modulées en considérant un seuil de dépassement sur le canal de sortie associé à la classe « neige ». Pour un seuil t_c égal à un, la neige n'est jamais prédite (sensibilité nulle). Pour un seuil $t_c = 0$, c'est l'inverse (spécificité nulle). Pour $t_c = 0.5$, on retrouve la prédiction nominale du modèle.

Ces modèles sont comparées aux fonctions de rang bivaluées entraînées sur AMOSssExt, à leur moyennage (mean_bivalued) et à la meilleure fonction de rang entraînée sur AMOSss.0 (ss_sl_due000.0)

Pour convertir un intervalle en détection, nous choisissons un indice t_r entre zéro et un, puis nous appliquons la règle de décision suivante :
Soit $z^+_{t_r}$ le quantile d'ordre t_r de la série des $z^+_i = f(x_i, w)^+$ obtenue en premant toutes les images x_i disponibles pendant le premier hiver. Pour une image x_j, on décide la classe neige si $f(x_j, w)^- > z^+_{t_r}$.

Les prédictions sont évaluées sur les neufs scènes pour lesquelles nous disposons de mesures fiables entre 8h et 18h et sur les deux hivers disponibles (oct. 2012 - mars. 2013 et oct. 2017 - mars 2018). La vérité terrain est fournie par la mesure locale de l'épaisseur. Chaque valeur strictement positive est comptée comme une occurrence. Les discordances entre mesure et apparence ne sont pas prises en compte [7].

5.5 Résultats et discussion

Les diagrammes sensibilité-spécificité de la figure 5.7 ont été obtenus en faisant varier les différents seuils t_c et t_r. Les courbes en pointillé représentent les compromis atteints par les trois classifieurs, les courbes en trait plein représentent les fonctions de rang bivaluées. En dehors de la caméra neige_nancy, ces dernières dépassent largement les premiers.
La fonction de rang bivaluée apprise sur AMOSss.0 est aussi largement devancée par les modèles entraînés sur AMOSvvExt sur les cinq dernières scènes.
Les croix noires représentent le compromis atteint par le modèle moyenné pour $t_r = 0.5$. Les jutesses (acc.) sensibilité (sens.) et spécificité (spec.) qui leurs sont associées sont affichées au centre des graphiques.

Dans [35], Kosmala et al. pointent la difficulté des CNN à voir détecter la neige sur des scènes qui n'ont pas été apprises. Sur les scènes TENEBRE, nous observons la même difficulté pour des classifeurs entraînés sur une tâche de classification, mais les fonctions de rang apprises sur AMOSssExt généralisent bien mieux. Nous obtenons sur ces neuf scènes une spécificité globale de 98 % et une sensibilité de 89 %.
Le nombre de scène étant limité l'annotation étant obtenue avec un instrument, la comparaison directe avec les scores affichés par Kosmala et al. [35] est encore limitée. Mais au moins peut-on rappeler les scores présentés par ces auteurs dans le cas où les scènes testées sont différentes des scènes vues à l'entraînement, soit une sensibilité globale de 83.3 % et une spécificité globale de 94 %.

En dehors des questions de comparaison, ce résultat nous dit que, hors épisode de neige, les fausses détections sont très rares tandis que de l'ordre de 90 % des images avec neige sont signalées -sans compter les causes légitimes de non détection [8] (obscurité, mauvaise

7. Pour rappel, nous savons qu'elles peuvent compter pour 4 % à 8 % des images (chapitre 2) et qu'elles correspondent majoritairement à des cas de neige apparente non détéctée par l'instrument
8. Voir note 7.

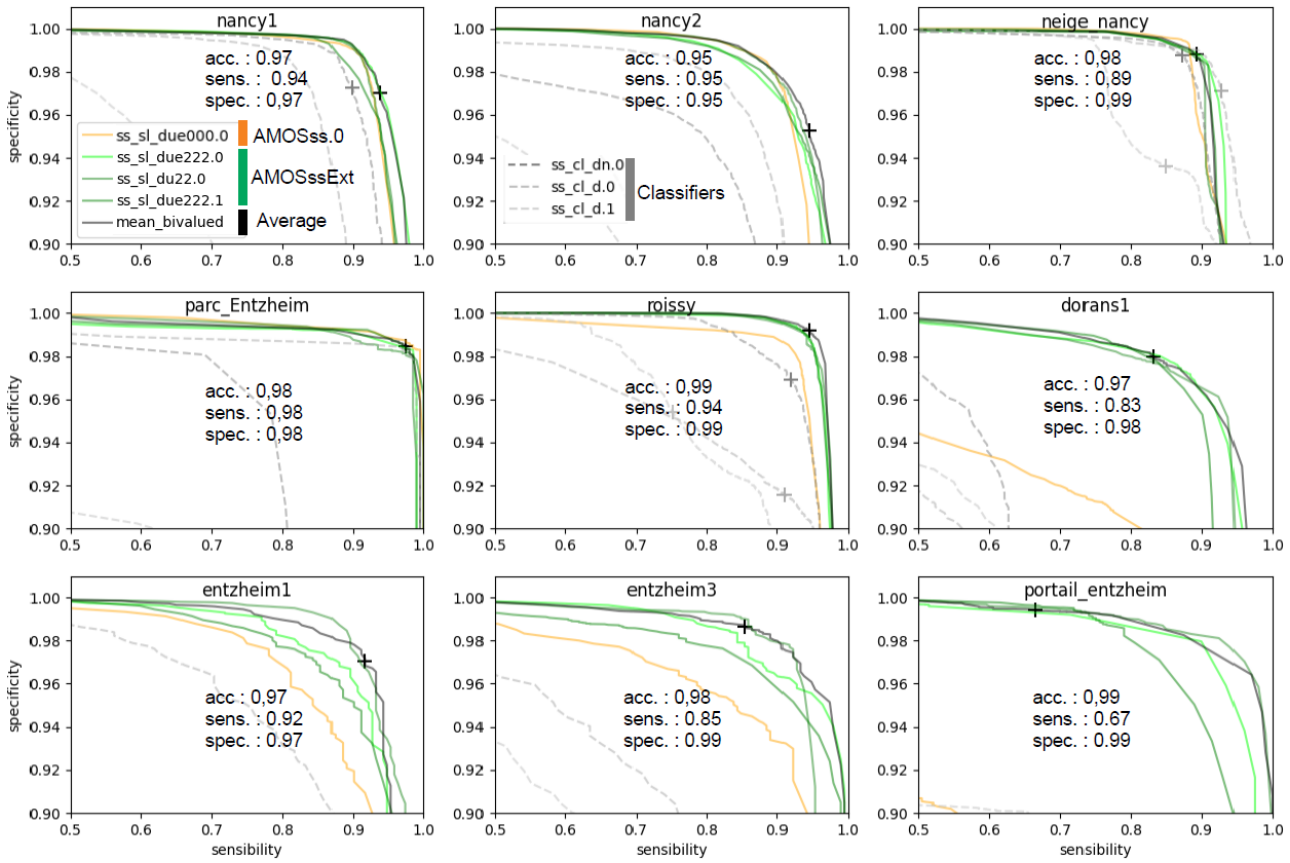


FIGURE 5.7 – Comparaison des classifieurs et des fonctions de rang bivaluées sur la classification « neige au sol/non neige au sol ». La vérité terrain est définie à partir des labels instrumentaux de la base TENEBRE_1218. Les 9 diagrammes Spécificités-Sensibilité correspondent aux caméras TENEBRE (figure 2.4)

qualité, non représentativité de la mesure). Mais est-ce suffisant pour passer à une application pratique ? Pour le savoir, il faut revenir au cadre qui nous intéresse réellement, et se concentrer sur les périodes où la neige tombe et tient au sol. C'est pendant ces courtes périodes que les fausses détections et les non-détections comptent vraiment. Ces périodes concentrent-elles les fausses détections ? Quels sont les délais avant une prédiction sûre ? Dans la section suivante, nous abordons ces questions à travers une étude de cas.

5.6 Suivi d'épisodes de neige en plaine pendant l'hiver 2020-2021

Dans cette section, nous illustrons le potentiel de nos méthodes sur des cas de neige survenus pendant l'hiver 2020-2021. Ce ne sont certes pas des événements marquants. Les hauteurs de neige sont restées modestes, de l'ordre de 5 cm à 10 cm, et ne se sont pas écartées des prévisions. Mais ici, nous nous intéressons moins à la capacité des réseaux à signaler un événement exceptionnel qu'au délai entre un début de tenue au sol et la première prédiction.
Cette question du délai est importante parce qu'elle touche à la rapidité de la prise de décision (fermeture de route, envoi d'une déneigeuse, etc). Nous l'abordons ici sans post-traitement, et en particulier, sans lissage : il s'agit avant tout d'évaluer la prédiction brute, par image. Dans la première partie, nous décrivons les séquences et les événements sur lesquels l'étude est conduite, les modèles utilisés et la méthode de conversion des prédictions -de nature ordinale- en détection d'événement. Les résultats sont décrits et discutés dans la seconde partie.

5.6.1 Données et méthode

L'étude a été conduite sur des webcams maintenues par les DIRs ou disponibles sur le site infoclimat.com. Les caméras utilisées pour l'apprentissage (entraînement et validation) ont été écartées. Des archives ont été constituées pendant les périodes de neige, sur les mois de décembre et janvier 2021. Le pas de temps est compris entre cinq et dix minutes. Sur ces deux mois, nous avons retenu une dizaine de jours de neige.
Pour chacun des jours de neige retenus, nous avons cherché les séquences d'images sur lesquelles l'étendue du manteau neigeux croit au moins une fois entre 8h UTC et 17h UTC. Nous avons ainsi sélectionné 68 séquences issues de 60 webcams différentes.

Sur chacune de ces séquence, nous considérons que la première image à partir de laquelle le manteau s'est étendu donne l'heure de début de l'évènement (début « observé »). Pour choisir cette image, nous avons souvent dû parcourir la séquence dans un sens et dans l'autre. La vérité terrain est donc déduite de l'image courante et du contexte [9].

Le début « prédit » est obtenu à partir des fonctions de rang bivaluées. Nous avons utilisé le modèle ss_sl_due000.0, entraîné sur AMOSss.0 et le modèle moyen mean_bivalued obtenu à partir des cinq fonctions de rang bivaluées entraînées sur AMOSssExt. Ces deux modèles sont appliqués à toutes les images webcam disponibles sur la période du 01/12/2020 au 31/01/2021.
Pour passer des intervalles prédits à la détection d'événements, nous reprenons la méthode de détection par dépassement de seuil définie dans la section précédente. La seule diffé-

9. De son côté, le réseau n'a accès qu'à l'image courante, ce qui le désavantage légèrement.

rence réside dans la détermination du seuil z_t^+. Pour fixer ce seuil, nous devons tenir compte de l'état du sol avant le début des chutes de neige. Lorsque le sol est sans neige, ce seuil est fixé à z_{1000}^+, c'est à dire la millième valeur[10] de la série complète des z_i^+ rangée dans l'ordre croissant. Lorsque la neige est déjà présente, nous ne considérons que les z_i^+ prédits pendant la journée et le seuil est fixé à la dixième valeur la plus grande.
Le début prédit est défini par le timestamp du premier dépassement de seuil.

5.6.2 Résultats et discussion

La figure 5.8 donne un aperçu représentatif des délais de prédictions au cours d'un événement de neige en plaine à enjeu (vigilance jaune/orange sur une partie des départements du bassin parisien). Pour la construire, nous avons considéré les 24 séquences archivées le 16/01/2021 au cours d'un événement de neige en plaine. Lors de cet événement, un front chaud circule d'est en ouest et donne des précipitations neigeuses sur le nord de la France. Les hauteurs de neige, nulles au début de l'événement, vont jusqu'à 5 cm en station sur les départements couverts par les webcams disponibles (figure 5.8.b).

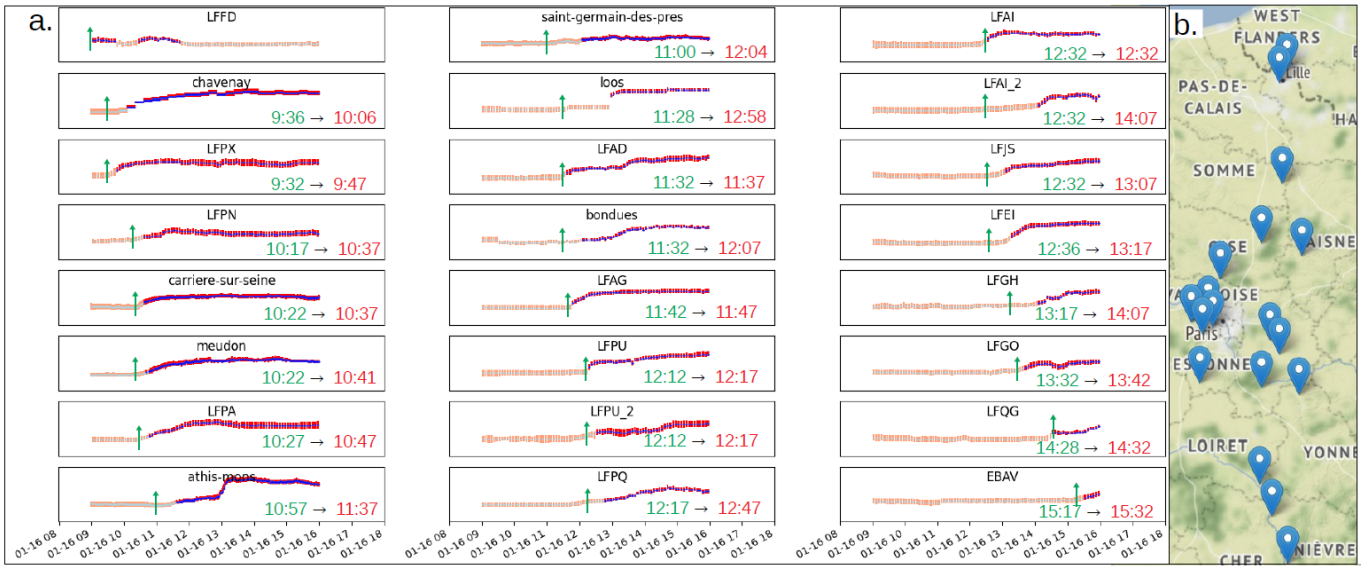


FIGURE 5.8 – a.Prédictions de mean_bivalued sur les séquences d'images du 16/01/2021. Les intervalles prédits ($[z_i^-, z_i^+]$) sont représentés en rouge foncé lorsque le seuil est dépassé ($z_i^- > z_{1000}^+$). La première image de la journée sur laquelle le seuil est dépassé donne le début « prédit » de l'événement. Les flèches vertes représentent l'heure UTC à laquelle la neige commence à apparaître sur le sol (début observé). Les heures des débuts observés (resp. prédits) sont affichées en vert (resp. en rouge). Le cas particulier de la webcam LFFD est décrit dans le texte. b.Localisation des webcams (sauf l'aérodrome EBAV, située en Belgique).

Sur ces 24 séquences, les débuts prédits à partir du modèle mean_bivalued sont toujours postérieurs aux débuts observés[11] (voir figure 5.8). Les prédictions sont cohérentes : la sé-

10. Ce paramètre n'a pas été optimisé. Les délais de détections sont relativement peu sensible au rang choisi.
11. La première séquence (LFFD) est particulière. de la neige était déjà présente sur l'image au début de la séquence. Les chutes de neige ont repris entre dix et onze heures mais la neige n'a tenu qu'un court moment.

quence est coupée en deux parties homogènes.
Sur ces séquences, les retards vont jusqu'à 1h30 pour la scène d'aérodrome LFAI_2. Mais sur les autres scènes, le retard est généralement inférieur à la demi-heure.

	Délai (min.)	Délai déc.	Délai janv.	ND	FD
ss_sl_due000.0	18.4	14.5	22	11	31
mean_bivalued	17.9	10	25	5	20

TABLE 5.4 – Délais moyens (sur tous les épisodes, lors des épisodes de janvier, lors des épisodes de décembre), non-détection (ND), et fausses détections (FD) rencontrées sur les 68 séquences.

Sur l'ensemble des 68 séquences disponibles, ce délai est de 18 minutes en moyenne pour les deux modèles. Certains événènements ne sont pas détectés. Cela se produit dans les cas où la neige couvre déjà toutes les surfaces naturelles de la scène et ne colonise qu'une petite portion de la voirie. Pour notre meilleur modèle (mean_bivalued), seuls 5 des 68 événements échappent à la détection.
Nous avons aussi comptabilisé les fausses détections. Précisément, non comptons toutes les fois où le modèle prédit une croissance ou une fonte de manière erronée. Dans la majorité des cas, les raisons sont identifiables : il s'agit du passage d'un véhicule blanc, d'un changement brutal dans les conditions d'éclairage, d'images voilées, de flocons ou de gouttelettes sur la lentille, d'incrustation de texte, etc. Ces fausses détections sont sensiblement moins nombreuses avec le modèle helvetica mean_bivalued. Seules 12 fausses détections, concernent plus de deux images consécutives. Elles sont presque toutes associées à des hydrométéores sur la lentille. Certaines de ces fausses détections peuvent être filtrées à partir de la taille relative des l'intervalles (voir chapitre 4, encadré 4.2). Mais quatre d'entre elles n'ont pas été associées à un intervalle plus large, et causent des fausses détections sur plusieurs dizaines de minutes difficiles à corriger par filtrage.

5.6.3 Discussion

Sur les séquences qui couvrent les phénomènes d'intérêt, les tendances du manteau neigeux apparaissent clairement et les fausses détections sont rares. 65 des 68 événements sélectionnés sont détectés avec un retard moyen de moins de 20 minutes sur la prédiction humaine (soit deux à quatre images, selon la fréquence de prise de vue).

Sur les webcams routières des DIRs échantillonnées pendant le mois de janvier, le délai moyen de détection est inférieur (14.5 min.) à celui sur les scènes d'infoclimat (22 min.) ; cette différence tient probablement au caractère atypique des scènes d'infoclimat. C'est d'ailleurs sur les scènes les plus éloignées d'une scène routière standard que les chevauchements sont les plus importants. Par exemple, sur la scène de saint-germain-des-prés, qui donne sur les toitures parisiennes, le modèle moyenné ne permet pas de distinguer plus de deux

classes d'enneigement.
Si l'on considère le grand nombre de caméras disponibles [12], la perspective d'une détection fiable de la tenue et de la progression de la neige au sol à 15 minutes d'écart offre des perspectives intéressantes en matière d'aide à la surveillance.

Certaines causes d'incertitude restent encore mal prises en compte par les modèles. En particulier, nous avons relevé des fautes de prudence dans des cas où la vue est complètement masquée (fausses détections comptées dans la table 5.4). Pour améliorer cette situation, nous avons tenté d'introduire des images de ce type au cours de l'apprentissage. Cette tentative, encore en cours au moment de la rédaction du manuscrit, n'a pas encore permis d'améliorer les prédictions.

Sur certaines scènes, enfin, la croissance de l'étendue présente plusieurs phases, comme sur la scène d'Athis-Mons pendant l'épisode du 16/01 (bas de la première colonne figure 5.8). Nous avons vérifié que la phase de croissance brutale correspond à un passage entre deux niveaux d'enneigement différents : un enneigement limité aux surfaces naturelles et un enneigement complet.
Ces transitions rapides entre deux niveaux d'enneigements successifs sont souvent visibles sur les histogrammes des prédictions. Ces derniers sont alors bimodaux et le deuxième mode correspond à un enneigement complet des surfaces naturelles. Cette remarque pourrait être exploitée pour appliquer les fonctions de rang à un problème de classification de l'état du sol plus complet. Faute de temps, nous n'avons pas creusé dans cette direction.

12. par exemple de l'ordre de 500 caméras pour la seule Direction des Routes d'Ile-de-France en 2018

5.7 Conclusion et perspectives

Dans ce chapitre, les méthodes décrites et développées aux chapitres précédents ont été appliquées aux cas de l'étendue et de l'épaisseur du manteau neigeux.
Nous avons commencé par entraîner des classifieurs à la prédiction par paires et des fonctions de rang à la prédiction par image sur les deux jeux de paires d'images annotées disponibles, AMOSss.1 (étendue) et AMOSsd.0 (épaisseur). Sur les webcams indépendantes des jeux de test, la prédiction par paire s'est montrée plus performante que les fonctions de rang implémentées sur des réseaux pré-entraînés, comme dans le cas de la visibilité. Les performances des classifieurs sur l'étendue se sont aussi montrées supérieures à plusieurs égards. Plutôt que de relancer une nouvelle étape d'annotation pour l'épaisseur, nous avons préféré améliorer les performances des fonctions de rang sur l'étendue.
Cette amélioration a été de nouveau cherchée à travers une approche semi-supervisée. Une méthode analogue à celle du chapitre 4 a été mise en oeuvre pour construire un jeu étendu, AMOSssExt. Apprises sur ce jeu, les fonctions de rang bivaluées sont sensiblement plus prudentes sur des caméras indépendantes.
Afin de tirer un bilan sur les progrès réalisés depuis les premiers essais en classification (début du chapitre 3), nos fonctions de rang bivaluées ont été ramenées par un mécanisme de seuillage à la détection de neige au sol. Ainsi transformées, elles présentent de bien meilleurs résultats que des réseaux entraînés directement sur une tâche de classification sur AMOSss.0.
Cette étude confirme l'avantage des fonctions de rang entraînées sur AMOSssExt en matière de généralisation. De plus, les scores obtenus sur deux hivers consécutifs sont, au minimum, à la hauteur de l'état de l'art sur le problème de la classification neige au sol/non neige au sol.
Enfin, une dernière étude a été réalisée sur 68 courtes séquences d'images prises pendant des chutes de neige par des caméras indépendantes des jeux d'apprentissage. Les tendances de l'étendue sont bien suivies et les fausses détections sont très rares. Converties en détecteurs d'événement, les fonctions de rang bivaluées signalent un début de croissance avec un retard moyen inférieur à vingt minutes. Ce délai, relativement court, combiné à un taux de fausse alarme très faible, offre des perspectives intéressantes pour des applications au suivi des épisodes de neige dans un contexte routier.

Perspectives pour la prédiction de l'étendue

Le dernière section montre qu'on peut tirer d'une fonction de rang bivaluée le moyen d'estimer avec un délai raisonnable le début d'un épisode d'accumulation de la neige au sol. Mais peut-on accéder à la classification plus fine que nous tentions d'apprendre au début du chapitre 3 (problème complet) ?

Nous évoquions à la fin de la dernière section la possibilité de déduire de l'histogramme des informations sur la classe d'enneigement du problème complet. Cependant, ces informations ne pourront pas suffire à préciser si la bande de roulement est atteinte ou si la voie est entièrement couverte par la neige avec une justesse comparable à celle d'un observateur humain. Pour avancer sur ces questions, une approche par segmentation sémantique nous a semblé intéressante. L'idée est de croiser une segmentation sémantique de la scène relative aux objets de la scène (route, toitures, trottoirs, etc), obtenue avec un modèle accessoire, avec la prédiction d'un enneigement par pixel. Apprendre à prédire si le pixel couvre une surface enneigée ou non sans effort d'annotation supplémentaire nous semble possible. Pour y parvenir, plusieurs approches ont été tentées ; l'une d'entre elles s'est montrée relativement prometteuse. Nous la décrivons brièvement dans le paragraphe qui suit.

L'idée principale est de tirer la frontière entre la zone enneigée et la zone sans neige d'un modèle de traduction d'image [13]. Dans un premier temps, on applique les centres d'intervalles d'une fonction de rang sur chacune des séquences d'aMOSssExt. Pour chaque paire intra-scène, (x_i, x_j) on dispose ainsi de prédictions (z_i, z_j). A l'entraînement, un réseau U-Net (voir chapitre 1) représenté par la fonction $f(.; w)$ prend en entrée des paires (x_i, z_j) et cible les quantités (x_j, z_i).
L'idée était d'utiliser la quantité $\left.\frac{\partial f(x_i, z; w)}{\partial z}\right|_{z=z_i}$ pour repérer la zone frontière entre neige et non neige. Une preuve de concept a été faite sur des images de synthèse. Mais sur des images réelles, la méthode n'a pas encore abouti.
Par contre, les apprentissages ont convergé vers un modèle de traduction d'image intéressant car progressif. Il suffit en effet, après apprentissage, de changer la valeur d'entrée du paramètre z pour « enneiger » une image (voir figure 5.9.a). Le même principe permet d'entraîner un modèle à modifier progressivement la visibilité apparente dans l'image (voir figure 5.9.b).

Encouragés par ces observations, nous avons tenté d'améliorer les scores de nos fonctions de rang à l'aide de cette tâche accessoire de génération d'image, suivant l'exemple de [121]. Nous n'y sommes pas encore parvenus.
En dehors d'applications directes aux problématiques de la thèse, nous estimons que cette approche originale peut trouver des applications dans d'autres domaines de la vision par ordinateur.

Perspectives pour la prédiction de l'épaisseur

Au sujet d'une estimation quantitative de l'épaisseur du manteau neigeux, nous pensons qu'il est nécessaire de mieux séparer l'épaisseur de l'étendue avant de reprendre le travail.

13. Image-to-image translation

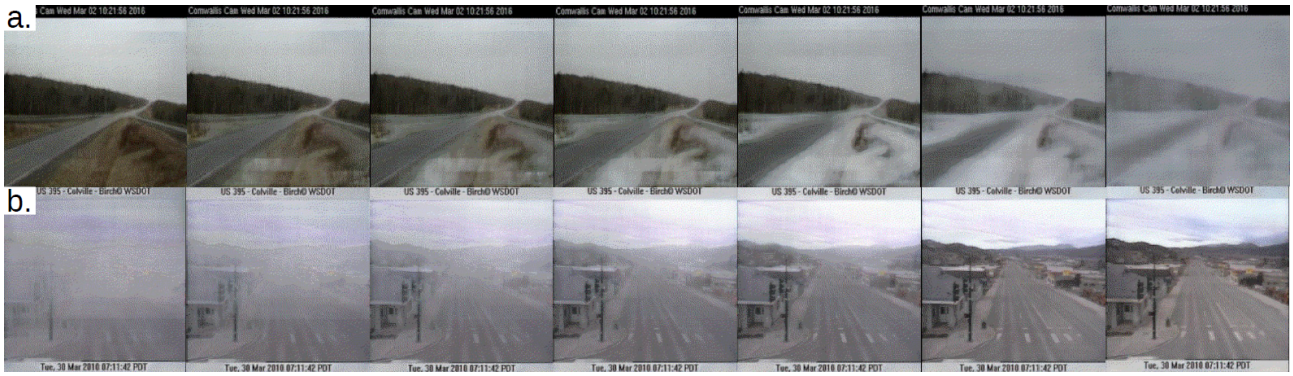


FIGURE 5.9 – Enneigement (ligne a.) et augmentation progressive de la visibilité (ligne b.) par « traduction d'image ». avec un UNet. Les deux images prises (une pour chaque ligne) en entrée appartiennent au jeu de test. La ligne a. a été générée à partir d'une image sans neige. La ligne b., à partir d'une image prise par beau temps.

Cet objectif implique sans doute une nouvelle campagne d'annotation, plus délicate à mettre en place parce que nécessitant des épisodes de fort enneigement, plus rare. A cet égard, le suivi des variations de la surface peut s'avérer un excellent outil pour rassembler des images pertinentes et compléter nos jeux d'entraînement, avant de reprendre un apprentissage par paires.

Mais anticipons et supposons qu'on ait à disposition un meilleur classifieur, et qu'il soit possible d'améliorer les fonctions de rang par une approche semi-supervisée analogue à celles mises en oeuvre pour la visibilité et l'étendue. Il resterait à étalonner les fonctions de rang pour proposer des encadrements.
Dans le chapitre précédent, nous avons proposé deux méthodes. Or aucune de ces méthodes ne semble pouvoir s'appliquer au cas de l'épaisseur du manteau : lorsque l'on s'intéresse aux distributions de hauteur de neige mesurées au cours de l'hiver 2012-2013 (jeu RADOMEsd1213), les distributions des mesures diffèrent nettement plus, d'une station à l'autre, que les distributions des visibilités. D'autre part, ni l'espérance ni la médiane conditionnelle estimées à partir de RADOMEsd1213, ne sont croissantes par rapport à v. Il semble donc nécessaire de proposer d'autres solutions pour étalonner la prédiction, la voie la plus simple consistant à étendre le jeu à des comparaisons inter-séquences.

Chapitre 6

Conclusion

6.1 Rappel des principales étapes

Ce projet de thèse est né d'une rencontre entre un besoin exprimé par Météo-France en matière d'exploitation de données d'opportunité et de nouvelles possibilités en matière de traitement d'image. Le thème de la neige en plaine est apparu comme un terrain de rencontre idéal.
Ce phénomène à enjeu échappe encore en partie au réseau d'observation. Les capteurs dédiés sont rares et la télédétection par satellite est pénalisée par la couverture nuageuse. D'un autre côté, les caméras fixes offrent souvent un complément d'information utile et fiable aux prévisionnistes. Leurs images permettent de préciser si la neige tient. Elles contiennent une information sur l'intensité des chutes de neige et sur l'évolution de la couche de neige au sol (fonte, accumulation). Sur une partie des images, il est même possible de donner un encadrement grossier de l'épaisseur de neige. Dans ce travail de thèse, il s'agissait d'explorer des méthodes d'apprentissage récente -le deep learning- en vue d'extraire ces informations de manière automatique.
La nature des informations ciblées nous a conduit à travailler sur des problème de classification et des problèmes d'apprentissage des préférences sur trois paramètres d'intérêt : la visibilité, l'étendue et l'épaisseur du manteau neigeux. Nous avons aussi cherché à construire une estimation quantitative par encadrements.

Pour aborder ces problèmes par deep learning, il s'est d'abord agi de construire des jeux de données adaptés, c'est à dire des jeux d'images représentatifs de la diversité du réel, non redondants, et associés à une donnée cible fiable (chapitre 2).
Parmi les sources d'images webcam disponibles, nous avons choisi d'exploiter les archives AMOS pour la grande diversité des matériels et des scènes qu'elles contenaient. Pour concentrer les événements d'intérêt (chutes de neige avec tenue de la neige au sol), nous avons défini des fenêtres d'extraction à partir des champs de la réanalyse ERA-5, en utilisant un géo-référencement approximatif des webcam d'AMOS.

Ces séries ont été complétées par des images de webcam françaises que nous avons collectées pendant des épisodes de neige en plaine.
Enfin, une partie de ces séries ont été éclaircies à la main pour obtenir des séquences d'images peu redondantes (600 séquences, 25.000 images). Une autre partie de ces séries a été mise à l'écart pour d'éventuelles approches faiblement-supervisées.
Ces séquences ont ensuite été annotées. Pour produire des cibles fiables, nous avons choisi l'annotation manuelle. Cette modalité permettait aussi de caractériser la qualité de l'information disponible dans l'image. Les premiers labels ont été posés individuellement, de manière à définir des problèmes de classification à grain fin sur plusieurs attributs. Des attributs supplémentaires ont été sélectionnés afin de pouvoir mieux interpréter les résultats. Nous avons ainsi posé de l'ordre de 150.000 labels individuels caractérisant le niveau d'enneigement, la nature des précipitations et la qualité de l'image (gouttelettes, flocons sur la lentille, etc). Dans le même temps, nous avons posé des labels relatifs aux paires d'images consécutives, de façon à décrire les variations de la visibilité, de l'épaisseur et de l'étendue du manteau neigeux.
Nous avons ensuite fait le choix d'augmenter l'annotation par paires sur ces trois paramètres, par la comparaison d'images non consécutives. Pour accélérer l'annotation des paires non-consécutives, nous avons développé un algorithme de tri basé sur des comparaisons binaires et adapté à la présence de paires d'images incomparables. Porté sur les trois paramètres, cet algorithme a fourni environ 200.000 labels par paramètre pour un total d'environ 80.000 comparaisons manuelles supplémentaires.
Pour compléter cet ensemble de données, les images des dix caméras du réseau TENEBRE ont été annotées avec les données instrumentales colocalisées.

A partir de ces jeux, nous avons pu aborder plusieurs tâches d'apprentissage de natures différentes (chapitre 3). Sur les problèmes de classification les performances sont restées limitées. Par contre, les performances des modèles entraînés sur les paires d'images se sont montrées plus prometteuses.
Nous avons d'abord entraîné des réseaux de neurones standard à comparer les paires d'images sur le critère de la visibilité, suivant deux approches différentes, prédiction par paire et fonctions de rang. Ces approches ont été implémentées sur des réseaux de neurones à couches de convolution standard.
Nous avons d'abord montré l'intérêt de l'annotation des paires d'images non consécutives. Ces dernières permettent d'améliorer sensiblement les performances en généralisation des modèles entraînés à la prédiction par paire.
Nous montrons aussi que la relation d'incomparabilité peut être apprise et restituée, quoique imparfaitement, toujours dans le cadre de la prédiction par paire.
Dans un second temps, nous avons comparé la prédiction par paire aux fonctions de rang. Ces dernières offrent des performances plus faibles malgré un pré-entraînement sur ImageNet. Par contre, les deux approches donnent des performances supérieures aux méthodes

de l'état de l'art.

Par la suite (chapitre 4), nous avons cherché à améliorer les performances en généralisation des fonctions de rang parce qu'elles sont simples d'utilisation et qu'elles sont susceptibles d'être étalonnées. Pour le faire, nous nous sommes servis des séries d'images de la base AMOS mises de côté. La prédiction par paire a été utilisé pour annoter les images supplémentaires. Entraînées sur un jeu ainsi étendu (AMOSvvExt), les fonctions de rang atteignent des performances en généralisation forte meilleures de plus de deux points de justesse (de 89,8 % à 92,2 %) sur l'ensemble des paires strictement ordonnées (problème à deux classes).
Un deuxième développement touche à la limite des fonctions de rang, en matière d'apprentissage et de restitution des relations d'incomparabilité (troisième classe). Nous avons simplement étendu le concept de fonction de rang à la prédiction d'intervalle. Pour chaque image, les fonctions de rang « bivaluées » prédisent un intervalle de $\mathbb{R}$. Les chevauchements entre les intervalles rendent compte de l'incomparabilité. Nous avons montré que ces fonctions de rang apprenaient les causes d'incomparabilité les plus fréquentes, comme un défaut de qualité lié à des parasites (e.g. des gouttelettes, flocons sur la lentille, etc) ou une scène qui se prête moins à l'estimation relative de la visibilité (e.g. portail_entzheim, figure 2.4). Ces fonctions de rang bivaluées conduisent par ailleurs à une nouvelle amélioration des performances en généralisation sur le problème à deux classes.
Les deux développements qui ont suivi concernent l'étalonnage des fonctions de rang. Nous avons d'abord précisé sous quelles hypothèses [1] les intervalles prédits par les fonctions de rang pouvaient être étalonnés.
Nous nous sommes ensuite donné une méthode d'étalonnage. L'histogram matching, méthode relativement peu exigeante, a été sélectionné. Nous avons montré qu'il était envisageable d'exploiter les mesures faites à distance (dans un rayon de 50 km de la caméra) pour estimer la distribution locale et caler les quantiles de la fonction de rang tout en contrôlant le biais.
Le dernier développement contient une approche alternative, par « intercalibration ». Nous avons cherché à entraîner une fonction de rang universelle, c'est à dire capable de prédire des relations inter-séquences aussi bien que les relations intra-séquence, à travers une approche faiblement supervisée. Nous avons en montré le potentiel : ces nouvelles fonctions de rang, étalonnées une fois pour toutes sur l'une des caméras du réseau TENEBRE, proposent une prédiction relativement homogène sur toutes les caméras du réseau.

En parallèle aux précédents développements, nous avons appliqué l'approche semi-supervisée et les fonctions de rang bivaluées aux deux autres paramètres, c'est à dire à l'étendue et à l'épaisseur du manteau neigeux (chapitre 5).

1. Ces hypothèses portent principalement sur la qualité de l'annotation et les performance du modèle.

Pour l'étendue du manteau neigeux, nous nous retrouvions dans la situation rencontrée pour la visibilité : une prédiction par paires qui généralise mieux que la prédiction par image. Nous avons donc appliqué une stratégie d'apprentissage semi-supervisée analogue. Les modèles résultants discriminent nettement mieux les cas de neige au sol des cas où la neige est absente. De plus, la capacité des fonctions de rang bivaluées à indiquer un défaut de qualité dans l'image a pu être de nouveau vérifiée.
Pour l'épaisseur du manteau neigeux, les scores étaient plus faibles que sur le paramètre de l'étendue. Améliorer nos fonctions de rang sur ce paramètre ne semblait pas possible sans une phase d'annotation supplémentaire. Faute de temps, cette étape d'annotation supplémentaire n'a pas été mise en œuvre.
Au contraire, nous avons préféré valoriser les progrès déjà accomplis sur l'étendue. Une étude sur des séquences d'images échantillonnées pendant des épisodes de neige en plaine au cours de l'hiver 2020-2021, nous a permis d'évaluer le délai de prédiction d'un événement (tenue et accumulation de la neige au sol) à moins de 15 minutes en moyenne pour des scènes routières.

6.2 Bilan global sur les objectifs

Les objectifs de départ étaient de détecter la neige au sol et sur la route, de caractériser l'intensité des chutes de neige, de préciser les tendances de l'évolution du manteau, et d'évaluer un encadrement de l'épaisseur du manteau neigeux.
Vu sous l'angle du machine learning, ces objectifs impliquaient des cibles de nature catégorielles, ordinale et quantitative. Dans cette thèse, nous avons fait le choix de nous concentrer sur les cibles de nature ordinale pour ensuite étendre au quantitatif et au catégoriel.
Nous avons ainsi concentré les efforts d'annotation sur une seule classe de problèmes (apprentissage de préférences). Les options que nous avons prises ont débouché sur une situation inédite : un jeu composé de nombreuses séquences presque entièrement triées à la main par rapport à chacun des trois paramètres d'intérêt, et triées en tenant compte des paires incomparables. Ce jeu et l'algorithme qui a été développé pour le construire constituent notre première contribution.
Le fait de choisir une approche par apprentissage des préférences n'est pas un point orignal de la thèse. Les travaux d'Islam et al. [18] contiennent déjà l'idée d'un tri automatique des séquences d'images. En 2019, You et al. [37] construisent aussi leur jeu de données à partir de paires comparées à la main sur le paramètre visiblité. Mais nous sommes les premiers à rapporter un effet positif d'une annotation par tri complet des séquences d'entraînement sur les performances en généralisation.

Pour apprendre sur ces jeux, les fonctions de rang implémentées sur des réseaux de neurones à couches de convolution standard constituaient notre outil principal. Les deux mé-

thodes génériques qui ont été développées pour en améliorer les performances sur des caméras quelconques constituent notre deuxième contribution. La première méthode a consisté à utiliser la prédiction par paires d'images concaténées, qui généralise mieux, pour construire un jeu étendu sur lequel apprendre nos fonctions de rang. Sur les deux paramètres pour lesquels cette approche est mise en oeuvre (visibilité et étendue du manteau), les performances en généralisation sont améliorées. C'est, à notre connaissance, la première fois que la prédiction par paire est utilisée pour guider la prédiction par image.
La seconde méthode a consisté à étendre la notion de fonction de rang à la prédiction d'intervalles, de manière à apprendre sur les relations d'incomparabilités. Nous avons amené ces fonctions à prédire un intervalle plus large pour des images plus souvent jugées incomparables. Sur les deux paramètres visibilité et étendue, la largeur des intervalles reflète bien, après apprentissage, le défaut d'information de l'image vis à vis du paramètre d'intérêt. Cette extension conceptuelle, simple et originale, est un premier pas vers la prédiction d'encadrements sur les paramètres d'intérêt.

A partir des fonctions de rang, nous avons exploré des applications en matière de prédiction qualitative et en matière de classification.
Les performances des fonctions de rang relatives à l'épaisseur du manteau n'étaient pas assez bonnes pour passer à l'estimation quantitative de ce paramètre. Nous avons concentré nos efforts sur la visibilité. Différentes pistes ont été explorées, et pour chacune, des résultats préliminaires indiquent une possibilité d'étalonnage à partir de données distantes. Elles indiquent aussi un potentiel débouché pour la caractérisation de l'intensité des précipitations neigeuses.

Les fonctions de rang ont aussi été converties en classifieurs par seuillage de manière à être comparées aux classifeurs entraînés sur le problème à deux classe neige au sol / non neige au sol. Sur ce problème, les performances en généralisation des fonctions de rang sont nettement supérieures. Evidemment, cela tient à un effort d'annotation plus important. Mais il faut souligner un point : le jeu d'images utilisé est le même pour les classifieurs et pour les fonctions de rang. Or dans un contexte ou le phénomène d'intérêt est relativement peu représenté dans les archives disponibles, cette plus-value en performance à nombre d'images constant est très intéressante.

Enfin sur des événements de neige en plaine observés pendant l'hiver 2020-2021 les fonctions de rang ont été évaluées sur leur capacité à détecter un début d'extension du manteau (ou un début de tenue de la neige au sol, lorsque la neige n'est pas déjà présente). Les délais sont de l'ordre de 20 minutes en moyenne, pour tous types de caméra, et sont réduits à moins de quinze minutes pour des caméras routières. Cela ouvre des perspectives en matière de surveillance des conditions météorologiques.

6.3 Limites et perspectives

Une des limites de notre travail a trait aux comparaisons avec l'existant. Les quelques comparaisons qui sont faites ne sont pas réalisées sur les mêmes caméras, faute de disposer des jeux complets utilisés par les auteurs. Il aurait fallu pouvoir disposer d'un jeu pour la recherche (la plupart des jeux disponibles sont peu fiables). Nous espérons, à ce titre, que la publication des annotations relatives aux caméras d'AMOS permettra de changer cette situation.

Dans cette thèse, l'accent a plutôt été mis sur les développements d'ordre méthodologique que sur les applications pratiques. En effet, les contraintes matériels de cette thèse ne permettaient pas de pousser à leur maximum les méthodes explorées. Ces dernières, par essence, fonctionnent mieux avec plus de données.
Soulignons à cet égard un aspect important : dans une perspective opérationnelle, il ne sera pas possible d'utiliser les mêmes données. En particulier, la base AMOS ne peut a priori pas être directement utilisée pour construire et vendre une prévision automatique à des clients de Météo-France. Une alternative consisterait à proposer un service gratuit (une estimation quantitative à partir de l'image) aux propriétaires de webcams, qui seraient invités à collaborer au projet en cédant leurs droits sur une petite partie des images.

Pour revenir à des perspectives d'ordre scientifique, considérons les défis qu'il reste à relever. Nous avons déjà parlé de l'estimation relative de l'épaisseur du manteau neigeux. Pour ce paramètre, il nous semble nécessaire de recueillir davantage de données. Soulignons que les modèles disponibles peuvent faciliter cette opération.
Un autre défi concerne les images de nuit. A ce propos, il conviendra de faire la différence entre le suivi du manteau et le suivi de la visibilité. La méthodologie développée peut s'appliquer aux paramètres relatifs au manteau neigeux lorsque les images de nuit sont intégrées au jeu de données. Mais pour la visibilité, la situation est différente dans la mesure ou des informations supplémentaires sont disponibles la nuit à travers les sources d'éclairage articielles (halos, atténuation). Cette situation est par nature incompatible avec un ordre d'intervalle.
Par contre d'autre paramètres intéressant la surveillance de l'environnement se prêteraient à la méthode d'annotation développée dans cette thèse, comme la nébulosité, la hauteur des vagues ou l'étendue d'un feu de forêt, la hauteur de l'eau lors d'une inondation, etc. Ces paramètres pourraient ensuite être abordés avec les mêmes méthodes.

Nous voulons terminer cette conclusion sur une dernière perspective d'ordre méthodologique. A aucun moment, dans ce manuscrit, l'aspect temporel n'a été exploité (sauf à la marge, pour la construction des jeux étendus).
Cet aspect a volontairement été mis de côté : il nous apparaissait plus important, dans un premier temps, de chercher à tirer le maximum d'information de l'image, ou d'une paire d'images.

Il semblerait maintenant intéressant d'évaluer l'intérêt de la prise en compte des images du passé. Cela pose un nouveau défi d'ordre méthodologique : il faudra pouvoir apprendre les réseaux récurrents -outil de prédilection pour une intégration temporelle, à partir d'une annotation par paires.

Bibliographie

[1] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv :1409.1556*, 2014.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[3] G. A. Pagani, J. W. Noteboom, and W. Wauben, “Deep neural network approach for automatic fog detection using traffic camera images,” 2018.

[4] T. M. Kwon, “An automatic visibility measurement system based on video cameras,” 1998.

[5] D. S. Raina, N. J. Parks, W.-W. Li, R. W. Gray, and S. L. Dattner, “Innovative monitoring of visibility using digital imaging technology in an arid urban environment,” in *Regional and Global Perspectives on Haze : Causes, Consequences and Controversies Visibility Specialty Conference*, Citeseer, 2004.

[6] J.-J. Liaw, S.-B. Lian, Y.-F. Huang, and R.-C. Chen, “Atmospheric visibility monitoring using digital image analysis techniques,” in *International Conference on Computer Analysis of Images and Patterns*, pp. 1204–1211, Springer, 2009.

[7] N. Graves and S. Newsam, “Using visibility cameras to estimate atmospheric light extinction,” in *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 577–584, IEEE, 2011.

[8] R. Salvatori, P. Plini, M. Giusto, M. Valt, R. Salzano, M. Montagnoli, A. Cagnati, G. Crepaz, and D. Sigismondi, “Snow cover monitoring with images from digital camera systems,” *Ital. J. Remote Sens*, vol. 43, no. 6, 2011.

[9] S. G. Narasimhan and S. K. Nayar, “Vision and the atmosphere,” *International Journal of Computer Vision*, vol. 48, no. 3, p. 233–254, 2002.

[10] S. G. Narasimhan and S. K. Nayar, “Shedding light on the weather,” in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA*, pp. 665–672, 2003.

[11] R. Garg, V. K. BG, G. Carneiro, and I. Reid, “Unsupervised cnn for single view depth estimation : Geometry to the rescue,” in *European Conference on Computer Vision*, pp. 740–756, Springer, 2016.

[12] S. K. Nayar and S. G. Narasimhan, “Vision in bad weather,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 820–827, IEEE, 1999.

[13] P. Duthon, *Descripteurs d’images pour les systèmes de vision routiers en situations atmosphériques dégradées et caractérisation des hydrométéores*. PhD thesis, 2017,.

[14] N. Hautière, R. Babari, É. Dumont, R. Brémond, and N. Paparoditis, “Estimating meteorological visibility using cameras : A probabilistic model-driven approach,” in *Asian Conference on Computer Vision*, pp. 243–254, Springer, 2010.

[15] C. W. Srinivasa G. Narasimhan, , and S. K. Nayar, “All the images of an outdoor scene,” pp. 148–162, 2002.

[16] L. Xie, M. A. Carreira-Perpinán, and S. Newsam, “Semi-supervised regression with temporal image sequences,” in *2010 IEEE International Conference on Image Processing*, pp. 2637–2640, IEEE, 2010.

[17] X.-C. Yin, T.-T. He, H.-W. Hao, X. Xu, X.-Z. Cao, and Q. Li, “Learning based visibility measuring with images,” in *International Conference on Neural Information Processing*, pp. 711–718, Springer, 2011.

[18] R. Babari, N. Hautière, É. Dumont, N. Paparoditis, and J. Misener, “Visibility monitoring using conventional roadside cameras–emerging applications,” *Transportation research part C : emerging technologies*, vol. 22, pp. 17–28, 2012.

[19] R. Hallowell, M. Matthews, and P. Pisano, “An automated visibility detection algorithm utilizing camera imagery,” in *23rd Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology (IIPS)*, 2007.

[20] N. Jacobs, N. Roman, and R. Pless, “Consistent temporal variations in many outdoor scenes,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–6, June 2007. Acceptance rate : 23.4%.

[21] M. Islam, N. Jacobs, H. Wu, and R. Souvenir, “Images+weather : Collection, validation, and refinement,” 01 2013.

[22] D. Glasner, P. Fua, T. Zickler, and L. Zelnik-Manor, “Hot or not : Exploring correlations between appearance and temperature,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, vol. 00, pp. 3997–4005, Dec. 2015.

[23] Z. Chen, F. Yang, A. Lindner, G. Barrenetxea, and M. Vetterli, “How is the weather : Automatic inference from images,” *Proceedings of IEEE International Conference on Image Processing (ICIP 2012)*, 2012.

[24] N. Graves and S. Newsam, “Camera-based visibility estimation : Incorporating multiple regions and unlabeled observations,” *Ecological informatics*, vol. 23, pp. 62–68, 2014.

[25] S. Varjo and J. Hannuksela, “Image based visibility estimation during day and night,” in *Asian Conference on Computer Vision*, pp. 277–289, Springer, 2014.

[26] A. Volokitin, R. Timofte, and L. V. Gool, “Deep features or not : Temperature and time prediction in outdoor scenes,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1136–1144, June 2016.

[27] W. Chu, K. Ho, and A. Borji, “Visual weather temperature prediction,” *CoRR*, vol. abs/1801.08267, 2018.

[28] W.-T. Chu, X.-Y. Zheng, and D.-S. Ding, “Camera as weather sensor : Estimating weather information from single images,” *Journal of Visual Communication and Image Representation*, vol. 46, pp. 233 – 249, 2017.

[29] M. Roser and F. Moosmann, “Classification of weather situations on single color images,” in *2008 IEEE Intelligent Vehicles Symposium*, pp. 798–803, June 2008.

[30] X. Yan, Y. Luo, and X. Zheng, “Weather recognition based on images captured by vision system in vehicle,” in *Advances in Neural Networks – ISNN 2009* (W. Yu, H. He, and N. Zhang, eds.), (Berlin, Heidelberg), pp. 390–398, Springer Berlin Heidelberg, 2009.

[31] C. Lu, D. Lin, J. Jia, and C.-K. Tang, “Two-class weather classification,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[32] M. Elhoseiny, S. Huang, and A. Elgammal, “Weather classification with deep convolutional neural networks,” in *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 3349–3353, Sep. 2015.

[33] Z. Zhang, H. Ma, H. Fu, and C. Zhang, “Scene-free multi-class weather classification on single images,” *Neurocomput.*, vol. 207, pp. 365–373, Sept. 2016.

[34] J. C. V. Guerra, Z. Khanam, S. Ehsan, R. Stolkin, and K. D. McDonald-Maier, “Weather classification : A new multi-class dataset, data augmentation approach and comprehensive evaluations of convolutional neural networks,” *CoRR*, vol. abs/1808.00588, 2018.

[35] K. Dahmane, P. Duthon, F. Bernardin, M. Colomb, F. Chausse, and C. Blanc, “Weathereye-proposal of an algorithm able to classify weather conditions from traffic camera images,” *Atmosphere*, vol. 12, no. 6, p. 717, 2021.

[36] C. Murdock, N. Jacobs, and R. Pless, “Webcam2satellite : Estimating cloud maps from webcam imagery,” in *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 214–221, Jan 2013.

[37] J. Wang, M. Korayem, S. Blanco, and D. J. Crandall, “Tracking natural events through social media and computer vision,” in *Proceedings of the 24th ACM International Conference on Multimedia*, MM ’16, (New York, NY, USA), pp. 1097–1101, ACM, 2016.

[38] M. Kosmala, K. Hufkens, and A. D. Richardson, “Integrating camera imagery, crowdsourcing, and deep learning to improve high-frequency automated monitoring of snow at continental-to-global scales,” *PloS one*, vol. 13, no. 12, p. e0209649, 2018.

[39] Y. You, C. Lu, W. Wang, and C.-K. Tang, “Relative cnn-rnn : Learning relative atmospheric visibility from images,” *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 45–55, 2019.

[40] A. R. Syed, “A review of cross validation and adaptive model selection,” 2011.

[41] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, 1992.

[42] T. K. Ho, “The random subspace method for constructing decision forests,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet : A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[45] H.-Y. Zhou, B.-B. Gao, and J. Wu, “Sunrise or sunset : Selective comparison learning for subtle attribute recognition.,” *CoRR*, vol. abs/1707.06335, 2017.

[46] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays, “Transient attributes for high-level understanding and editing of outdoor scenes,” *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, vol. 33, no. 4, 2014.

[47] A. Benoit, L. Cuevas, and J.-B. Thomas, “Deep learning for dehazing : Comparison and analysis,” *arXiv preprint arXiv :1806.10923*, 2018.

[48] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, “Clearing the skies : A deep network architecture for single-image rain removal,” *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, 2017.

[49] Y.-F. Liu, D.-W. Jaw, S.-C. Huang, and J.-N. Hwang, “Desnownet : Context-aware deep network for snow removal,” *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 3064–3073, 2018.

[50] K. Zhang, R. Li, Y. Yu, W. Luo, C. Li, and H. Li, “Deep dense multi-scale network for snow removal using semantic and geometric priors,” *arXiv preprint arXiv :2103.11298*, 2021.

[51] J. Valera, J. Valera, and Y. Gelogo, “A review on facial recognition for online learning authentication,” in *2015 8th International Conference on Bio-Science and Bio-Technology (BSBT)*, pp. 16–19, IEEE, 2015.

[52] J. Fürnkranz and E. Hüllermeier, “Preference learning and ranking by pairwise comparison,” in *Preference learning*, pp. 65–82, Springer, 2010.

[53] D. Parikh and K. Grauman, “Relative attributes,” in *2011 International Conference on Computer Vision*, pp. 503–510, IEEE, 2011.

[54] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li, “Listwise approach to learning to rank : theory and algorithm,” in *Proceedings of the 25th international conference on Machine learning*, pp. 1192–1199, 2008.

[55] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, “Signature verification using a" siamese" time delay neural network,” in *Advances in neural information processing systems*, pp. 737–744, 1994.

[56] Y. Souri, E. Noury, and E. Adeli, “Deep relative attributes,” in *Asian conference on computer vision*, pp. 118–133, Springer, 2016.

[57] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, “Adversarially learned inference,” *arXiv preprint arXiv :1606.00704*, 2016.

[58] O. Ronneberger, P. Fischer, and T. Brox, “U-net : Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[59] X.-J. Mao, C. Shen, and Y.-B. Yang, “Image restoration using convolutional auto-encoders with symmetric skip connections,” *arXiv preprint arXiv :1606.08921*, 2016.

[60] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[61] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[62] S. Ioffe and C. Szegedy, “Batch normalization : Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, pp. 448–456, PMLR, 2015.

[63] P. Luo, X. Wang, W. Shao, and Z. Peng, “Towards understanding regularization in batch normalization,” *arXiv preprint arXiv :1809.00846*, 2018.

[64] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, “Deep learning for emotion recognition on small datasets using transfer learning,” in *Proceedings of the 2015 ACM on international conference on multimodal interaction*, pp. 443–449, 2015.

[65] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, “What’s the point : Semantic segmentation with point supervision,” in *European conference on computer vision*, pp. 549–565, Springer, 2016.

[66] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” *CoRR*, vol. abs/1707.02968, 2017.

[67] D. Rolnick, A. Veit, S. Belongie, and N. Shavit, “Deep learning is robust to massive label noise,” *arXiv preprint arXiv :1705.10694*, 2017.

[68] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[69] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf : an astounding baseline for recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813, 2014.

[70] R. Caruana, “Multitask learning,” *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[71] G. Antipov, M. Baccouche, S.-A. Berrani, and J.-L. Dugelay, “Effective training of convolutional neural networks for face-based gender and age prediction,” *Pattern Recognition*, vol. 72, pp. 15–26, 2017.

[72] S. Ruder, “An overview of multi-task learning in deep neural networks,” *arXiv preprint arXiv :1706.05098*, 2017.

[73] A. Kendall, Y. Gal, and R. Cipolla, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7482–7491, 2018.

[74] A. Hertz, S. Fogel, R. Hanocka, R. Giryes, and D. Cohen-Or, “Blind visual motif removal from a single image,” *arXiv preprint arXiv :1904.02756*, 2019.

[75] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” *arXiv preprint arXiv :1803.07728*, 2018.

[76] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1422–1430, 2015.

[77] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders : Feature learning by inpainting,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

[78] L. C. Pickup, Z. Pan, D. Wei, Y. Shih, C. Zhang, A. Zisserman, B. Scholkopf, and W. T. Freeman, “Seeing the arrow of time,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2035–2042, 2014.

[79] I. Misra, C. L. Zitnick, and M. Hebert, “Shuffle and learn : unsupervised learning using temporal order verification,” in *European Conference on Computer Vision*, pp. 527–544, Springer, 2016.

[80] V. Vapnik and A. Vashist, “A new learning paradigm : Learning using privileged information,” *Neural networks*, vol. 22, no. 5-6, pp. 544–557, 2009.

[81] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik, “Unifying distillation and privileged information,” *arXiv preprint arXiv :1511.03643*, 2015.

[82] Z.-H. Zhou, “A brief introduction to weakly supervised learning,” *National Science Review*, vol. 5, no. 1, pp. 44–53, 2017.

[83] B. Frénay and M. Verleysen, “Classification in the presence of label noise : a survey,” *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2013.

[84] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, “Training convolutional networks with noisy labels,” *arXiv preprint arXiv :1406.2080*, 2014.

[85] Y. Wang, W. Liu, X. Ma, J. Bailey, H. Zha, L. Song, and S.-T. Xia, “Iterative learning with open-set noisy labels,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8688–8696, 2018.

[86] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, “Training deep neural networks on noisy labels with bootstrapping,” *arXiv preprint arXiv :1412.6596*, 2014.

[87] O. Petit, N. Thome, A. Charnoz, A. Hostettler, and L. Soler, “Handling missing annotations for semantic segmentation with deep convnets,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pp. 20–28, Springer, 2018.

[88] R. M. Rasmussen, J. Vivekanandan, J. Cole, B. Myers, and C. Masters, “The estimation of snowfall rate using visibility,” *Journal of Applied Meteorology and Climatology*, vol. 38, no. 10, pp. 1542–1563, 1999.

[89] T. Joachims, “Optimizing search engines using clickthrough data,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 133–142, ACM, 2002.

[90] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, A. Simmons, C. Soci, S. Abdalla, X. Abellan, G. Balsamo, P. Bechtold, G. Biavati, J. Bidlot, M. Bonavita, G. De Chiara, P. Dahlgren, D. Dee, M. Diamantakis, R. Dragani, J. Flemming, R. Forbes, M. Fuentes, A. Geer, L. Haimberger, S. Healy, R. J. Hogan, E. Hólm, M. Janisková, S. Keeley, P. Laloyaux, P. Lopez, C. Lupu, G. Radnoti, P. de Rosnay, I. Rozum, F. Vamborg, S. Villaume, and J.-N. Thépaut, “The era5 global reanalysis,” *Quarterly Journal of the Royal Meteorological Society*, vol. n/a, no. n/a.

[91] S. Bendickson, “Relationship between visibility and snowfall intensity,” tech. rep., 2003.

[92] D. E. Knuth, *The art of computer programming*, vol. 3. Pearson Education, 1997.

[93] C. Daskalakis, R. M. Karp, E. Mossel, S. J. Riesenfeld, and E. Verbin, “Sorting and selection in posets,” *SIAM Journal on Computing*, vol. 40, no. 3, pp. 597–622, 2011.

[94] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, “The balanced accuracy and its posterior distribution,” in *2010 20th international conference on pattern recognition*, pp. 3121–3124, IEEE, 2010.

[95] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, “Semi-supervised learning with deep generative models,” in *Advances in neural information processing systems*, pp. 3581–3589, 2014.

[96] W. Cheng, M. Rademaker, B. De Baets, and E. Hüllermeier, “Predicting partial orders : ranking with abstention,” in *Joint European conference on machine learning and knowledge discovery in databases*, pp. 215–230, Springer, 2010.

[97] S. Zagoruyko and N. Komodakis, “Learning to compare image patches via convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4353–4361, 2015.

[98] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

[99] W. W. Cohen, R. E. Schapire, and Y. Singer, “Learning to order things,” *Journal of artificial intelligence research*, vol. 10, pp. 243–270, 1999.

[100] S. Zagoruyko and N. Komodakis, “Learning to compare image patches via convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4353–4361, 2015.

[101] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[102] L. Pang, Y. Lan, J. Guo, J. Xu, J. Xu, and X. Cheng, “Deeprank : A new deep architecture for relevance ranking in information retrieval,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 257–266, 2017.

[103] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, “Learning to rank using gradient descent,” in *Proceedings of the 22nd international conference on Machine learning*, pp. 89–96, 2005.

[104] P. L. Bartlett and M. H. Wegkamp, “Classification with a reject option using a hinge loss.,” *Journal of Machine Learning Research*, vol. 9, no. 8, 2008.

[105] Z. Liu, Z. Wang, P. P. Liang, R. R. Salakhutdinov, L.-P. Morency, and M. Ueda, “Deep gamblers : Learning to abstain with portfolio theory,” *Advances in Neural Information Processing Systems*, vol. 32, pp. 10623–10633, 2019.

[106] Y. Geifman and R. El-Yaniv, “Selective classification for deep neural networks,” *arXiv preprint arXiv :1705.08500*, 2017.

[107] K. Tang, J. Yang, and J. Wang, “Investigating haze-relevant features in a learning framework for image dehazing,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2995–3000, 2014.

[108] R. D. Luce, “Semiorders and a theory of utility discrimination,” *Econometrica, Journal of the Econometric Society*, pp. 178–191, 1956.

[109] A. N. Arslan, C. M. Tanis, S. Metsämäki, M. Aurela, K. Böttcher, M. Linkosalmi, and M. Peltoniemi, “Automated webcam monitoring of fractional snow cover in northern boreal conditions,” *Geosciences*, vol. 7, no. 3, p. 55, 2017.

[110] P. C. Fishburn, “Intransitive indifference with unequal indifference intervals,” *Journal of Mathematical Psychology*, vol. 7, no. 1, pp. 144–149, 1970.

[111] M. Bousquet-Mélou, A. Claesson, M. Dukes, and S. Kitaev, “(2+2)-free posets, ascent sequences and pattern avoiding permutations,” *Journal of Combinatorial Theory, Series A*, vol. 117, no. 7, pp. 884–909, 2010.

[112] L. Liebel and M. Körner, “Auxiliary tasks in multi-task learning,” *arXiv preprint arXiv :1805.06334*, 2018.

[113] R. C. Gonzalez, R. E. Woods, *et al.*, “Digital image processing,” 2002.

[114] M. L. Puri, D. A. Ralescu, and L. Zadeh, “Fuzzy random variables,” in *Readings in fuzzy sets for intelligent systems*, pp. 265–271, Elsevier, 1993.

[115] M. Arefi, R. Viertl, and S. M. Taheri, “Fuzzy density estimation,” *Metrika*, vol. 75, no. 1, pp. 5–22, 2012.

[116] H.-S. Lee and M. OMahony, “Sensory difference testing : Thurstonian models,” *Food Science and Biotechnology*, vol. 13, no. 6, pp. 841–847, 2004.

[117] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, “Noise2noise : Learning image restoration without clean data,” *arXiv preprint arXiv :1803.04189*, 2018.

[118] E. A. Nadaraya, “On estimating regression,” *Theory of Probability & Its Applications*, vol. 9, no. 1, pp. 141–142, 1964.

[119] Y. Liu, A. W. K. Kong, and C. K. Goh, “A constrained deep neural network for ordinal regression,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 831–839, 2018.

[120] J. Garvelmann, S. Pohl, and M. Weiler, “From observation to the quantification of snow processes with a time-lapse camera network,” *Hydrology and Earth System Sciences*, vol. 17, no. 4, pp. 1415–1429, 2013.

[121] C. Doersch and A. Zisserman, “Multi-task self-supervised visual learning,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2051–2060, 2017.

[122] Commission Internationale de L'éclairage, *International lighting vocabulary*. Commission Internationale de L'éclairage, 1987.

[123] Middleton, *Vision through the atmosphere*. University of Toronto Press, 1952.

[124] World Meteorological Organization, *Guide to Meteorological Instruments and Methods of Observation No. 8*. WMO, 2014. Geneva, Switzerland,(https ://www.wmo.int/pages/prog/www/ IMOP/CIMO-Guide.html).

[125] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank : from pairwise approach to listwise approach," in *Proceedings of the 24th international conference on Machine learning*, pp. 129–136, 2007.

Annexe A

Compléments sur l'annotation

A.1 Critères d'annotation

A.1.1 Labels relatifs à une image

Les paragraphes se rapportent aux attributs de la figure 2.5.

A.1.1.1 L'attribut état du sol

Lorsqu'un épisode de précipitations débute, les premiers indices sûrs sont bien souvent la couleur et la texture des surfaces artificielles. Le revêtement de la route s'assombrit. Lorsque les précipitations durent, un film d'eau liquide se forme sur la chaussée et des reflets apparaissent, plus marqués de nuit que de jour. Ce film peut aussi être trahi par la présence de « buissons » d'eau projetée en arrière des roues des véhicules. Ces éléments permettent de distinguer les cas de « route sèche » des autres. Le label **dry_road** n'est porté que lorsque la couleur, l'absence de reflets et de « buissons » excluent une route humide, mouillée ou enneigée. Dans le doute, on porte le label « **wet_road** ». Les labels suivants sont consacrés à des cas où la neige est visible au sol ; lorsque la neige a tenu en dehors de la chaussée, on utilise **snow_ground** si la chaussée est humide et **dry_road_snow_ground** si elle est sèche. **snow_road** est porté lorsque le manteau s'est étendu jusque sur la bande de roulement. Enfin, **white_road** est porté lorsque toute la chaussée est couverte.

Au début et à la fin de la période d'enneigement, il peut n'y avoir que quelques traces de neige au sol. Le label **ground** (attribut snow_traces, voir figure 2.5) est alors coché. De même, lorsque la neige ne tient qu'en quelques endroits de la chaussée, c'est signalé par **road** (attribut snow_traces).

Enfin, lorsque faute d'une visibilité suffisante, l'état du sol n'est pas connu avec certitude, on porte le label **doubt**. Sur un nombre non négligeable d'images prises de nuit, seul l'enneige-

ment de la chaussée peut être exclu. On utilise alors le label **no_snow_road**.

Le choix du label a pu être compliqué par les situation de contrejour avec reflets sur la voie, les motifs blancs apparaissant sur les routes en vision nocturne, la salaison des routes ou encore la gelée blanche (figure A-1).

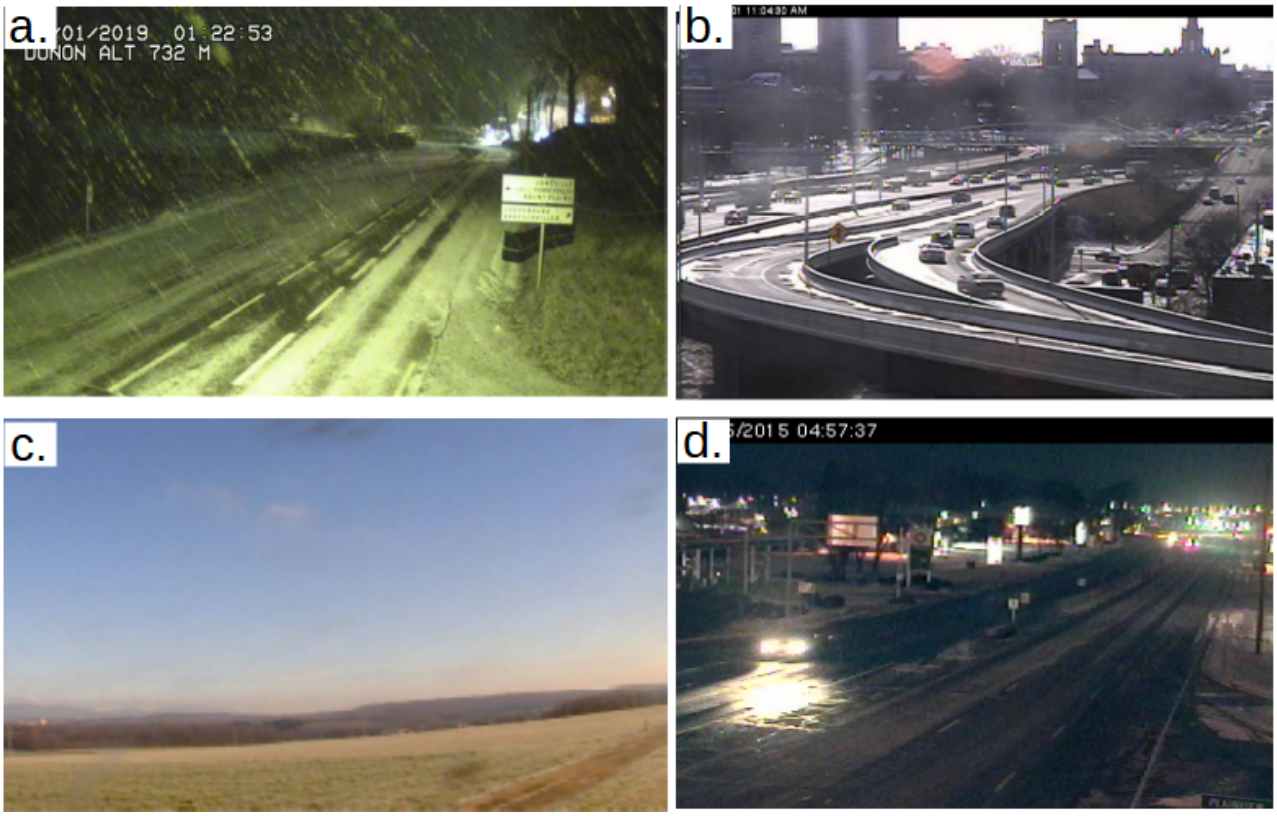


FIGURE A-1 – Le contraste entre les parties claire et sombre de la chaussée sont dus à l'humidité. b. la route est exposée au soleil après une averse de neige. Les reflets du soleil peuvent donner l'impression que la route est couverte de neige. c. un cas de gelée blanche. d. sel fraîchement déposé sur la voie. Contrairement au routes enneigées, le sel s'étale depuis le centre.

A.1.1.2 Flocons en chute libre (snowfall)

Dans le cas où des flocons en chute libre apparaissent dans l'image, on porte le label **streaks**. De nuit, ils apparaissent sous forme de traînées, à cause des temps de pause plus longs (figure A-2.d).

Ces traînées ne sont visibles qu'autour des sources de lumières (lampadaires, phares de voiture, lampe associée à la caméra). De jour, ce sont des tâches blanches plus ou moins allongées, aux contours généralement flous, qui ne sont visibles que dans les parties sombres de l'image, par contraste. Lorsqu'un doute existe sur la nature des tâches/traînées, on porte le label **doubt**. Il faut faire attention à certains phénomènes « mimétiques » comme le passage d'oiseaux ou la « neige » parasite. Notons que les flocons ne sont qu'assez rarement

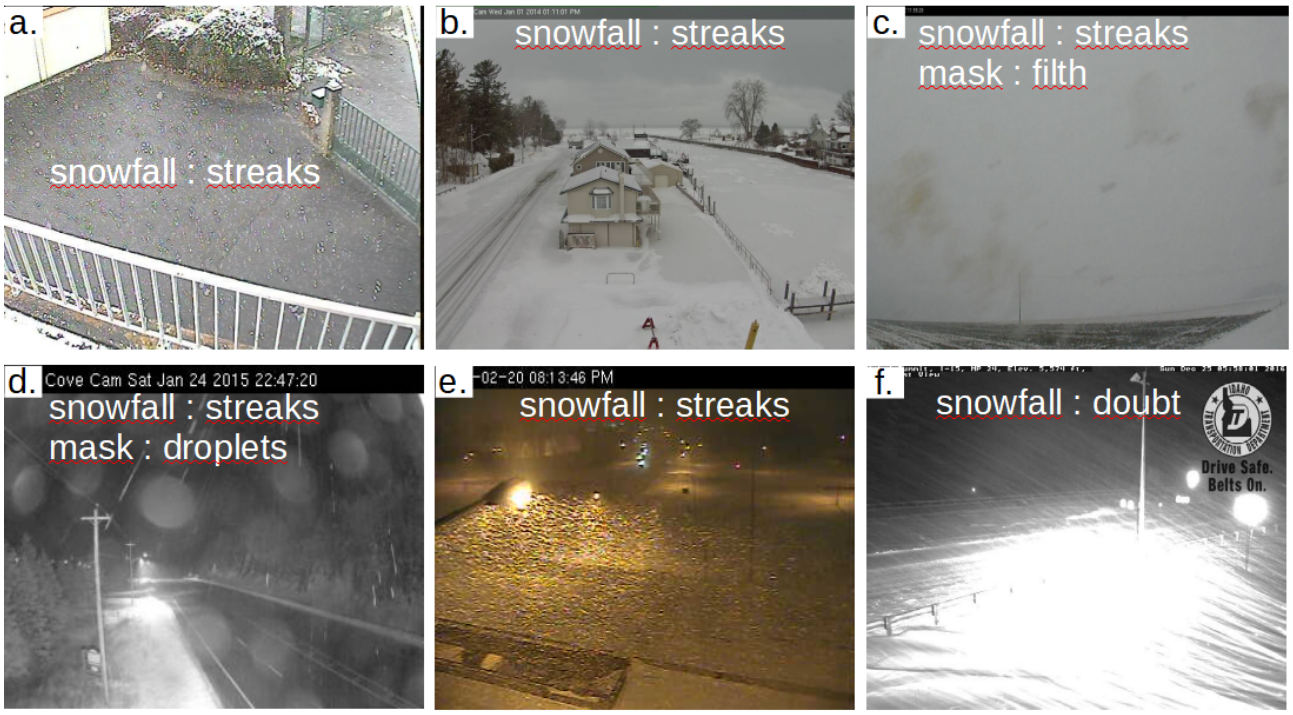


FIGURE A-2 – Première ligne : flocons en vol, de jour. a. trainées pixelisées visibles sur toute l'image. b. les flocons ne sont visibles que par contraste, devant les sapins. c. quelques flocons au premier plan. Des traces de saleté sont aussi visibles sur la vitre de protection. Seconde ligne. De nuit. d. la source de lumière est proche de la caméra, la trainée laissée par les flocons est plus grande que sur l'image suivante (e.), où les flocons n'aparaissent que sous les réverbères. f. sur cette image, il ne s'agit probablement pas de flocons en vol, mais de neige soufflée par le vent.

visibles sur l'image alors que des chutes neige ont lieu. Dans les situations hivernales échantillonnées, les gouttes d'eau ne donnent que très rarement lieu à des traînées.

A.1.1.3 Gouttes/flocons sur la lentille (mask) :

Selon la direction du vent et la vitesse du vent, l'intensité des précipitations, des hydrométéores viennent se coller à la lentille de la caméra ou à la vitre de protection. Sur l'image, ce phénomène peut se traduire par des effets variés qui dépendent de la nature des hydrométéores.
Les gouttes d'eau collées conservent généralement leur forme ronde. Elles peuvent être nettes ou focalisées. Mais elles ne sont pas opaques. Au contraire, les flocons présentent souvent des formes irrégulières, et sont généralement opaques. De nuit, la lumière réfractée par l'eau liquide forme des motifs reconnaissables (voir figure A-2).

Le label **droplets** signale des gouttes d'eau liquide tandis que **snowflake** signale les flocons (voir figure A-3) et les cas de doute ou de mélange entre flocons et gouttelettes. Lorsque plus de la moitié de l'image est affectée par la présence d'hydrométéores, les autres labels sont difficiles à porter. On le signale par les labels supplémentaires **acc_droplets** et **acc_snowflakes**.

Enfin, notons que ces hydrométéores collés à la lentilles peuvent être confondus avec d'autres traces, avec des flocons en vol ou avec des reflets d'objectifs.

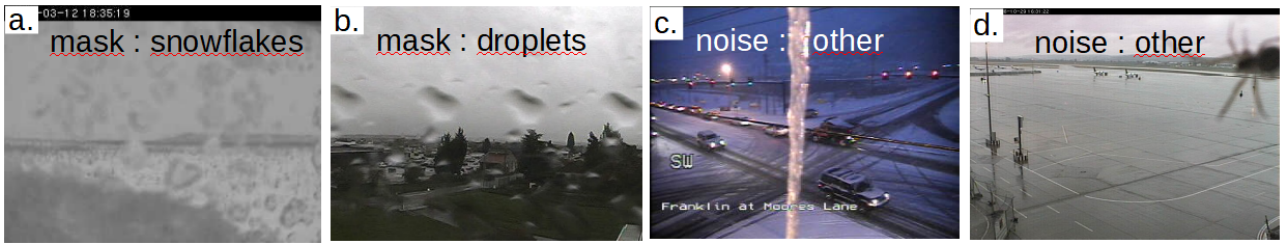


FIGURE A-3 – Quatre masque physiques. Les flocons sur la lentille, et les gouttes sur la vitre de protection sont très fréquents pendant les épisodes de précipitation. Les stalactites, les araignées sont plus anecdotiques, mais susceptibles de biaiser les prédictions sur de longues séquences d'images.

A.1.1.4 Typologie des précipitations (atmo)

Pour exprimer la nature des précipitations telle qu'elle apparaît sur l'image, nous avons retenu sept classes qui correspondent aux cas de figures les plus fréquents. Ces attributs sont assez largement corrélés avec les attributs précédents.

Lorsque la nébulosité ou le genre des nuages présents sont incompatibles avec l'occurrence de précipitations, et en dehors des cas de brouillard ou de brume, on porte le label **no_precip**. Lorsque le ciel n'est pas visible, les reflets du soleil, les ombres portées au sol et les ombres propres sont utilisés comme critères. Une chaussée sèche est un autre élément dont on peut tenir compte en cas de doute.

Les précipitations sont en général difficiles à distinguer les unes des autres dans la mesure où les hydrométéores sont le plus souvent invisibles sur les images webcam. Le label **precip** est utilisé lorsque la chaussée est mouillée, que la nébulosité apparente est maximale (8 octats), et qu'un « voile » vient réduire la portée optique, sans autre indice supplémentaire. Avec d'autres éléments, on peut choisir un label plus précis :

- si la neige au sol s'est accumulée et/ou que des flocons sont visibles, soit en chute libre, soit contre la lentille de la caméra, on porte le label **snow**.
- si la portée optique reste relativement grande, qu'il n'y a pas de neige au sol, mais que la chaussée est « mouillée » et/ou que des gouttelettes sont présentes depuis peu de temps sur la lentille (ie, absentes de l'image précédente), on porte le label **rain**.
- si la portée optique est réduite au premier plan et que les images voisines évoquent un brouillard : **fog** ou, dans le doute, **fog or snow**.

Dans les autres situations (précipitations possibles mais pas sûres, brume) le label **doubt** est porté. Il est d'autant plus courant que la résolution de l'image est mauvaise et que la ligne de mire de la caméra est inclinée vers le sol.

A.1.1.5 Autres phénomènes parasites

Nous avons vu que la qualité de l'image pouvait être affectée par les gouttes de pluies et les flocons sur la lentille. En fait, de nombreux autres phénomènes peuvent réduire la qualité de l'image. Nous en avons annoté quelques-uns, choisis selon trois critères :

— leur ressemblance avec les phénomènes météorologiques d'intérêt.
— leur impact sur la précision avec laquelle les paramètres météorologiques d'intérêt peuvent être évalués.
— leur fréquence

Parmi ces phénomènes supplémentaires, on compte d'abord les dépôts de matière opaque (label **filth**) et les reflets d'objectifs (label **artefacts**) illustrés figure A-2, qui pourraient passer pour des gouttes collées à la lentille ou des flocons, collés ou en vol.
D'autres exemples sont donnés par les cas de surexposition (label **surexp**). De jour, c'est souvent le cas lorsque le soleil, la route et la caméra sont dans le même plan. Le reflet du soleil sur la route sature le capteur et la route apparaît blanche. Selon l'orientation de la caméra, ces images peuvent être relativement fréquentes. Elles sont potentiellement confondues avec des images de route enneigée (figure A-4. a.). De nuit, les sources de lumière peuvent saturer le capteur et produire un voile artificiel qui rend l'estimation de la portée optique difficile (figure A-4, b.).
Enfin, des zones de l'image peuvent être complètement masquées par des problèmes de transmission (label **rec**) ou plus rarement par le boîtier de protection de la caméra, des stalactites de glace, des toiles d'araignée, des inscriptions en dur dans l'image (points cardinaux, logos, rectangles de censure, horodatage), etc (label **other**). Sur certaines séquences, les problèmes de transmission peuvent affecter jusqu'à la moitié des images disponibles.

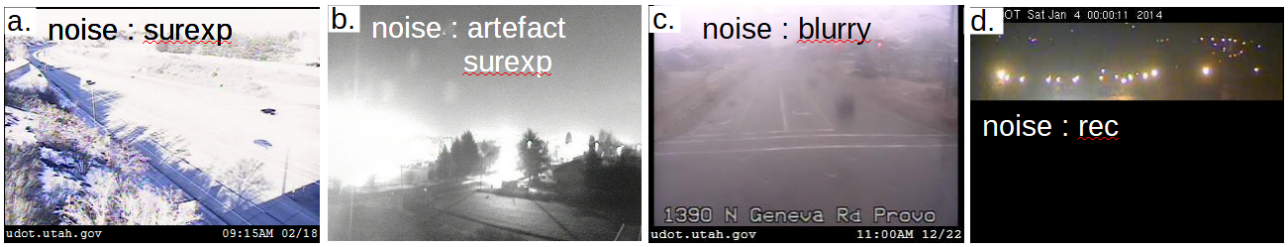


FIGURE A-4 – a. saturation du capteur de jour, b. saturation de nuit, pendant un épisode de neige. c. image floue (condensation). d. une partie de l'image manque.

A.1.1.6 Eclairage

L'horodatage des images correspond généralement à l'heure de prise de vue. La hauteur du soleil aurait donc pu être déduite automatiquement. Cependant, la météorologie locale, l'orientation et les paramètres intrinsèques des caméras, les multiples traitements appliqués à l'image et la configuration de l'éclairage artificiel nocturne éloignent les conditions d'éclai-

rage de celles qu'on observe par beau temps avec une caméra routière standard, sans post-traitement particulier.
Nous avons finalement distingué quatre catégories dont les contours sont relativement subjectifs. Les images de « jour » (label **day**) et de nuit sont délimitées par les images « crépusculaires » (label **inter**). Entre jour et crépuscule, le changement de label est décidé de la façon suivante : dans le cas général, la première image de la journée à partir de laquelle la lumière se met à décroître sensiblement est notée « **inter** ». Lorsque le temps est très couvert, ce qui est fréquent sur nos images, la décroissance de la luminosité s'échelonne sur plusieurs images. L'allumage des phares ou de l'éclairage public sert alors de balise. Dans le cas assez courant où le mode de fonctionnement de la caméra bascule à l'approche de la nuit, c'est à l'occasion du basculement qu'on change de label.
Les images annotées **inter** sont ainsi très variées, en terme de conditions d'éclairage. Lorsque, le soir, l'éclairage de la scène se stabilise à nouveau, le label passe à **night** ou **dark_night**. Le label **night** suppose un éclairage artificiel ou naturel (lune, lumière résiduelle) stable et suffisant pour pouvoir distinguer la neige au sol sur une large partie de l'image. Les scènes annotées **dark_night**, lorsqu'elles sont éclairées, ne le sont que de manière transitoire, généralement par le passage d'un véhicule.

A.1.1.7 Annotation du type de scène

Nous avons distingué quatre classes de scènes relativement homogènes. Les premières sont les autoroutes au sens large (label **highway**), c'est à dire toutes les routes à plus de quatre voies. Ces scènes présentent des avantages pour une approche automatique de la caractérisation de la météorologie. D'abord, il s'agit de scènes visuellement proches les unes des autres, relativement simples. Les caméras routières étant généralement inclinées avec un angle faible, ces scènes contiennent une part de ciel. Ce sont aussi des paysages assez dégagés, « profonds », qui permettent d'évaluer facilement la portée optique. Les talus, le terre-plein central, sont souvent couverts de terre et s'approchent de la surface standard sur laquelle la hauteur de neige est théoriquement estimée. Enfin, ce sont les scènes les plus nombreuses de la base AMOS.
Par contre, la présence de plusieurs voies, ou celle, fréquente, d'une route annexe sur l'image (le plus souvent, une bretelle d'accès), peut poser un problème de labellisation. En effet, il faut préciser la règle à suivre dans le cas où la neige tient sur l'une des voies, mais pas sur l'autre. Dans ce cas, nous nous en sommes toujours référés à la voie qui occupe la plus grande partie de l'image.
Proches de ces scènes, les routes de campagne (label **field_road**) comportent deux voies. Ces scènes laissent davantage de place aux surfaces naturelles, souvent plus accidentées. Les orientations des caméras sont plus variables, certaines caméras ayant même une visée au nadir. Plus variables encore sont les scènes urbaines (label **city**). La diversité des revêtements est importante. Dans ce cas, la plupart des surfaces visibles sont artificielles. Ces

scènes sont rarement aussi dégagées que les précédentes. Elles sont en général bien éclairées la nuit, mais l'éclairage peut beaucoup varier d'une image à l'autre, surtout en début de soirée et au matin.
Comme avec les autoroute, l'annotation peut poser problème lorsque plusieurs voies apparaissent. Nous suivons là-encore la règle de la voie principale.
Enfin, certaines scènes sont particulières, inclassables (label « **special** », figures A-5). Il s'agit par exemple d'une route côtière, d'un pont routier, d'un tunnel, une voie d'accès pour un bac, d'un parking, etc. Ces scènes constituent un challenge pour les modèles de prédiction.

FIGURE A-5 – Les deux premières (AMOS 1550 et 1002) sont des exemples de scènes urbaines, les deux autres sont des exemples de scènes spéciales. La troisième scène est une vue en plongée sur une route de campagne enneigée. Les créneaux en bas permettent d'estimer la hauteur de neige.

A.1.1.8 Autres labels qualitatifs

A des fins statistiques, nous comptabilisons aussi les débuts d'épisodes de tenue de la neige sur route (label **onset**), les rares cas de congère (**drift**) et de neige soufflée (**blown_snow**) . Enfin, nous précisons les cas où le passage d'une scène à l'autre implique un zoom ou une rotation sensible de la caméra (label **zoom**).

A.1.2 Annotation relative à une paire d'images (consécutives et non consécutives)

A.1.2.1 Critères de comparaison pour la visibilité

Pendant les deux premières étapes d'annotation, nous indiquons si, pour une paire d'images donnée, la visibilité sur la deuxième image est supérieure (label **farer**, notation : $\prec_v$), inférieure (label **farer**, notation : $\succ_v$), incomparable (label **eq3**, notation : $\perp_v$) ou équivalente (label **eq**, notation : $\sim_v$). Dans le premier paragraphe, on rappelle la définition de la visibilité puis l'on indique les critères utilisés. Lorsque aucun des critères n'est applicable ou lorsqu'ils conduisent à des diagnostics différents, l'annotateur rejette la comparaison. Quelques cas d'incomparabilité fréquents sont présentés. Enfin, on précise dans quelles situations les images peuvent être considérées comme « équivalentes ».

Définition et caractérisation de la visibilité

La visibilité est définie comme la plus grande distance à laquelle un objet noir de dimensions suffisantes peut être reconnu de jour sur fond de ciel. Le seuil de contraste retenu par la CIE est de 5 % [122]. Sous les hypothèses d'un éclairage homogène et d'une atmosphère homogène, on peut faire le lien entre la visiblité et le coefficient d'extinction par la loi de Koschmieder [123]. Sous ces hypothèses, la luminance d'un point de la scène se décompose en deux contributions : celle de l'atmosphère et celle de l'objet qui est vu à travers. Pour une caméra sans post-traitement et avec une fonction de gain linéaire, cette loi s'écrit [11] :

$$I = I_0 e^{-kd} + A_\infty (1 - e^{-kd}) \tag{A-1}$$

où I est l'intensité du pixel, A_∞ est l'intensité du ciel, I_0 est l'intensité du pixel « sans atmosphère ». k est le coefficient d'extinction et d la distance entre la caméra et l'élément de la scène couvert par pixel. On en déduit la loi d'atténuation des contrastes [1] par rapport au ciel :

$$C = \frac{(A_\infty - I_0)}{A_\infty} e^{-kd} \tag{A-2}$$

où $C \triangleq (A - I)/A$. De l'équation (A-2) et de la définition on déduit le lien entre la visibilité et le coefficient d'extinction du modèle :

$$v \approx -3/k$$

Ainsi :

$$I = I_0 e^{-3d/v} + A_\infty (1 - e^{-3d/v}) \tag{A-3}$$

critères de comparaison :

- **critère n°1** : critère de la portée optique.
  Ce premier critère est grossier, mais il suffit souvent pour décider : lorsqu'une partie des sols ou des objets visibles sur l'image a ne sont plus apparents sur l'image b, on décide $a \prec_v b$. Elle ne s'applique pas lorsque des objets situés en hauteur sont masqués par un nuage.
- **critère n°2** : blanchiment des objets lointains.
  Dans l'équation A-3, l'intensité du pixel apparaît comme une moyenne pondérée des contributions de l'atmosphère (« voile ») et de l'objet vu à travers. Sur deux images consécutives, lorsque l'intensité du ciel reste constante, mais que la visibilité baisse, l'intensité du pixel est tirée vers celle du ciel. Ce voile, d'autant plus prononcé que la

1. Il s'agit ici d'un contraste par rapport au fond (le ciel), et non du contraste de Michelson, où le dénominateur égale la moyenne des intensités. Dans la pratique, l'objet pris pour comparer les contrastes est sombre ($I_0 << A_\infty$) et les deux définitions sont proches.

distance est grande, permet de comparer rapidement des images consécutives, pour lesquelles de A_∞ (et celles de I_0) sont généralement très proches.
Ce critère de comparaison est d'autant plus fin que la scène est « profonde » (i.e. avec de nombreux objets situés à des distances diverses).

— **Critère n°3** : contraste par rapport au ciel.
Dans le cas d'images non-consécutives, la comparaison est souvent compliquée par des couleurs de ciel différentes. Dans ce cas, on peut comparer les contrastes avec le ciel. En effet, de jour, la luminance intrinsèque de l'objet peut être considérée comme proportionnelle à la luminance du ciel. On a alors $C_0 = (A_\infty - I_0)/A_\infty$ qui est indépendant de A_∞.

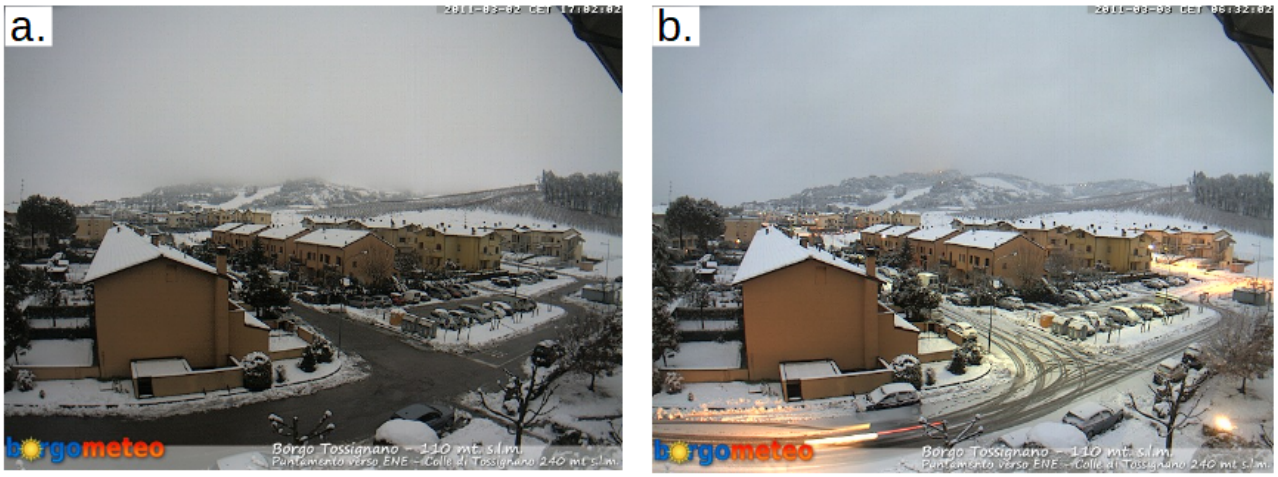


FIGURE A-6 – Sur a, le sommet de la colline est masqué par un plafond nuageux plus bas : on ne peut pas appliquer la règle 1. Par contre, le contraste à l'horizon est plus faible que sur b (chutes de neige en cours). On décide donc $b \prec_v a$.

— **Critère n°4** : contraste entre objets de même luminance.
Lorsque le ciel n'est pas visible ou que les luminances du ciel sont trop différentes, on peut s'appuyer sur le contraste qui apparaît entre les pixels quand la visibilité baisse. Ce contraste apparaît plus clairement sur des régions de l'image associées à des objets de même luminance intrinsèques et situés à des distances différentes comme sur la figure A-7.b : sur cette image, un contraste est apparu entre les arbres du premier plan et ceux à l'arrière plan. La visibilité a donc diminué.

Remarque : les critères 3 et 4, qui font intervenir une comparaison des contrastes, ont été appliquées avec discernement. En effet, le contraste local perçu dépend du contexte.
En particulier, dans des situations orageuses, en début ou en fin de journée, la scène reçoit peu de lumière et les contrastes apparaissent plus faibles. On décide donc plus facilement d'un ordre strict lorsque c'est sur l'image la plus sombre que le contraste entre les régions paraît le plus fort.
D'autre part, les hypothèses sur lesquelles ces critères sont basées sont à garder en tête. Par

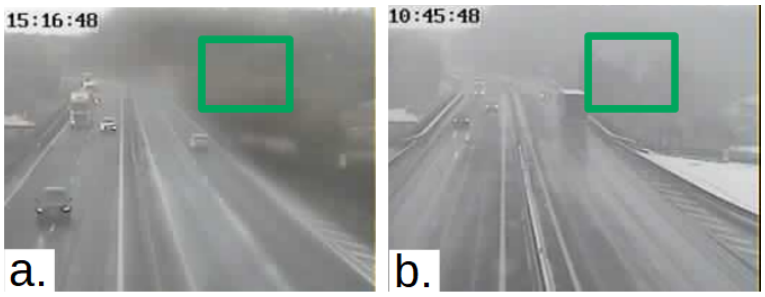


FIGURE A-7 – Règle des contrastes entre objets de même luminance. Un constraste apparaît dans le carré vert sur l'image b. On décide alors $a \prec_v b$.

exemple, les luminances intrinsèques des surfaces peuvent changer d'une image à l'autre. C'est le cas lorsque l'on compare une image sans neige à une scène enneigée. En particulier, le contraste forêt-ciel est amoindri lorsque les arbres sont couverts de neige. Cela complique la comparaison, et l'annotateur choisit le label 3 (incomparabilité) plus fréquemment que pour une paire d'images sans neige. D'autres situations particulières sont présentées dans les paragraphe suivants :

— Cas des scènes fermées.
  Sur certaines scènes, le plus lointain objet visible est à moins de dix mètres. C'est le cas des vues en plongée (e.g. figure A-5) ou de scènes urbaines sans second plan (e.g. portail_entzheim figure 2.4). Sur ces scènes, en général, seuls deux niveaux de visibilité peuvent être distingués : on reconnaît un temps ensoleillé aux ombrages, comme dans [28] et les chutes de neige aux flocons en vol. On ne décide alors d'ordre strict qu'entre ces deux catégories d'images. Rarement, le brouillard est assez dense pour blanchir les objets les plus éloignés. Dans ce cas, on observe souvent un effet de saturation qu'il ne faut pas confondre avec un cas de lumière rasante (voir plus bas).

— Cas des images en lumière crépusculaire et de nuit.
  A l'aube, au crépuscule et par une nuit claire, le premier critère est appliqué avec prudence. On complète le critère de la portée optique avec ceux de l'extinction des sources lumineuses lointaines et de l'amplification des halos autour des sources proches (figure A-8). Comme la deuxième étape d'annotation (chapitre 2) n'a pas été étendue à ces images, ces règles restent à préciser.

Cas d'incomparabilité :

— Cas d'incomparabilité 1 : Contradictions.
  Lorsque les règles se contredisent selon la région de l'image où on les applique, on décide une relation d'incomparabilité. Par exemple, sur la figure A-9, une nappe de

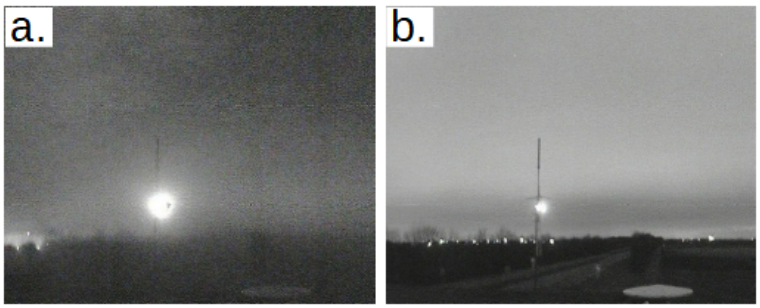


FIGURE A-8 – Halo amplifié par les mauvaises conditions météorologiques.

brouillard discontinue (ou un stratus très bas) limite la portée optique (a.). Un épisode de neige homogène affecte l'image en b. Les règles de contrastes appliquées aux conifères des deux côtés de la route, impliquent $b \prec_v a$, mais la portée optique la plus grande est sur l'image b.
De façon générale, les images sur lesquelles l'atmosphère n'est visiblement pas homogène génèrent davantage d'incomparabilités.

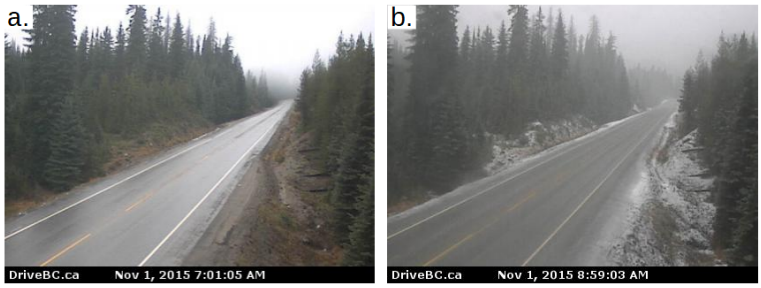


FIGURE A-9 – Un banc de brouillard réduit la portée optique sur l'image a. Elle est moins grande que sur l'image b prise lors d'un épisode de neige. Mais les critères 3-4 indiquent $b \prec_v a$. On décide donc l'incomparabilité ($a \perp_v b$).

— Cas d'incomparabilité 2 : contrejours.
Par mauvais temps, la couche nuageuse est généralement assez épaisse pour tamiser la lumière. L'hypothèse d'un éclairage homogène est justifiée. Mais par beau temps ou lorsque la brume (ou le brouillard) sont d'extension verticale limitée, le facteur A_∞ dépend de la position du soleil. En particulier, il augmente sensiblement lorsque le soleil est dans (ou est proche) du champ de la caméra. Quand cette situation est identifiée, on ne tient pas compte des critères 2-4. L'incomparabilité est donc décidée plus souvent.

— Cas d'incomparabilité 3 : parasites.
Pendant et après les précipitations, il arrive souvent que le champ de la caméra soit plus ou moins masqué par des gouttelettes ou des flocons ou de la buée. L'application des critères 1-4 en devient difficile. Les images affectées par un bruit génèrent ainsi

plus d'incomparabilité.

Cas d'équivalence :
Les cas d'équivalence se rencontrent le plus souvent au moment de l'annotation des paires consécutives. Cependant, à cause de la règle 3, on ne porte pas le label 2 à plus de trois paires ou quatre paires consécutives.
Sur les paires non-consécutives, deux cas d'équivalence se présentent souvent. Le premier cas concerne les images annotées par le label **no_precip** (c'est à dire sans précipitation, sans brouillard, ni brume). Néanmoins, on n'a pas rangé directement toutes les images de cette catégorie dans une même classe d'équivalence, à cause des fréquentes exceptions. A ce titre, il aurait été raisonnable d'ajouter une classe qualitative « visibilité max. » à partir de laquelle une classe d'équivalence aurait pu être formée.

A.1.2.2 Critères de comparaison pour le manteau neigeux

Pendant la première étape d'annotation (images consécutives), nous indiquons si, pour une paire d'image donnée, l'étendue du manteau neigeux sur la deuxième image est supérieure (label **snow_up**, notation : $\prec_s$), inférieure (label **snow_down**, notation : $\succ_s$), incomparable (label **eq3**, notation : $\perp_s$) ou équivalente (label **eq**, notation : $\sim_s$). Pour annoter les cas où l'épaisseur et l'étendue ne varient pas de la même manière, cas peu fréquent, nous utilisons le label supplémentaire **new_snow**. Les règles d'utilisation de ce dernier label sont complexes et n'ont pas d'intérêt propre. Nous ne les décrirons pas ici.
Pendant la deuxième étape d'annotation, l'étendue et l'épaisseur ont été annotés séparément. Les notations pour l'épaisseur sont $\prec_d$, $\succ_d$, $\perp_d$ et $\sim_d$.
Dans le premier paragraphe, nous donnons les définitions de l'étendue et les critères utilisés pour la comparer. Même chose pour l'épaisseur. Pour les deux paramètres, les images sans neige au sol sont toutes considérées comme équivalentes les unes aux autres.

Etendue du manteau : définition et critères de comparaison
Ce que nous appelons étendue du manteau correspond en première approximation au nombre de pixels couvrant des surfaces enneigées (pixels « enneigés »). En superposant deux images avec neige au sol, il est en général assez facile de dire si l'étendue est plus importante sur une image que sur une autre. En effet, la neige colonise souvent les différents types de surface suivant la même séquence [2].
Le **premier critère** pour l'étendue est donc :

2. Par des températures proche de 0°C, la neige tient et s'accumule d'abord sur les surfaces en herbe, les feuillages et les toitures, puis sur la terre nue, et ensuite sur les surfaces artificielles : zones piétonnes et bandes latérales puis bandes de roulement. Par des températures plus froides, l'enneigement peut avoir lieu sur toutes les surfaces en même temps.

Si, sur l'image b, la neige atteint un type de surface plus avancé dans la séquence classique que sur l'image a, alors $a \prec_s b$.
Dans un cas « d'égalité », on précise le premier critère par : si, sur le dernier type de surface colonisé, les pixels enneigés de l'image a sont aussi enneigés sur l'image b, mais que l'inverse n'est pas vrai, on décide $a \prec_s b$. Dans les autres cas, on choisit $a \sim_s b$ lorsque les manteaux sont d'extension identique et on choisit $a \perp_s b$:

— lorsque des pixels sont atteints par la neige sur a. et pas sur b et des pixels atteints par la neige sur b et pas sur a. Cela se produit par exemple lorsque la neige s'accumule préférentiellement d'un côté ou de l'autre de la route, sous l'effet du vent.
— lorsque sur l'une des images les surfaces sont masquées, soient à cause des véhicules, (cas d'embouteillages sur une chaussée enneigée), soit parce que la visibilité est trop faible (brouillard épais, chutes de neige intenses), soit parce que la neige s'est accumulée en bas de la lentille, ou sur la lentille (image de mauvaise qualité), etc.

Epaisseur du manteau : définition et critères de comparaison
La hauteur de neige est définie par l'épaisseur en centimètres de la couche de neige au sol. Les instruments et les protocoles de mesure varient [124] Mais on utilise en général une surface plane, faite dans un matériau isolant. Sur une scène routière, ce sont les surfaces planes en herbe ou en béton qui sont les mieux indiquées pour une estimation de l'épaisseur. Pendant l'annotation, ces surfaces sont priviligiées quand elles existent. A défaut, les autres surfaces sont utilisées.
Pour prendre une décision, deux critères sont exploités :

— un critère relatif aux textures : plus le manteau est épais, et plus la surface de la neige paraît uniforme. Ce critère est surtout utilisable pour des manteaux de faible épaisseur. Sur des surfaces naturelles, il doit être utilisé prudemment lorsque la comparaison est faite à plusieurs mois d'écart : en début de saison, la végétation peut être encore haute et la texture apparaît hétérogène alors que l'épaisseur peut-être importante.
— Pour les épaisseurs plus importantes, nous utilisons les surfaces verticales (marche des trottoirs, poteau de signalisation, etc) comme une jauge.

L'équivalence ($\sim_d$) est appliquée avec parcimonie, lorsque les deux manteaux semblent d'égale épaisseur. Dans le cas où les critères donnent des résultats contradictoires, on décide l'incomparabilité ($\perp_d$). Les autres cas d'incomparabilité sont les mêmes que pour l'étendue. Précisons un dernier élément : les épaisseurs ne sont pas jaugées dans les zones de remblais alimentées par les déneigeuses.

A.1.2.3 Représentation et nature de la relation d'incomparabilité

En appliquant les critères précédents à travers l'algorithme d'annotation semi-automatique présenté au chapitre 2, nous construisons des relations d'ordre partiel sur les images des séquences. Ces relations encodent les difficultés de l'annotateur sur certaines catégories d'image (les images prises à contre-jour ou bruitées, les précipitations ou les situations d'hétérogénéité spatiale marquée) ou devant deux images prises dans des conditions d'éclairage ou d'enneigement très différents.
Lorsque toutes les images d'une séquence ont été annotées, on peut distinguer ces deux sources d'incomparabilité (par image ou par paire) dans les graphes associés aux relations d'ordre résultantes. Seule la première source d'incomparabilité peut être « captée » par un ordre d' intervalle (voir chapitre 4).

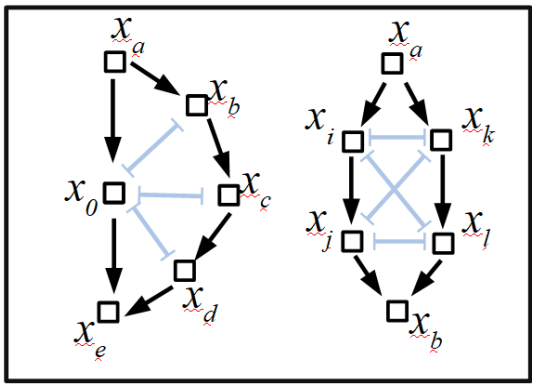


FIGURE A-10 – De l'application des critères résultent des relations d'ordre pouvant être représentées sous forme de graphes. Ces représentations permettent de différencier les causes d'incomparabilité qui sont intrinsèque à l'image de causes d'incomparabilité qui sont dues à la différence d'aspect entre les deux images. Par exemple, à gauche, l'image x_0 est plus souvent trouvée incomparable parce que sa qualité est affectée par des parasites. A droite, les incomparabilités sont liées à des différence d'aspect qui compliquent la comparaison des contrastes (par exemple lorsque l'éclairage global est d'intensité très différente, ou lorsque les surfaces sont enneigées sur l'une des deux images et pas sur l'autre).

Enfin, soulignons deux éléments importants. D'une part le niveau de prudence de l'annotateur, et donc sa propension a valider les critères de comparaison fluctue nécessairement dans le temps. Un choix entre comparaison stricte et incomparabilité est donc moins objectif qu'un choix entre les deux classes de relation stricte. Il faut en avoir conscience lorsque l'on choisit la méthode d'évaluation associée à un problème d'apprentissage.
Ajoutons que dans le cas de la visibilité, le label incomparable reflète souvent une limite cognitive : il est difficile de comparer deux contrastes. Ce n'est pas nécessairement une limite fondamentale. Autrement dit, un modèle peut a priori faire mieux qu'un humain sur la tâche de comparaison. C'est moins le cas pour l'étendue et l'épaisseur. Pour ces paramètres, la prédiction d'un ordre strict plutôt qu'une incomparabilité reflète plus certainement une erreur

d'interprétation. Cette remarque est à la source d'un changement de score (chapitre 5).

A.2 Annotation par sous-séquences (détails de la troisième étape d'annotation)

Les premiers modèles appris par paires sur le critères de l'étendue du manteau neigeux se sont montrés peu performants. Sur des séquences non annotées du jeu AMOSvvExt qui ont été rassemblées pour un apprentissage semi-supervisé d'un indice de visibilité (voir chapitre 4), les fausses détections et les non-détections sont nombreuses. Pour améliorer les performances des modèles, nous avons sélectionné des sous-séquences d'AMOSvvExt sur lesquelles ces premiers modèles prédisent une croissance ou une décroissance du manteau neigeux (A.2.1). Pour chacune de ces sous-séquences, on regarde comment évolue l'indice de visibilité. Trois groupes sont constitués : les sous-séquences où le manteau est prédit croissant et la visibilité est basse (groupe 1), celles où le manteau est prédit croissant et la visiblité est bonne (groupe 1 bis), celles où le manteau est prédit décroissant.
Les sous-séquences des groupes 1 et 2 sont rapidement parcourues à la main et annotées (A.2.1.1). Les paires d'images du groupe 1 bis sont automatiquement considérées comme incomparables.

A.2.1 Préparation des sous-séquences

Les 12.000 séquences du jeu AMOSvvExt sont parcourues par les modèles vv_sl_due111.0 et ss_sl_due000.0 (voir Annexe C, nomenclature des modèles). On définit les période de basses visibilité par une borne supérieure prédite inférieure au 30ème centile des bornes inférerieures et une période de bonne visibilité comme une borne inférieure prédite supérieure au 35ème centile des bornes inférieures. Pour le groupe 1 (resp. groupe 1 bis), on sélectionne toutes les sous-séquences d'images s'étalant sur deux heures telles que :
-ces images sont comprises dans des périodes de basse visibilité (resp. bonne visibilité)
-les deux premières images sont prédites comme étant moins enneigées que les deux dernières par ss_sl_due000.0.
Pour le groupe 2, la fenêtre est allongée à six heures pour tenir compte du fait que la fonte est généralement plus lente que la croissance du manteau. On sélectionne ainsi toutes les sous-séquences s'étalant sur six heures telles que les trois premières images sont prédites comme étant plus enneigées que les trois dernières par ss_sl_due000.0. Le nombre de sous-séquences obtenues est indiqué en première colonne de la table A-1.a.

groupe	seq.	sous-seq.	images
groupe 1 : croissance détectée	1139	1629	8831
gr. 1 bis : fausse croissance	628	1153	5811
groupe 2 : fonte	1095	2220	26209

a. Sous-séquences sélectionnées parmi les séquences d'AMOSvvExt

> (0)	< (1)	Eq (2,4,6,8)				Incomp (3,5,9)			Ch (7)	Mixt (10)
		mas.	surexp.	sans neige	tot.	mas.	surexp.	tot.		
14	944	21	97	110	213	64	30	95	35	6

b. annotations des sous-séquences du groupe 1

> (0)	< (1)	Eq (2,4,6,8,12,14)						Incomp (3,5,9,13,15)					Ch (7)	Mixt (10)	No ground
		mas.	surexp.	sans neige	reflets	soir	tot.	mas.	surexp.	soir	reflets	tot.			
940	3	27	199	164	39	140	348	19	125	26	4	101	37	8	40

c. annotations des sous-séquences du groupe 2

TABLE A-1 – Résultats de la troisième étape d'annotation

A.2.1.1 Règles d'annotation des sous-séquences

Pour chaque sous-séquence parcourue des groupes 1 et 2, on note s'il y a croissance ou décroissance de l'étendue du manteau. 42 % des épisodes de croissance du groupe 1 sont des fausses détections. Dans quelques cas, la neige a plutôt fondu, ou il y a eu croissance et décroissance (label 10, colonne Mixt., table A-1). Dans le reste, le manteau n'a pas évolué. On distingue alors le lien entre les images (incomparabilité ou équivalence) et les causes présumées de l'erreur par un code chiffré. Les labels associés à une « équivalence » (colonnes « Eq », tables A-1.b-c) ne sont portés que lorsque que toutes les paires d'images consécutives peuvent effectivement être considérées comme équivalentes. Plusieurs causes d'erreurs sont récurrentes : le réseau est leurré par des masques sur la lentille (labels 6 ou 5), par des cas de surexposition (8 ou 9) ou de reflets de la lumière du soleil sur des surfaces plates. Dans 7 % des cas, il n'y a pas de neige au sol (label 4).
Le nombre de fausses détections dans le groupe 2 est très élevé (58%). L'éclairage est là encore responsable de ces fluctuations, en particulier à cause des reflets du soleil qui s'estompent progressivement (labels 12,13), un ombrage qui augmente, la luminosité qui baisse en soirée (labels 14,15), expliquent une bonne part des erreurs.

A.2.1.2 Conversion en comparaisons par paires

Des sous-séquences « croissantes » trouvées, seules les paires formées à partir d'une des deux dernières images et d'une des deux premières images sont stockées dans le graphe $\mathcal{D}_1^s$ des relations d'ordre strictes. Des sous-séquences décroissantes, plus longues on n'ajoute dans $\mathcal{D}_1^s$ que les paires formées d'une des trois premières images et d'une des

trois dernières images. Ces précautions assurent que l'évolution du manteau est bien visible sur les paires qu'on a conservées.

Dans les sous-séquences annotées « incomparabilité » (resp. « équivalence »), on forme toutes les paires possibles et on les stocke dans $\mathcal{U}_1^s$ (resp . $\mathcal{E}_1^s$). Les tailles de ces graphes [3] sont précisées sur la table A-2.

caméras	images	arêtes de $\mathcal{D}_1^s$	arêtes de $\mathcal{U}_1^s$	arêtes de $\mathcal{E}_1^s$
1178	30.181	12.240	13.320	37.090

TABLE A-2 – Nombre de nouvelles relations (comparaisons strictes, incomparabilités et équivalences)

3. Avant l'ajout des produits de la deuxième étape d'annotation.

Annexe B

Description détaillée des principaux jeux d'apprentissage

Dans cette section, nous donnons un aperçu de la diversité des séquences sélectionnées pour l'annotation manuelle, automatique et instrumentale.
Les premiers tableaux sont relatifs aux 25.000 images annotées à la main (étape 1 et 2), ou de manière automatique (approche semi-supervisée). Le tableau associé à l'annotation automatique donne une bonne idée de ce que contient la base AMOS en terme de scènes.
Les tableaux qui suivent sont relatifs aux images du réseau TENEBRE : on y décrit la base TENEBRE et le détail des résultats de la comparaison entre annotation à la main et annotation instrumentale.

B.1 Description des images annotées à la main.

Dans ces tableaux, la partition train/val/test correspond à celles de AMOSss et AMOSsd. La répartition de AMOSvv est similaire, mais deux jeux de validation sont utilisés (voir chapitre 3). Les lignes bleues démarquent les deux jeux utilisés au chapitre 3 section 1 pour apprendre sur des tâches de classification. La table A-1 est un récapitulatif. La table A-2 donne le détail de la répartition pour les attributs relatifs à l'état du sol. La table A-3 donne la répartition pour les attributs relatifs à l'état de l'atmosphère. La table A-4 détaille la répartition en terme de types de scènes et la table A-5 celles des attributs relatifs à la qualité des images.

source	set	Sites internet	clips	séquences	images	Images jour+inter	onsets	Images Neige au sol	Images Précip.	Images défaut
AMOS	Train	103	693	308	9038	5378	181	6.912	5.505	1993
	Val	23	56	53	1773	1065		1256	735	370
	Test	28	28	28	1664	965		1206	768	311
	Total	154	777	389	12475	7408		5374	7008	2674
DIRs	Train	52	52	52	6689	4317	47	3329	5472	967
	Val	10	10	10	461	270		422	248	63
	Test	36	36	36	1640	975		1479	770	189
	Total	98	98	98	8790	5501		5155	6442	1219
Infoclimat (test)	Test	102	102	102	1285	1124		962	534	285
Total	Train	155	745	359	15727	9695		10241	10977	2960
	Val	33	66	63	2234	1335		1678	983	433
	Test	166	166	166	4589	3064		3647	2072	785
	total	354	875	589	22550	14094		15566	14032	4178
TENEBRE_IH	Eval. labels	8	11	11	2505	1164		1352	776	706

TABLE A-1 – Récapitulatif de la répartition des images par attribut et par jeu. Dans la colonne « clips », nous indiquons le nombre de changement de scènes total dans les séries d'image d'origine. Un travail de racordement en partie automatisé (non présenté dans ce manuscrit) a permis de rassembler les clips associés à une même caméra. Ainsi, les 777 clips extraits d'AMOS sont ramenés à 389 séquences, chacune associée à une caméra différente -voire exceptionnellement, à deux caméras très proches.
Le nombre d'onsets, qui n'a été déterminé que pour AMOS, correspond au nombre de « débuts observés » (voir chapitre 5) dans le jeu. Les images avec défaut correspondent à l'ensemble des images comptées dans la table A-5.

source	set	Etat du sol									
		no_snow			no_snow - _road	snow_ground			snow_ road	white_ road	doubt
		dry_ road	wet_road	snow traces (ground)		wet_ road	dry_ road	snow_tr (road)			
AMOS	Train	575	1314	119	91	3293	136	344	2773	710	146
	Val	161	292	53	31	626	11	72	440	179	33
	Test	142	461	22	4	618	8	23	461	119	18
	Total	878	2067	194	126	4537	155	439	3674	1008	197
DIRs	Train	386	520	137	130	2987	114	467	2187	184	181
	Val	6	33	6	0	281	5	2	95	41	0
	Test	1	150	71	6	964	5	17	454	56	4
	Total	393	703	214	136	4232	124	486	2736	281	185
Infoclimat (test)	Test	19	278	15	0	505	2	3	231	224	26
Totaux	Train	961	1834	256	221	6280	250	811	4960	894	327
	Val	167	325	59	31	907	16	74	535	220	33
	Test	162	889	108	10	2087	15	43	1146	399	48
	Total	1290	3048	423	262	9274	281	928	6641	1513	408
TENEBRE_IH	Eval. labels	162	423	40	44	776	3	24	290	283	524

TABLE A-2 – Détail par label de la répartition des niveaux d'enneigement. Les labels sont décrits dans l'annexe A. Le colonne snow_ground / dry_road compte les images annotée **snow_ground_dry_road**. La colonne snow_ground / wet_road compte toutes les autres images avec neige sur la route n'étant ni à l'état de traces ni entièrement couvrante.

sources	set	Etat de l'atmosphère									
		No precip	doubt	Precipitations (fog incl.)							
				Precip. (indif)	rain	snow				fog	fog_or_snow
						total	streaks	Streaks night	doubt		
AMOS	Train	1176	2357	1019	251	4131	776	537	350	83	21
	Val	222	816	106	22	604	130	50	60	3	0
	Test	212	684	80	2	677	135	96	58	9	0
	Total	1410	3857	1205	275	5412	1041	683	1094	95	21
DIRs	Train	1262	2098	90	0	3110	1829	369	361	123	6
	Val	55	158	1	0	208	49	75	5	28	11
	Test	155	715	24	0	653	129	177	4	84	9
	Total	1472	2971	115	0	3971	2007	621	370	235	26
Infoclimat (test)	Test	277	474	30	3	501	275	67	12	0	0
Total	Train	2438	4455	1109	251	7241	2605	906	711	206	27
	Val	277	974	107	22	812	179	125	65	31	11
	Test	644	1873	134	5	1831	539	340	74	93	9
	Total	3359	7302	1350	278	9884	3323	1371	850	330	47
TENEBRE_IH	Eval.. labels	448	1281	228	4	381	28	2	45	62	101

TABLE A-3 – Répartition des images pour les attributs relatifs à l'état de l'atmosphère. Les labels sont décrits dans l'annexe A. Les deux colonnes « doubt » correspondent l'une à l'occurrence de précipitations, l'autre à la présence dans l'image de traînées dues à des flocons en chute.

source	set	Type de scène				Eclairage			
		autoroute	Route campagne	urbain	spécial	jour	inter	Nuit claire	Nuit sombre
AMOS (hand.)	Train	476	25	37	11	4792	586	3591	69
	Val	66	9	4	3	940	125	704	4
	Test	12	8	4	4	857	108	673	26
	Total					6589			
DIRs (hand.)	Train	28	20	1	3	3766	551	1571	801
	Val	2	5	0	2	209	61	183	8
	Test	4	24	1	6	754	221	638	27
	Total					4729			
Infoclimat (hand.)	Test	0	7	5	90 dont : 72 aérodromes	1008	116	148	13
Totaux	Train	504	45	38	14	8558	1137	5162	870
	Val	68	13	4	5	1149	186	887	12
	Test	16	38	9	25/71	2620	445	1459	66
	Total	588	96	51	115	12327	1768	7508	948
TENEBRE_IH	Eval. labels	2	3	0	6	1014	150	1020	321

TABLE A-4 – Répartition des images en terme de type de scène.

source	set	Masques et bruits									
		droplets	droplets_acc	snowflakes	snowflakes_acc	blurry	filth	artefacts	surexp	box/rec	other
AMOS	Train	941	116	410	82	332	197	13	34	39	8
	Val	140	16	81	15	49	33	67	18	1	12
	Test	112	10	67	23	42	20	42	16	0	16
	Total	1193	142	558	120	423	250	122	68	40	36
DIRs	Train	106	12	462	153	173	23	119	84	28	121
	Val	6	0	5	4	4	0	2	22	25	0
	Test	0	0	45	5	33	4	16	77	18	1
	Total	112	12	512	162	210	27	237	193	71	122
Infoclimat (test)	Test	77	3	79	20	25	68	13	11	15	14
Total	Train	1047	128	872	235	505	220	132	118	67	129
	Val	146	16	86	19	53	33	69	40	26	12
	Test	189	13	193	48	100	92	71	104	33	31
	Total	1389	157	1151	302	658	345	272	262	126	172
TENEBRE_IH	Eval. labels	83	6	55	19	28	263	60	187	2	53

TABLE A-5 – Répartition par attribut relatif à la qualité de l'image. Les labels sont décrits dans l'annexe A.

B.2 Description des séquences d'AMOSssExt.

Dans la table A-6, nous précisions la répartition des types de scène dans AMOSssExt, jeu présenté au chapitre 5. Par exemple, sur l'ensemble des séquences (en comptant celles d'AMOSss0.0, pour 93 séquences indépendantes, seule la route est visible dans le champ de la caméra (scènes autoroutières). Sur les scènes urbaines, la neige est souvent visible sur les toitures avant d'être présente au sol. Lorsque la montagne est visible au loin (1521 séquences), la neige est visible sur presque une moitié de l'année, à grande distance, sur les sommets.

La répartition dans AMOSvvExt (voir chapitre 4), qui comprend à peu près les mêmes images, est très proche. Elle n'est pas détaillée dans ce manuscrit.

Statistiques sur 3.699 séquences d' AMOSssExt (jeu d'entraînement)		Type de scène								
		route	ville	camp.	mont.	eau	spécial	plongée	sol abs.	large
Proportion dans l'image	only	93	118	194	85	2	62	523	27	321
	mainly	2463	527	1153	219	167				
	some	3618	1701	3152	1521	272				

TABLE A-6 – Répartition par types de scène (jeu AMOSssExt). Les colonnes 2 à 6 sont relatives à la nature des objets présents dans la scène. Campagne (camp.) : pré, champs, forêts de plaine. Plans d'eau (eau.) : il peut s'agir de caméras côtières (only), ou de scènes donnant sur la mer, un lac ou une rivière. Montagne (mont.) : roche affleurant, falaises, reliefs proches (« mainly/only ») ou lointains (« some »). Les séquences spéciales comptent des environnements très particuliers (cataractes, coeur urbain, chantier, base internationale en antarctique, etc). Les trois dernières colonnes décrivent le type de plan. Sol absent (sol abs) : seuls des immeubles, le feuillage des arbres sont visibles. Plan large (large) : cas où la caméra, située en altitude, offre une vue d'ensemble sur une plaine ou une vallée.

B.3 Description du jeu TENEBRE.

Dans cette section, nous donnons d'abord des précisions sur les instruments utilisées pour l'annotation instrumentale des archives TENEBRE (table A-7). Ensuite, nous donnons une description du jeu TENEBRE_1218 (table A-8). Les deux dernières tables (A-9 et A-10) donnent un complément d'information sur la concordance entre annotation manuelle et annotation instrumentale (cas d'une annotation relative à l'image). Elles ont été réalisées de

façon à estimer le taux d'erreur d'annotation commises (de l'ordre de 1-2 %) et à le comparer à la proportion de mesures non représentatives de la scène (de l'ordre de 4% à 8% pour la présence de neige).
Les mesures RR ne seront pas utilisées dans la suite de la thèse parce qu'elles sont très peu représentatives de l'image (table A-10).
Des études analogues ont été réalisées pour l'annotation par paire. Elles n'ont pas été mises en forme, mais elles sont synthétisées à la fin du chapitre 2.

Paramètre		hauteur d'eau (RR)	Visibilité (VV)	Épaisseur (SH)
Pas de temps		1 min et 6 min	1 min	1 h
station/capteur	Nancy-Essey (54)	PWD22	DF320	SHM30
	Strasbourg-Entzheim (67)	PWD22	DF320	SHM30
	Dorans (90)	PWD22	PWD22	SHM30
	Roissy (95)	PWD22	DF320	SHM30

TABLE A-7 – Instruments utilisés pour l'annotation automatique du jeu TENEBRE_1218.

Caméra (station)	station	Séries	Images (par série)	Périodes de neige au sol	SH>0 RR=0	SH>0 RR>0	SH=0 RR>0 T<-2	SH=0 RR>0 T>2	SH=0 VV<1000
dorans	Dorans	dorans1	51083	2	1284	132	2	1350	1550
entzheim	Strasbourg-Entzheim	entzheim1 entzheim3	51281	3	500	40	0	670	1180
entzheim_parc		entzheim_parc	52120	4	500	40	0	670	1180
entzheim_portail		entzheim_ portail	52185	4	500	40	0	670	1180
nancy	Nancy-Essey	nancy1 nancy2	51755	5	4350	210	9	1870	980
nancy_neige		nancy_neige (rupture)	46484	5	4150	200	3	1600	870
roissy	Roissy	roissy (rupture)	48398	3	3750	260	0	1620	780
Total		10	350 k	30	16 k	1 k	14	9,6 k	10 k

TABLE A-8 – Description du jeu TENEBRE_1218.

label	sh == 0	sh > 0
wet_road, dry_road	475	12
Neige au sol (tout confondu)	119	1104
snow_ground, snow_ground_dry_road	91	639
snow_road	15	245
white_road	13 pannes Markstein	220
snow_traces	28	3 (parmi les 12 au-dessus)
Labels **doubt , no_snow_road**	330	64

Détails :
-12 cas de neige pas vue mais mesurée (à des valeurs toutes inférieures à 1 cm):
7 traces dont 3 signalées avec ground_traces, 4 difficiles à voir.
3 erreurs de labellisation (2 évidentes, une erreur de prudence)
2 images sur lesquelles la neige n'est pas visible
Sur les 119 cas de neige vue mais pas mesurée, on compte :
-13 discordances en cas de « white_road »
toutes sur la même caméra (Markstein). Mesures non représentatives.
-15 discordances en cas de « snow_road »
9 erreurs de l'instrument sûres, quatre erreurs d'annotation commises sur scènes nocturnes.
-91 discordances en cas de « snow_ground »
Sur 70 images, la neige est apparente, dont 33 sur lesquelles elle n'est visible que par endroits, 17 cas de manteaux neigeux ancien.
Sur 2 d'entre elles, il s'agissait certainement d'erreur d'annotation. Sur 19 autres, de potentielles erreurs, dont 4 éventuelles gelées blanches.

TABLE A-9 – Comparaison entre annotation à la main et mesures sur la détection de neige au sol. La première colonne comptabilise les images associées à une mesure de hauteur de neige (sh) nulle. L'étude porte sur huit des dix séquences de TENEBRE_IH.

Label	$RR_{1mn} == 0$	$RR_{1mn} > 0$	$RR_{6mn} == 0$	$RR_{6mn} > 0$
« snow »	359	8	338	37
Précipitations (snow, precip, rain)	573	11	545	48
« doubt »	1164	14	1132	55
« no_precip »	424	1	420	4
« streaks »	26	4	24	6
« Snowflakes » or « droplets »	154	5	140	22

Détails :
-Nombreuses non-détections instrumentales.
Dans les cas où des flocons en chute libre sont observés on compte une erreur d'annotation (oiseaux :nancy1_2013-03-14_16_06). Les autres annotations sont correctes (les chutes de neige ont échappé au pluviomètre). Cela s'améliore un peu en agrégeant sur 6 minutes (RR_{6min}).
-Rares scènes où la pluie est détectée sans qu'on la voie.
Ce sont des cas d'annotation « imprudente » pour les 5 cas observés (les images auraient dû êtres classées « **doubt** ».

TABLE A-10 – Comparaison entre annotation à la main et mesures pour la détection des précipitations. La première (resp.troisième) colonne comptabilise les images associées à un cumul nul au pas de temps 1 minute (resp 6 minutes). L'étude porte sur huit des dix séquences de TENEBRE_IH.

Annexe C

Nomenclature des modèles

C.1 Nomenclature

La nomenclature des modèles fournit (en général) des indications sur le paramètre appris. Le mode de prédiction apparaît presque systématiquement dans le nom utilisé dans le manuscrit (cl pour la classification, pl pour la prédiction par paires et sl -siamese learning- pour la prédiction par image). Dans le cas d'un apprentissage des préférences, on indique les graphes sur lesquels les modèles ont été entraînés (d -directed- pour les comparaisons strictes, u -undirected- pour les incomparabilités, et e pour les équivalences). Les chiffres qui suivent indiquent la version du graphe utilisée. Par exemple vv_pl_vgg13_du00.0 (table A-2) est le premier VGG13 entraîné à la prédiction par paire sur le paramètre visibilité sur les graphes $\mathcal{D}_0^v$ et $\mathcal{U}_0^v$.

Les modèles des tables 1-7 correspondent à 5-10 % des modèles entraînés pendant la thèse. Ce sont ceux qui ont présenté les meilleurs résultats en validation pour un type d'expérience donnée.
Lorsque deux paramètres apparaissent au début du nom (e.g. vvss_pl_vgg13_du00.3, table A-2), c'est que le modèle a été entraîné en multi-tâche (multi-paramètre) sur un jeu spécifique (voir l'annexe D.3). Dans certains cas, l'apprentissge fait porter sur une tâche de nature accessoire plutôt que sur un paramètre accessoire. Nous avons en effet entraîné des fonctions de rang avec une tâche de classification accessoire. Par exemple, ss_slcl_due000.0 (table A-5) est entraîné à la fois sur une tâche de prédiction des préférences et sur une tâche de classification neige au sol / absence de neige au sol accessoire (voir annexe D.3).

C.2 Modèles mis à disposition

Nom dans le répertoire models/classif	Nom dans la thèse
ResNet50_ground5_0_bm	ss_cl_dn.0
ResNet50_ground5_1_bm	ss_cl_d.0
ResNet50_ground5_2_bm	ss_cl_d.1

TABLE A-1 – Modèles entraînés à la classification décrits section 5.4.

nom dans le répertoire *models*/vv/par_paire	nom dans la thèse
sd_without2__140920_relative_vv__day_vgg16_scratch_300	vgg16_2classes
sd_with2_crop75_step20__140920_relative_vv__day_vgg16_scratch_300	vgg16_3classes
sd_with2_021020_relative_vv__day_vgg13_scratch_Adam_sizein256_300_bm	vv_pl_vgg13_du00.0
0401_preference_learning_vvss__day_vgg13_scratch_mtl_Adam_600_bm	vvss_pl_vgg13_du00.3
sd_with2_200920_relative_vv__day_vgg11_bn_scratch_Adam_300	vv_pl_vgg11_du00.0

TABLE A-2 – Meilleurs modèles entraînés à la prédiction par paire sur AMOSvv (voir section 3.3).

nom dans le répertoire *models*/vv/par_image/	nom dans la thèse
0411_not_registred_we_day_vv_sigmay__resnet50_imagenet25_bm	vv_sl_d0.0
1411_siamese_new_daug_resnet50_imagenet30_bm	vv_sl_d0.1
1511_image_wise_day_vv_sigmay_dgeg_resnet50_imagenet_mtl50_bm	vv_sl_de00.0
1311_not_registred_we_day_vv_sigmay_new_daug_semi_supervised_resnet50_imagenet25_bm	vv_sl_d1.0
1411_siamese_newdaug_daug_semi_supervised_resnet50_imagenet30	vv_sl_d1.1
1511_image_wise_day_vv_sigmay_dgeg_semi_supervised_resnet50_imagenet_mtl25_bm	vv_sl_de11.0
2711_image_wise_day_vv_sigmay_dgugeg_ranknet_semisupervised_resnet50_imagenet_mtl40_bm	vv_sl_due111.0
1412_image_wise_day_vv_dgugeg_daug3_ranknet2_semisupervised_resnet50_imagenet_mtl40_bm	vv_sl_due111.1
1612_image_wise_day_vv_dgugeg_daug3_ranknet2_semisupervised_resnet50_imagenet_mtl40_bm	vv_sl_due111.2
1612_image_wise_day_vv_dgugeg_daug3_ranknet2_semisupervised_resnet50_imagenet_mtl20_bm	vv_sl_due111.3
2012_image_wise_day_vv_dgugeg_daug3_ranknet2_semisupervised_ug0_dg1_eg1__resnet50_imagenet_mtl20_bm	vv_sl_due101.0

TABLE A-3 – Meilleurs modèles entraînés à la prédiction par image sur AMOSvv et AMOSvvExt (voir sections 3.3, 4.1 et 4.2).

nom dans le répertoire *models*/ss/par_paire	nom dans la thèse
241220_preference_learning_dgug_shsurface__day_vgg16_scratch_Adam_600_bm	ss_pl_du00.0
1101_preference_learning_vvss__day_vgg13_scratch_mtl_Adam_600_bm	ssvv_pl_du00.0
150321_preference_learning_dgug_ext_shsurface__day_vgg13_scratch_Adam_1100_bm	ss_pl_du11.0
150321_preference_learning_dgug_ext_strongerdag_shsurface__day_vgg13_scratch_Adam_1100_bm	ss_pl_du11.1

TABLE A-4 – Meilleurs modèles entraînés à la prédiction par paire sur AMOSss.0 et AMOSss.1 (voir section 5.2).

nom dans le répertoire *models*/ss/par_image	nom dans la thèse
2212_image_wise_day_shsurface_dgugeg_daug3_ranknet_resnet50_imagenet_mtl25_bm	ss_sl_due000.0
2912_image_wise_day_ss_dgugeg_roof_ranknet_resnet101_imagenet_mtl100_bm	ss_slcl_due000.0
1703_image_wise_day_ss_dgugeg_ranknet_ext_pierre2_resnet50_imagenet_mtl20_p_ug0.25_p_eg0.25_bm	ss_sl_due111.0
1703_image_wise_day_ss_dgugeg_ranknet_ext_daynight2_resnet50_imagenet_mtl10_p_ug0.25_p_eg0.25_bm	ss_sl_due111.1
1706_image_wise_day_semi_supervised_daugsnow_stratified_reverse_resnet50_imagenet_mtl100_p_ug20_p_eg20_bm	ss_sl_rev_due222.0
1706_image_wise_day_semi_supervised_daugsnow_stratified_reverse_resnet50_imagenet_mtl100_p_ug20_p_eg20	ss_sl_rev_due222.1
v2_image_wise_day_semi_supervised_daugsnow_stratified_ref0_resnet50_imagenet_mtl60_p_ug20_p_eg20_bm	ss_sl_due222.0 (zay)
v2_image_wise_day_semi_supervised_daugsnow_stratified_noeg1_resnet50_imagenet_mtl60_p_ug40_p_eg0_bm	ss_sl_du22.0 (zay)
v2_image_wise_day_semi_supervised_daugsnow_stratified_newweighting4_resnet50_imagenet_mtl60_p_ug20_p_eg20_bm	ss_sl_due222.1 (zay)
070921_image_wise_day_semi_supervised_daugsnow_dg0_resnet50_imagenet_mtl60_p_ug0_p_eg0_bm	ss_sl_d2.0
v2_image_wise_day_semi_supervised_daugsnow_stratified_noug2_resnet50_imagenet_mtl60_p_ug0_p_eg40	ss_sl_de22.0

TABLE A-5 – Meilleurs modèles entraînés à la prédiction par image sur AMOSss.0, AMOSss.1 et AMOSssExt (voir sections 5.2 et 5.3).

nom dans le répertoire *models*/sh/par_paire	nom dans la thèse
241220_preference_learning_dgug_shheight__day_vgg16_scratch_Adam_600	sd_pl_du00.0
3012_preference_learning_vvsssh__day_vgg13_scratch_mtl_Adam_200_bm	sdssvv_pl_du00.0
310721_preference_learning_dgug_shheight__day_vgg16_scratch_Adam_1500_bm	sd_pl_du11.1

TABLE A-6 – Meilleurs modèles entraînés à la prédiction par paire sur AMOSsh.0, (voir section 5.2).

nom dans le répertoire *models*/sh/par_image	nom dans la thèse
2212_image_wise_day_shheight_dgugeg_daug3_ranknet_resnet50_imagenet_mtl25_0_bm	sd_sl_due000.0
2212_image_wise_day_shheight_dgugeg_daug3_ranknet_resnet50_imagenet_mtl25_1_bm	sd_sl_due000.1
2412_image_wise_day_sh_dgugeg_roof_ranknet_resnet50_imagenet_mtl30_0_bm	sd_sl_cl_due000.0
2412_image_wise_day_sh_dgugeg_roof_ranknet_resnet50_imagenet_mtl25_bm	sd_sl_cl_due000.1

TABLE A-7 – Meilleurs modèles entraînés à la prédiction par par image sur AMOSsh.0, (voir section 5.2).

Annexe D

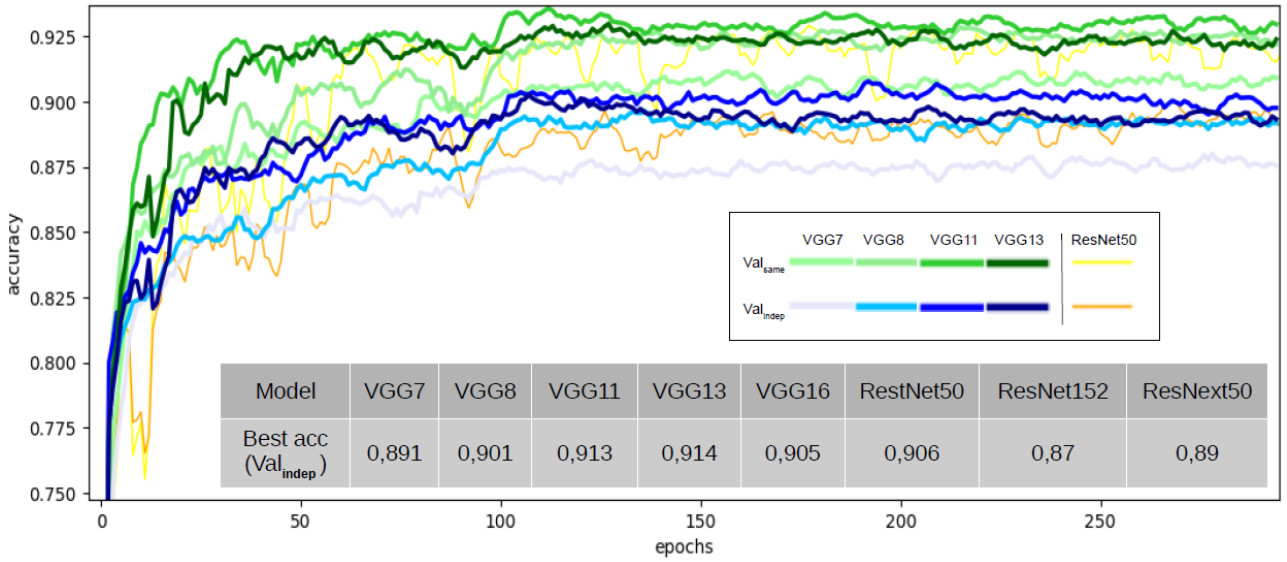
Compléments sur les chapitres 3 et 4

D.1 Choix des modèles

Pour les comparaisons de la section 3.3, nous avons sélectionné les hyperparamètres sur le jeu de validation (VAL_{indep}). Le critère utilisé était la justesse sur le problème à deux classes ($\{\dot{\prec};\dot{\succ}\}$).
Dans cette section, nous nous concentrons sur deux choix importants : la catégorie d'architecture et la taille du modèle. Ces choix sont présentés pour les deux types de tâche envisagés : la prédiction par paire et la prédiction par image.

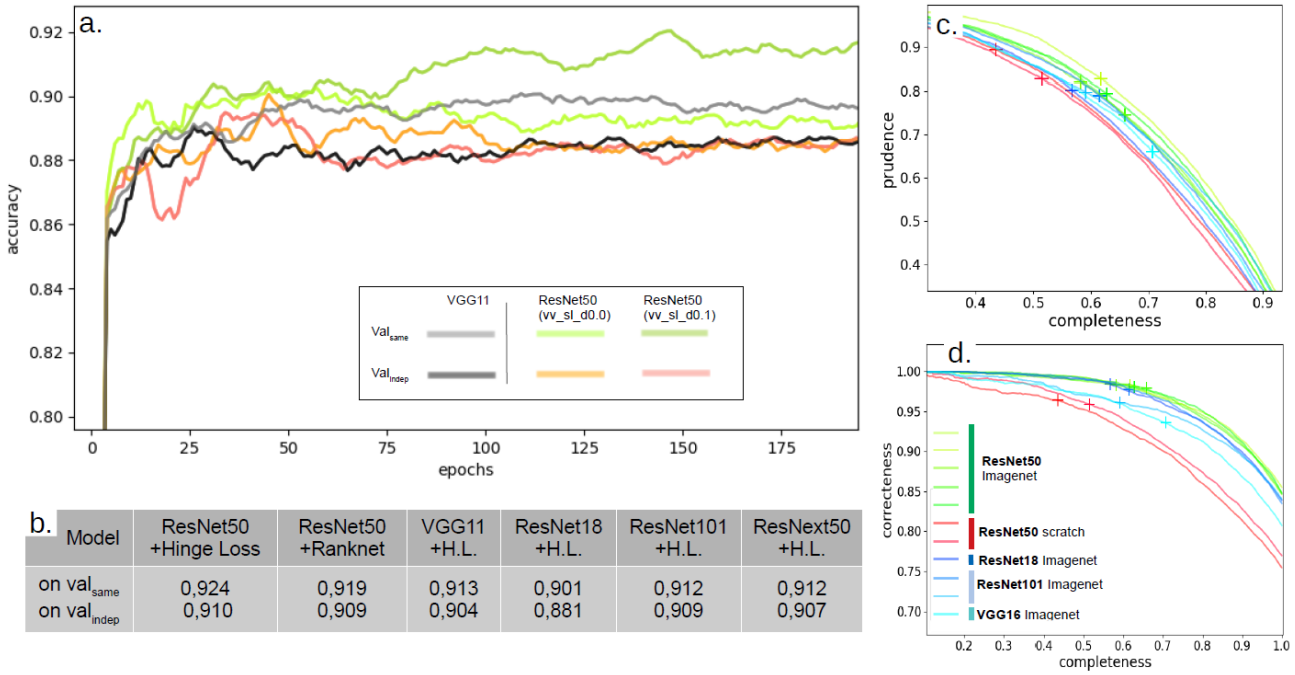
Pour la prédiction par paires, les modèles n'ont pas été pré-entraînés. Les architectures ResNet sont moins performantes que les architectures VGG en validation. Le meilleur ResNet obtenu est à 0.7 point sous les VGG11 et VGG13 sur des caméras indépendantes (voir figure A-1).
Le VGG13 a été sélectionné malgré des performances apparemment plus faibles (effet du lissage) que ceux du VGG11. En tout cas, l'effet d'une augmentation ou d'une baisse importante du nombre de couches de convolution a un effet négatif clair sur les performances en généralisation.

Pour la prédiction par image, il fallait non seulement choisir l'architecture et la taille mais aussi la fonction de coût (Ranknet Loss ou Hinge Loss). Sur le jeu de validation , les performances des modèles de taille moyenne étaient très proches, et toutes atteintes avant 100 époques (figure A-2.a-b). Suivie à la lettre, la procédure de sélection a conduit à chosir un ResNet50 avec Hinge Loss plutôt que des modèles VGG.
Ce choix a par la suite été confirmé par une comparaison sur le jeu de test d'AMOSvv (figure A-2.c-d) de modèles ayant été entraînés sur AMOSvvExt. Les modèles ResNet50 présentent les meilleurs performances, en comparaison aux VGG (le meilleur score est atteint avec un VGG16) et aux autres architectures de sa catégorie.



Model	VGG7	VGG8	VGG11	VGG13	VGG16	RestNet50	ResNet152	ResNext50
Best acc (Val_{indep})	0,891	0,901	0,913	0,914	0,905	0,906	0,87	0,89

FIGURE A-1 – Apprentissage des préférences sur AMOSvv. Performances en validation de la prédiction par paire. Les modèles VGG7 et VGG8 contiennent respectivement 4 et 5 couches de convolution. Les courbes ont été lissées par moyenne mobile (largeur : 5 époques).



b.

Model	ResNet50 +Hinge Loss	ResNet50 +Ranknet	VGG11 +H.L.	ResNet18 +H.L.	ResNet101 +H.L.	ResNext50 +H.L.
on val_{same}	0,924	0,919	0,913	0,901	0,912	0,912
on val_{indep}	0,910	0,909	0,904	0,881	0,909	0,907

FIGURE A-2 – Apprentissage des préférences sur MAMOSvv. Performances en validation de la prédiction par image. a. Courbes d'apprentissage (après lissage) sur AMOSvv. b.meilleures performances sur le jeu de validation. c-d. Séparation des différentes architectures sur le jeu de test, après apprentissage sur AMOSvvExt.

D.2 Construction d'AMOSvvExt

Dans cette section, nous présentons la méthode de construction d'AMOSvvExt. L'objectif est d'améliorer les fonctions de rang en étendant les jeux de données par annotation automatique. Cette annotation automatique est fourni par un classifieur entraîné à la prédiction par paire, dont les performances en généralisations sont meilleures que celles des fonctions de rang initiales.
Pour le faire, nous prenons le parti de reconstruire une relation d'ordre partiel (un ordre d'intervalle) sur les séries de données AMOS qui ont été mises de côté exprès (voir chapitre 2).
Plusieurs problèmes technique se posaient. Il fallait en particulier construire un jeu de séquences d'images de jour venant de caméras identiques (section D.2.1). Puis, pour éviter la redondance, nous avons cherché les épisodes de mauvais temps pour y concentrer la sélection des arêtes (section D.2.2).
Dans cette section, nous noterons $>_c$, $<_c$ et $\perp_c$ les comparaisons prédites par le classifieur vv_pl_vgg13_du00.0 et $C(.,.;\omega_c)$ la forward function associée.

D.2.1 Extraction des séquences

Les séquences webcam du jeu de données supplémentaire sont toutes tirées des archives AMOS. Dans ces archives, chaque répertoire contient une séquence d'images d'extérieur prise par une ou plusieurs webcams et publiées sur un site internet. Ces séquences sont longues de quelques mois à plusieurs années, avec un pas de temps de l'ordre de 30 minutes. Nous en avons extrait 6.295 dans des répertoires différents de ceux utilisés pour constituer les jeux validation et de test.
Contrairement au jeu AMOSvv, les scènes n'ont pas fait l'objet d'un tri ; néanmoins, il s'agit toujours, en majorité, de scènes routières.
Pour extraire les images prises de jour, un classifieur (VGG11) est entraîné sur AMOSvv à distinguer jour et nuit. L'utilisation d'un classifieur permet d'éviter les effets des erreurs de timestamp (fréquentes) et ceux des incertitudes sur la localisation géographique.
Un autre problème tient au fait qu'un même dossier AMOS peut contenir les images de plusieurs caméras différentes. Un changement de caméra peut être difficile à reconnaître. En effet, aucune information sur la caméra de provenance n'est disponible. De plus, l'alternance a souvent lieu pendant les épisodes de mauvais temps, pendant lesquels l'apparence de la scène change sensiblement.
Pour détecter ces changements de scènes, nous avons, là-encore, exploité un réseau de neurones entraîné sur AMOSvv sur une tâche d'identification (« même caméra »/« caméra différente ») en prenant en entrée la paire d'images concaténées (cf. figure A-3). Ce réseau permet de découper une série temporelle d'images de jour en une suite de séquences homogènes (dont les images proviennent de la même caméra). Nous n'avons conservé que les

séquences contenant plus de 20 images.
Après ces deux étapes, nous disposions de 12.241 séquences.

D.2.2 Tri automatique des sous-séquences

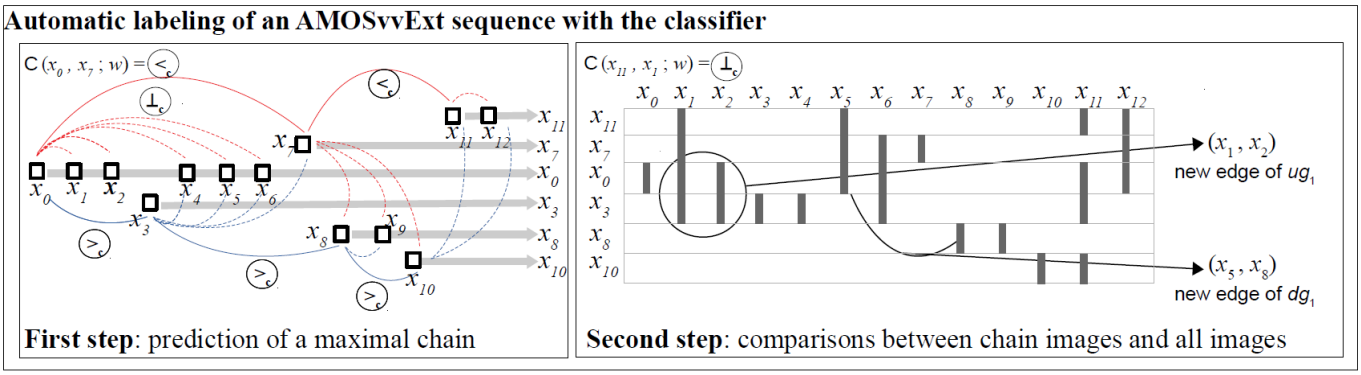


FIGURE A-3 – Construction d'un ordre d'intervalle sur les séquences d'AMOSvvExt (voir section D.2.2). Les arcs rouges correspondent aux comparaisons faites à la première passe, les bleus, à la seconde. Les incomparabilités sont représentées par des arcs en pointillé. Sur le graphique de droite, on ne représente que les relations d'incomparabilité (segments verticaux gris).

La même procédure d'annotation est ensuite appliquée à chacune des 12.241 séquences. Cette procédure d'annotation repose sur la recherche empirique d'une chaîne maximale d'images pour la relation $>_c$ (première étape). Cette chaîne est utilisée comme une échelle pour préciser les positions relatives de l'ensemble des images de la sous-séquence (deuxième étape).
Ces positions relatives sont exploitées pour concentrer l'annotation sur les périodes de « mauvais temps », pendant lesquels la visibilité horizontale fluctue (troisième étape). Comme pour AMOSvv, les nouvelles relations sont stockées dans des graphes, notés $\mathcal{G}_1^v$, $\mathcal{U}_1^v$ et $\mathcal{E}_1^v$ (quatrième étape). Les deux premières étapes sont représentées sur la Figure A-3).

D.2.2.1 Prédiction d'une chaîne maximale par sous-séquence

La première étape est effectuée en parcourant la sous-séquence dans l'ordre chronologique par deux fois. Au premier parcours, on utilise le classifieur pour trouver des maxima successifs (arcs rouges, figure A-3). Par exemple, sur la figure A-3, x_7 est la première image classée comme strictement supérieure à x_0. Puis x_{11}, la première supérieure à x_7, etc.
Au second parcours, on complète la chaîne par des minima successifs (arcs bleus). La taille m de la chaîne dépasse rarement dix images.

D.2.2.2 Définition d'une échelle

Une fois la chaîne $c = (x_{\phi(k)}), k \in [1, m]$ obtenue, on compare les m images de la chaîne aux n images de la sous-séquence, pour une complexité globale négligeable devant n^2.
Les résultats de ces comparaisons sont représentés sur la figure A-3 (à droite). Seules les relations d'incomparabilité ont été représentées (segments gris verticaux). Lorsqu'une image est associée à plusieurs segments, ces segments sont reliés. Par exemple x_{11} est associée à trois segments disjoints. Les relier revient à corriger les prédictions du classifieur par $x_{11} \perp x_7$ et $x_{11} \perp x_8$.
La représentation obtenue est alors compatible avec un ordre d'intervalle (voir section 4.2).

D.2.2.3 Recherche d'épisodes de mauvais temps

Pour trouver les épisodes de « mauvais temps », on procède de façon empirique. La procédure suivante a été réglée sur les 50 premières séquences d'images. On considère d'abord la fonction :

$$\mathcal{I}(k) = card(\{j | x_j \perp_c x_{\phi(k)}\})$$

On détermine $s = max(\mathcal{I}(k))$, puis $k^* = min\{k | \mathcal{I}(k) > 0.5s\}$. k^* est un indice-seuil empirique : on considère qu'au dessus, les images de la chaîne sont probablement associées à un temps clair.
On décompose alors la sous-séquence en l'union disjointe :

$$S = S_{haze} + \overline{S_{haze}}$$
$$\text{où} \quad S_{haze} := \{x_j | k > k^* \implies x_j <_c x_{\phi(k)}\}$$

S_{haze} contient ainsi les images correspondants aux épisodes de faible visibilité. Ces images sont ajoutées aux noeuds de $\mathcal{D}_1^v$ et $\mathcal{U}_1^v$. Aux même graphes, on ajoute ensuite autant d'images prises au hasard dans $\overline{S_{haze}}$. Ainsi, lorsque $S_{haze} = \emptyset$, la séquence n'est pas utilisée. A la fin de la procédure, on a sélectionné 135.430 issues de 4.390 séquences (parmi les 12.241 de départ). Ces séquences sont issues de 2.596 répertoires d'AMOS et comptent donc au moins autant de caméras différentes.

D.2.2.4 Sélections des arêtes de $\mathcal{G}_1^v$, $\mathcal{U}_1^v$ et $\mathcal{E}_1^v$

Pour sélectionner les arêtes de $\mathcal{G}_1^v$ et $\mathcal{U}_1^v$, nous utilisons les intervalles représentés sur la figure A-3. On élimine les situations équivoques :
si x_a est associée à l'intervalle $[k_a^-, k_a^+]$ et x_b à $[k_b^-, k_b^+]$, on ne décide $x_a > x_b$ que si $k_a^- > k_b^+ + 1$ et $x_a \perp x_b$ que si $k_a^- \leq k_b^+ - 1$ et $k_b^- \leq k_a^+ - 1$. Par exemple, sur la figure A-3, $x_3 \succ x_8$ et $x_1 \perp x_2$.

Ces relations sont stockées dans les graphes $\mathcal{D}_1^v$ et $\mathcal{U}_1^v$. Nous stockons aussi dans le graphe $\mathcal{U}_1^v$ les positions relatives des intervalles (chevauchement ou inclusion), en vue d'entraîner les fonctions de rang bivaluées (section 4.2) en les codant comme sur la figure 4.3.
Enfin, les éléments maximaux de $\overline{S_{haze}}$ correspondent dans leur très grande majorité à des images prises par temps clair, pour lesquelles les visibilités sont indiscernables. Nous pouvons donc considérer les paires de tels éléments (pris dans une même séquence) comme équivalentes. Elles sont stockées dans le graphe $\mathcal{E}_1^v$. Les graphes résultants sont présentés dans la table 5.3.

D.3 Approches multi-tâches

Dans cette thèse, l'approche multi-tâche a été abordée de plusieurs façons différentes. La plus-value sur la tâche d'intérêt n'a jamais été évidente, sauf dans un cas particulier. Ces résultats peu encourageants ne permettent pas d'écarter les approches multi-tâche de nos domaines d'application. L'investissement en temps n'a peut-être pas été suffisant. En approfondissant le choix de l'architecture (voir par exemple [72]), des performances plus importantes auraient peut-être été atteintes. Cette absence générale de plus-value ne doit donc pas être considérée comme un résultat. Dans les paragraphes qui suivent, nous donnons quelques précisions sur les façons dont l'approche multi-tâche a été mise en oeuvre.

Pendant la thèse, trois approches « multi-tâche » ont été expérimentées.
La première consistait à apprendre des préférences sur plusieurs paramètres simultanément (section D.3.1). C'est l'équivalent d'une classication multi-attribut dans le domaine du learning-to-rank.
Avec la seconde approche, un apprentissage de préférences était conduit en parallèle à une tâche de classification (section D.3.2). La dernière approche consistait à considérer le cas d'égalité d'un apprentissage des préférences comme une tâche à part. Cette dernière modalité, mise en oeuvre pour l'apprentissage des fonctions bivaluées, a déjà été décrite en section 4.2. Dans la suite nous revenons sur les deux premières approches.

D.3.1 Apprentissage des préférences multi-paramètre

La première approche a d'abord été mise en oeuvre à travers des tâches de prédiction par paire, soit à travers des tâches de prédiction par image. Pour chacune de ces formes d'apprentissage, nous avons construit un jeu de données spécifique. Dans les deux cas, nous avons prêté attention à la question de la corrélation entre les variables d'intérêt. Nous avons ainsi cherché à augmenter la fréquence relative des paires d'images sur lesquelles les paramètres ne varient pas dans le sens habituel (paires contradictoires). En renforçant ces paires contradictoires, nous espérions aider les modèles à bien séparer les paramètres.
Nous avons testé des réseaux de neurones standard pour lesquels le nombre de neurones de la couche finale est doublé ou triplé (cas de la prédiction par image), ou bien des réseaux à « deux têtes » (cas de la prédiction par paire), partageant les mêmes couches de convolution, mais avec une tête (classifieur) par paramètre. La fonction de coût utilisée est celle de [73].
Evalués sur les diagrammes correctness-completeness et prudence-completeness, ces modèles se sont montrés aussi performants dans le cas de la prédiction par paire (vvss_pl_vgg13_du00.3 et . Dans le cas de la prédiction par image, ils se sont montrés moins performants que des

modèles appris sur la tâche d'intérêt.

D.3.2 Apprentissage des préférences + classification

De même, la seconde approche a été évaluées suivant les deux modalités (prédiction par paire et par image) mais sans effort particulier de rééquilibrage. Les tests n'ont été conduits que sur la prédiction de l'étendue du manteau neigeux, avec une tâche de classification supplémentaire qui consistait à la classification neige non-neige.
Là encore, aucune plus-value n'a été constatée (comparer par exemple ss_sl_due000.0 à ss_slcl_due000.0 sur les figures du chapitre 5).

Par contre, sur les images de synthèse de la section D.4, une fonction de rang bivaluée apprise avec une tâche de classification binaire complémentaire montre un comportement intéressant : les performances sur la tâche de tri ne changent pas mais les deux classes sont beaucoup mieux séparées. Cet effet n'a pas été sensible sur les données réelles, faute, peut-être, de disposer d'un réservoir d'images sans neige assez étendu dans AMOSss.0.

D.3.3 Conclusion

Sur nos problèmes, en dehors d'un impact positif obtenu sur des données de synthèse, l'approche multi-tâche n'a pas eu d'effet.

D.4 Prédictions d'un ordre d'intervalle sur un jeu de synthèse

D.4.1 Définition du problème

Pour entraîner à la restitution d'un ordre d'intervalle, nous avons créé un jeu de paires d'images comparées suivant une relation d'ordre d'intervalle.
Les images, illustrées figure A-4 sont des tableaux 2D (un seul canal) générés par la procédure suivante :

— Pour chaque image i, on tire au hasard une valeur μ_i dans l'intervalle $[1, 10]$ et une valeur σ_i dans l'intervalle $[0, 1]$.
— On tire au hasard la position du centre d'un disque de rayon 15 pixels.
— Les intensités des pixels du disque sont des tirages i.i.d. d'une loi normale centrée sur μ_i d'écart-type σ_i. Le fond de l'image est d'intensité nulle.
— A l'image résultante, on superpose des rectangles pleins ou bruités (voir figure A-4)

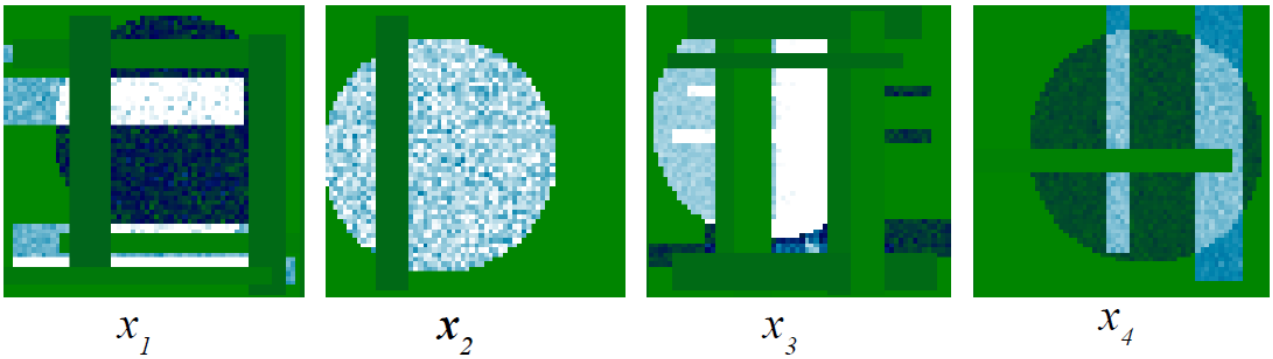


FIGURE A-4 – Exemple d'images du jeu de synthèse. En général, les disques contiennent assez d'information pour estimer les bornes d'un intervalle contenant la valeur cible.

Pour générer l'annotation, nous considérons que la valeur cible associée à l'image i est comprise dans l'intervalle $\mathcal{I}_i = [\mu_i - p\sigma_i, \mu_i + p\sigma_i]$ et que l'annotation disponible est donnée par l'ordre d'intervalle défini par les $\mathcal{I}_i$ (voir figure A-5). Le paramètre p contrôle la résolution de l'annotation.
Les jeux d'entraînement (SYN10000) et de validation (SYN2000) comprennent respectivement 10.000 et 2.000 images. Pour l'annotation, on fixe $p = 2$. On sélectionne au hasard dix comparaisons strictes et dix incomparabilités par image pour l'entraînement (soit 200.000 paires annotées). Pour chaque image de l'ensemble de validation, on choisit une seconde image au hasard dans le même ensemble pour former une paire. Il a donc 2000 paires annotées dans le jeu de validation.

Nous avons essayé d'apprendre l'ordre d'intervalle contenu dans les données suivant deux approches différentes. La première approche est indirecte. Elle comporte trois étapes.

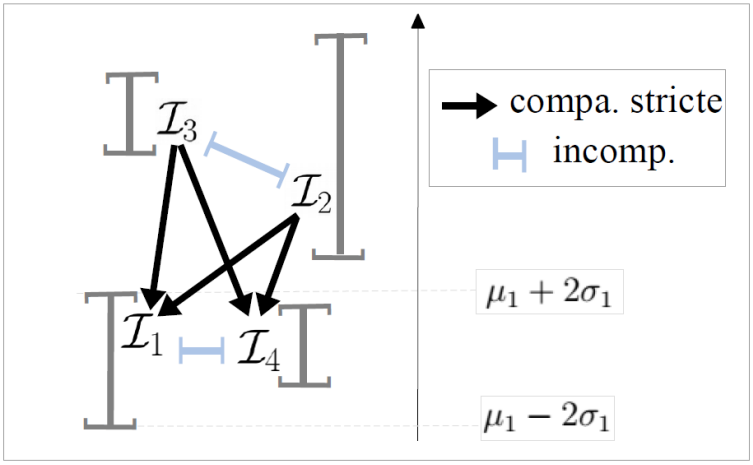


FIGURE A-5 – Ordre d'intervalle associé aux images de la figure A-4.

En dehors de la deuxième étape, qui sera exploitée au chapitre 5, nous n'en présentons ici que les grandes lignes. La deuxième approche implique des fonctions d'ordre bivaluées apprises de bout en bout. La procédure d'apprentissage est déjà décrite au chapitre 4. Ici, nous vérifions qu'elle fonctionne sur le jeu d'images de synthèse. Nous évoquons aussi des formes d'implémentation alternatives qui n'auront finalement pas été retenues.

D.4.2 L'approche indirecte :

La première étape consistait à apprendre un classifieur (VGG11) sur une tâche de comparaison par paires. Le classifieur (un VGG11) réussit sans difficulté à reproduire les comparaisons (justesse de plus de 98 % sur le jeu de validation).
La deuxième étape consistait à reconstruire l'odre d'intervalle à partir des sorties du classifieur par la méthode variationnelle décrite dans l'encadré D.1. La convergence de varIO est vérifiée expérimentalement sur le jeu d'images de synthèse avec le paramétrage par défaut.
La troisième consistait à apprendre une fonction d'ordre bivaluée en ciblant les bornes des intervalles fournis par la méthode variationnelle. La fonction de coût est la RMSE. Le réseau (un autre VGG11) n'a alors aucun mal à apprendre les bornes des intervalles.

ENCADRÉ D.1 – varIO : reconstruction d'un ordre d'intervalle à partir de comparaisons binaires

Les images sont notées x_k, les intervalles $\mathcal{I}_k$ sont paramétrés par leur centre z_k et leur rayon r_k. La méthode s'apparente à une descente de gradient stochastique.

Initialisation :
Les centres et les rayons sont initialisés suivant les lois uniformes respectives $\mathcal{U}([0,1])$ et $\mathcal{U}([0.1,1])$.

Mise à jour (**MJ**) :
A chaque itération, on sélectionne au hasard deux images x_i,x_j pour lesquelles on a un label $c_{i,j}$ appartenant à $\mathcal{C} = \{\dot{\prec}, \dot{\succ}, \dot{\perp}\}$. On définit alors :

$$\begin{aligned}
\mathcal{J}_{\dot{\prec}}(z,z',r,r') &= \mathcal{L}^{h_0}(z'-z-r-r')) \\
\mathcal{J}_{\dot{\succ}}(z,z',r,r') &= \mathcal{J}_{\dot{\prec}}(z',z,r',r) \\
\mathcal{J}_{\dot{\perp}}(z,z',r,r') &= \mathcal{L}^{h_1}(r+r'-|z'-z|)
\end{aligned}$$

où $\mathcal{L}^h$ est la Hinge Loss, définie au chapitre 3 (equation 3.5). On en déduit le potentiel d'interaction :

$$\mathcal{J}(z_i,z_j,r_i,r_j,c_{i,j}) = \sum_{c\in\mathcal{C}} \delta_{c=c_{i,j}} \mathcal{J}_c(z_i,z_j,r_i,r_j) + \mathcal{L}^{h_2}(r_i) + \mathcal{L}^{h_2}(r_j)$$

où δ est le symbole de Kronecker. Les deux derniers termes maintiennent positives les valeurs des rayons. L'opération de mise à jour avec un taux d'apprentissage de η_m est donnée pour $k \in \{i,j\}$ par :

$$y_k := y_k + \eta_m \partial_y \mathcal{J} \tag{A-1}$$
$$r_k := r_k + \eta_m \partial_r \mathcal{J} \tag{A-2}$$
$$\tag{A-3}$$

L'algorithme varIO est alors défini par :

Pour $m = 0..m_{max}$:

—Choisir : $\eta_m = \eta_0 \times \dfrac{1}{2^m}$
—Appliquer **MJ** à n couples pris au hasard

Par défaut, les paramètres de la méthode sont réglés de la manière suivante :
$\eta_0 = 0.1$ $\quad h_0 = 0.05$ $\quad h_1 = 0.05$ $\quad h_2 = 0$ $\quad m_{max} = 5$ $\quad n = 10^6$

D.4.3 L'approche directe

D.4.3.1 Avec la méthode décrite au chapitre 4 :

Les fonctions d'ordre bivaluées sont testées ici dans une situation idéale : d'une part l'annotation est parfaitement compatible avec un ordre d'intervalle, de l'autre l'image contient suffisamment d'information pour prédire des tailles d'intervalle cohérentes.
Dans ces conditions, on observe la convergence progressive du réseau (un VGG11) vers un isorphisme d'ordre dans le sens où la justesse du problème à trois classes $\{\dot{\prec}, \dot{\succ}, \dot{\perp}\}$ tend vers 1. Précisément, entraîné dans les conditions standards de la table 3.6 avec la fonction de coût de l'équation 4.3, un VGG11 atteint 98 % en validation sur ce problème.

La matrice de confusion donnée table A-1 est obtenue au bout de cinq cents époques sur le jeu de validation. Il n'y a aucune discordance. Pour détailler les autres formes d'erreurs, la classe $\dot{\perp}$ a été découpée en quatre sous-classes correspondant aux quatre cas d'intersections non-nulle entre deux intervalles : les deux chevauchements, notés $\dot{\preccurlyeq}$ et $\dot{\succcurlyeq}$, et les deux inclusions, notées $\dot{\subset}$ et $\dot{\supset}$. La majorité des fautes de prudence et des rejets commis le sont entre ordre strict et chevauchements de mêmes sens (en orange dans la matrice). Ces erreurs tiennent donc à des intervalles légèrement décalés ; le même phénomène se retrouve dans la matrice de confusion entre chevauchement et inclusion (en vert) [1].

Ces légers décalages sont visible sur la figure A-6 où les cercles noirs indiquent une erreur de rejet (voire une discordance). Cette nouvelle représentation permet de mieux apprécier la qualité de la restitution. Dans l'idéal, l'ensemble des points se trouve sur le graphe d'une fonction bijective (au moins la partie centrale).

D.4.4 Variantes

Avec la configuration précisée dans la table 3.6, les performances en validation maximales ne sont atteintes qu'après plusieurs milliers d'époques. Nous avons donc essayé d'accélérer la convergence en modifiant la méthode d'apprentissage. Nous listons ici les pistes qui ont été tentées et celles qui restent à suivre.

— *Paramétrisation par centres et rayon*. L'idée était de donner un autre sens aux neurones de sorties du réseau : le premier (z_0) devant fournir le centre de l'intervalle et le second (z_1), le rayon. La fonction de coût $\mathcal{L}_{du}$ est appliquée au couple ($z_1 - z_2$,$z_1 + z_2$). Cette paramétrisation n'a pas eu d'effet sur l'apprentissage.

1. Les chiffres en vert ne correspondent pas nécessairement à des erreurs du fait que les positions relatives des intervalles associés aux éléments extrémaux pour l'ordre d'intervalle ne sont pas déterminées.

pred. / obs.	$\delta_{i,j}$	$\prec$ {+1;+1;+1;+1}	$\succ$ {-1;-1;-1;-1}	$\precsim$ {+1;+1;-1;+1}	$\succsim$ {-1;+1;-1;-1}	$\subset$ {-1;+1;-1;+1}	$\supset$ {+1;+1;-1;-1}
ordre strict	$\prec$ {+1;+1;+1;+1}	643	0	17	0	1	1
	$\succ$ {-1;-1;-1;-1}	0	655	0	15	0	1
chevauchement	$\precsim$ {+1;+1;-1;+1}	31	0	165	1	7	7
	$\succsim$ {-1;+1;-1;-1}	0	24	1	177	6	10
inclusion	$\subset$ {+1;+1;-1;+1}	3	5	14	26	80	0
	$\supset$ {+1;+1;-1;-1}	3	1	21	18	1	76

TABLE A-1 – Matrice de confusion sur le jeu de validation SYN2000 du modèle SYN_slbiv_VGG11.0, à l'époque 500. La classe $\perp$ a été redécoupée (voir texte). Les vecteurs δ_{ij} correspondent à l'encodage des positions relatives d'intervalles défini sur la figure 4.3).

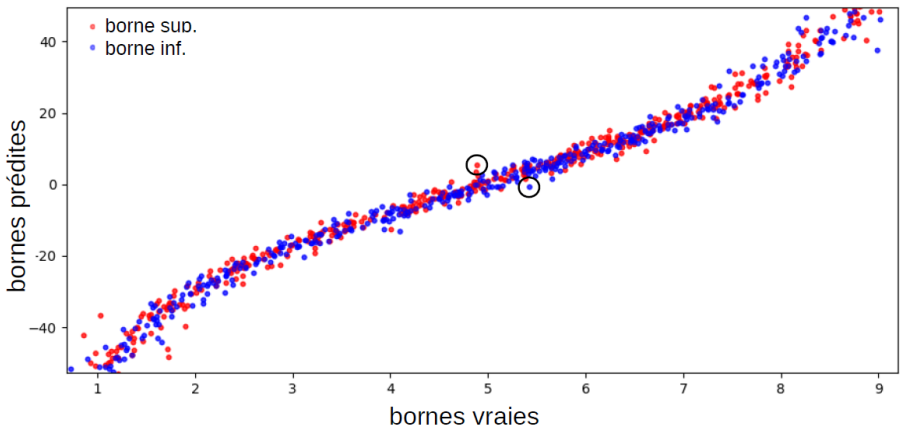


FIGURE A-6 – Correspondance entre les bornes supérieures et inférieures prédites et observées. Les points rouges (resp. bleus) sont de coordonnées $(\mu_i+2\times\sigma_i, z_i^+)$ (resp. $(\mu_i-2\times\sigma_i, z_i^-)$) où les $z_i^\pm$ sont les sorties de la fonction d'ordre bivaluée. Le diagramme est effectué sur les 500 premières images de SYN2000. La prédictions sont obtenues avec la fonction d'ordre bivaluée SYN_slbiv_VGG11.0. Une erreur est commise lorsqu'entre un point rouge et un point bleu, l'ordre des ordonnées est contraire à celui des abscisses (par exemple, entre les deux points cerclés de noir.

— *Approche séquentielle*. Intuitivement, il semble être plus simple de ne choisir le rayon qu'après avoir choisi le centre. Nous avons donc essayé d'apprendre le réseau séquentiellement. L'apprentissage porte alternativement sur le choix du centre (premier canal) et celui du rayon (deuxième canal). Plusieurs formes d'implémentation ont été tentées, aucune avec un succès net.

— *Apprentissage par listes*. Nous avions vérifié sur un problème de restitution d'ordre total, la fonction de coût ListNet [125], qui généralise RankNet aux arrangements sur des listes de taille n quelconque, accélère efficacement la convergence d'un réseau de neurone. Il aurait été intéressant de définir un apprentissage par liste d'un ordre d'intervalle. Généraliser $\mathcal{L}_{du}$ de façon à pénaliser des arrangements plutôt que de simples transpositions se fait sans difficulté. Le problème serait plutôt d'adapter une telle fonction de coût au cas où les arrangements ne sont pas complètement connus : en effet les vraies données ne déterminent pas entièrement la relation d'ordre sur chaque séquence.

— *Rééquilibrage*. Les exemples qui vont contribuer au réglage fin du modèle sont de plus en rares. En première approximation, dans SYN10000, le nombre de paires d'intervalles $(\mathcal{I}_i, \mathcal{I}_j)$ telles que $|\mathcal{I}_i^+ - \mathcal{J}_j^-| < \epsilon$ décroît linéairement avec ϵ. Il y a donc une analogie avec un problème de jeu déséquilibré. Une stratégie de pondération, soit au niveau du tirage des batches, soit au niveau de la fonction de coût, pourrait contribuer à une amélioration. Cette piste est d'autant plus intéressante qu'elle peut être facilement mise en oeuvre avec les données réelles, en favorisant par exemple les arêtes de la réduction transitive du graphe $\mathcal{G}_1$. Cela aurait aussi pour conséquence la surreprésentation des arêtes associées aux paires d'images consécutives. Quoi qu'il en soit, cette piste n'a pas été testée.

Pour terminer, citons deux approches qui permettent d'améliorer les performances en validation. La première consiste à utiliser la moyenne de plusieurs fonctions d'ordre bivaluées. Cette approche est décrite dans le chapitre 4, où elle est mise en oeuvre. La seconde consiste à utiliser, dans cette moyenne, des fonctions de rang entraînées à restituer la relation duale. Cette stratégie a été mise en oeuvre sur des images réelles pour caractériser l'étendue du manteau neigeux (voir l'annexe D.5.2).

D.5 Compléments sur les résultats des fonctions de rang bivaluées

D.5.1 Prudence sur les scènes difficiles

Dans cette section, nous cherchons à savoir si les fonctions de rang bivaluées sont plus prudentes sur les scènes les moins propices aux comparaisons à la main.
Par exemple, sur les scènes où seul le premier plan est visible, ou au contraire, lorsqu'il n'y a pas de premier plan, les incomparabilités sont relativement plus fréquentes.
Pour le vérifier, nous avons créé des scènes peu propices à partir de scènes normales. Nous avons rassemblé les séquences du jeu de test qui présentaient un horizon plus haut que les deux tiers (scène à horizon haut) et des séquences qui présentaient un horizon compris entre la moitié et le tiers inférieur de l'image (scène à horizon bas).
Pour chacune de ces séquences, deux autres séquences d'images sont créées, l'une par rognage du tiers inféreur des images (rognage bas), l'autre par rognage du tiers supérieur (rognage haut). Pour une séquence à horizon bas, c'est le rognage bas qui supprime une bonne part de l'information utile, pour une séquence à horizon haut, c'est le rognage haut.
Pour observer les effets des rognages sur l'incomparabilité, nous avions plusieurs possibilités. Soit, utiliser un taux d'incomparabilité (nombre de paires considérées comme incomparables sur le nombre de paires totales), soit chercher à comparer des « largeurs d'intervalles ». Nous avons choisi la deuxième voie [2].
Pour déterminer des largeurs d'intervalle qui soient comparables, nous nous ramenons à un rang. Précisement :
-la séquence est d'abord convertie en un ensemble d'intervalles $[z_i^-, z_i^+]$ par la fonction de rang bivaluée (vv_sl_due111.0).
-on considère ensuite l'ensemble formé par toutes les bornes des intervalles $E = \bigcup_i \{z_i^-; z_i^+\}$ et la fonction de répartition empirique associée à E, notée $\mathcal{R}_E$.
Les intervalles $[\mathcal{R}_E(z_i^-), \mathcal{R}_E(z_i^+)]$ forment une représentation canonique de l'ordre d'intervalle.
Pour des séquences de même taille, les chevauchements sont d'autant plus nombreux que les largeurs des intervalles $l_j = \mathcal{R}_E(z_j^+) - \mathcal{R}_E(z_j^-)$ sont grandes. Nous utilisons alors la largeur moyenne $\overline{l_j}$ sur la séquence pour caractériser le taux d'incomparabilité.
Pour chaque séquence, nous calculons ainsi une largeur moyenne d'intervalle après rognage bas ou rognage haut (voir figure A-7).

Comme attendu, pour les scènes à horizon haut, les intervalles sont relativement plus larges après rognage haut et inversement pour les scènes à horizon bas. La largeur de l'intervalle semble donc capter aussi certaines difficultés liées à la scène.
Néanmoins, un autre phénomène peut expliquer cette situation. Les scènes les plus difficiles sont aussi les plus rares. Or, sur les outliers, le modèle est souvent conduit à réduire

2. a posteriori, ce choix était probablement maladroit, car inutilement compliqué et sensible aux valeurs extrêmes.

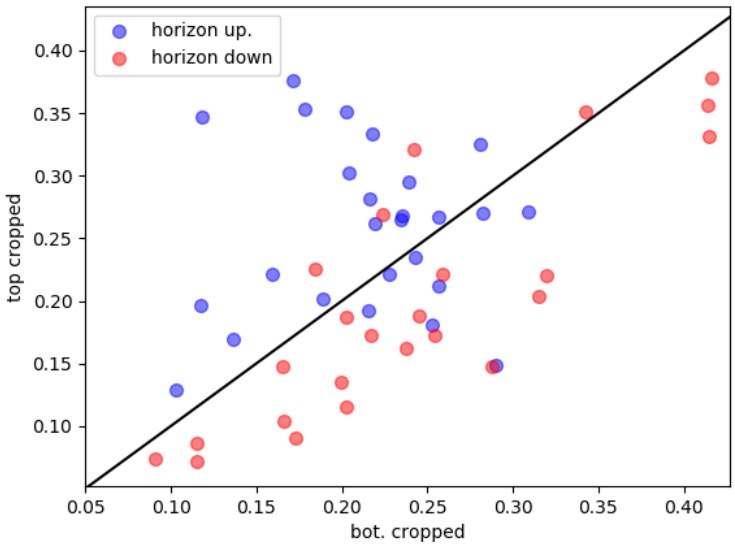


FIGURE A-7 – Chaque point correspond à une séquence du jeu de test. Les points rouges (resp bleus), aux scènes à horizon bas (resp. à horizon haut). Les coordonnées correspondent aux largeurs moyennes des intervalles après rognage bas (abscisses) ou haut (ordonnées).

les écarts de prédiction pour optimiser l'espérance de la fonction de coût. Le modèle est donc conduit à réduire l'écart entre les prédiction sans nécessairement avoir appris en quoi consiste une scène peu propice à partir des relations d'incomparabilité. A cet égard, il aurait été intéressant de comparer avec des fonctions de rang à valeurs réelles épaissies.

D.5.2 Prédiction des inclusions

Dans cette section, nous nous posons la question du gain en performance sur la restitution des incomparabilités en comparaison aux fonctions de rang épaissies. Précisément, nous cherchons à savoir si les intervalles sont positionnés et « taillés » de manière à économiser des fautes de prudence.
Pour répondre à cette question, nous nous sommes d'abord intéressés aux situations d'inclusion. Nous avons vu section 4.2.3 que l'annotation contraint les positions relatives des bornes des intervalles.
Parfois, la configuration des graphes n'est compatible qu'avec une inclusion. Nous parlerons d'inclusion inférée.
Or, une fonction d'ordre univaluée épaissie ne peut pas générer une inclusion. Une fonction bivaluée le peut et pour chaque inclusion correctement prédite, elle économise une faute de prudence.

On peut évaluer quantitativement la capacité des modèles à prédire correctement les inclusions inférées. Parmi toutes les inclusions inférées, seul un petit nombre sont prédites.

Nous appellerons rappel cette proportion. Par exemple, le modèle vv_sl_due111.0 ne reconnaît correctement que $23 + 26$ des 647 inclusions (table A-2). Par contre, les 66 inclusions qui sont prédites et inférées le sont généralement dans le même ordre (49 cas sur 66). Cette proportion (« cohérence ») est plus importante pour des modèles appris avec une fréquence de présentation des paires incomparables plus élevée (table A-3). Elle s'améliore aussi lorsque l'on restreint l'analyse aux paires dont l'une des images est de mauvaise qualité (paires « bruitées »).

pred. / obs.	$\delta_{i,j}$	ordre strict		chevauchement		inclusion	
		$\prec$ {+1;+1;+1;+1}	$\succ$ {-1;-1;-1;-1}	$\precsim$ {+1;+1;-1;+1}	$\succsim$ {-1;+1;-1;-1}	$\subset$ {-1;+1;-1;+1}	$\supset$ {+1;+1;-1;-1}
ordre strict	$\cdot\succ$ {+1,+1,+1,+1}	0	0	0	0	0	0
	$\cdot\prec$ {-1,-1,-1,-1}	343	11991	539	1792	164	188
chevauchement	$\cdot\succsim$ {+1,+1,-1,+1}	304	42	106	72	10	30
	$\cdot\precsim$ {-1,+1,-1,-1}	60	280	68	125	31	13
inclusion	$\cdot\supset$ {+1,+1,+1,+1}	93	128	31	47	23	7
	$\cdot\subset$ {+1,+1,-1,-1}	117	61	72	32	10	26

TABLE A-2 – Matrice de confusion sur le problème à six classes pour le modèle vv_sl_due111.0. Les quatre dernières classes sont des sous-classes de la relation d'incomparabilité. Elles sont définies par les positions relatives des intervalles, chevauchements stricts ou inclusions, encodées dans les $\delta_i j$ définis section 4.2.3.

Enfin, curieusement, les erreurs commises sur les inclusions inférées ne sont pas symétriques : tous les modèles placent plus souvent le petit intervalle au-dessus du gros plutôt qu'en dessous (chiffres rouges dans la table A-2). Cette tendance avait déjà été observée sur les matrices de confusion de modèles entraînés sur le jeu d'image de synthèse. C'est cette raison qui nous a conduit à apprendre une relation dans « l'autre sens » (voir section 5.3.2, modèles ss_sl_rev_due222.0-1), pour compenser par moyennage.

modèles	Jeu de test complet		Paires bruitées	
	rappel	cohérence	rappel	cohérence
vv_sl_du11.0	0.06	0.73	0.07	0.78
vv_sl_du11.1	0.08	0.75	0.11	0.77
vv_sl_due111.0	0.08	0.74	0.09	0.81
vv_sl_due111.1	0.11	0.76	0.12	0.94
vv_sl_due111.2	0.14	0.86	0.19	0.87
vv_sl_due111.3	0.21	0.85	0.26	0.96
vv_sl_due101.0	0.12	0.88	0.15	0.86
mean_bivalued_gr1gr2	0.06	0.81	0.09	0.86

TABLE A-3 – Prédiction des inclusions dans le jeu de test complet (647 inclusions inférées) et après restriction aux paires d'images comptant au moins une image de mauvaise qualité (417 inclusions inférées).

D.5.3 Démonstration de la propriété 2

D.5.3.1 Modèle de formation de l'image

Ecrivons d'abord le processus de formation d'une image x par une caméra c :

$$x = \mathcal{G}(v, x^s, x^t) \tag{A-4}$$

où $v \in \mathcal{D}_v = [v_{min}, v_{max}]$ est la valeur de la visibilité sur la scène, x^c représente les élements permanents vus par la caméra c (paysage, horizon, etc) et x^t représente les éléments transients (autres paramètres météo, éclairage, masques, piétons, véhicules). Sans réduire la portée du modèle, on peut supposer que $\mathcal{G}$ est un opérateur continu et borné et que son domaine de définition est compact.
Pour pour une caméra c donnée, posons alors E^c l'ensemble des couples image-visibilité (x, v) observables.

D.5.3.2 Démonstration

Plaçons-nous dans le cas où l'ensemble des images générées par la caméra c peut être muni d'un ordre d'intervalle compatible avec l'annotation. Supposons enfin qu'on dispose d'une fonction de rang bivaluée continue f^c qui conserve parfaitement cet ordre d'intervalle.

La compatiblité avec l'annotation implique que pour deux éléments (x_i, v_i) et (x_j, v_j) de E^c, on ait $f^c(x_i)^- > f^c(x_j)^+ \Rightarrow v_i > v_j$.
Définissons alors $\rho_c^+(v)$ (resp. $\rho_c^-(v)$) comme la plus petite borne supérieure (resp.plus grande borne inférieure) prédite par f^c pour une valeur de visibilité mesurée égale à v :

$$\rho_c^+(v) = \min_{\{x|(x,v)\in E^c\}} z^+(x) \qquad \rho_c^-(v) = \min_{\{x|(x,v)\in E^c\}} z^-(x) \tag{A-5}$$

où $f^c(x) = [z^-(x), z^+(x)]$.

On vérifie facilement que les ensembles $\mathcal{I}(v) = \bigcap_{(x,v)\in E^c}[z^-(x), z^+(x)]$ ne peuvent pas être vides ; ces bornes sont donc bien définies sur $\mathcal{D}_v$. Par compacité, ce sont des fonctions continues de la variable v. L'hypothèse 1 implique que les $\mathcal{I}(v)$ sont disjoints ou égaux deux à deux. En effet, plaçons-nous dans le cas d'un chevauchement et supposons, sans perte de généralité, que $\mathcal{I}(v_1)^+ \geq \mathcal{I}(v_2)^-$. Dans ce cas, tout couple (x, v_2) de E^c est indissociable de v_1. Donc (x, v_1) est aussi dans E^c. On en déduit que $\mathcal{I}(v_1)^- = \mathcal{I}(v_2)^-$, et par symétrie, que $\mathcal{I}(v_1) = \mathcal{I}(v_2)$.
L'hypothèse 2 implique que les intervalles sont tous disjoints deux à deux. Cela n'est possible que si les bornes ρ^+ et ρ^- sont confondues et strictement croissantes par rapport à la visibilité.
La fonction réciproque de ρ_s^+, notée $\tilde{g}^c$, est définie sur l'intervalle $\mathcal{I}_f = [\min_{(x,v)\in E^c} z^+(x), \max_{(x,v)\in E^c} z^-(x)$
On peut l'étendre en dehors de cet intervalle par $g^c(z) = v_{min}$ si $z < \mathcal{I}_f^-$ et $g^c(z) = v_{max}$ si $z > \mathcal{I}_f^+$.
Pour un couple (x, v) quelconque dans E^c, on a $\rho_s^+(v) \in [z^-, z^+] = f(x, w)$, ce qui implique $v \in [g^c(z^-), g^c(z^+)]$. On aurait bien sûr pu étendre g_c de manière à former une fonction strictement croissante sur $\mathcal{D}_z$ qui présente la même propriété.

D.5.4 Visualisation des résultats après étalonnage

Compléments sur l'étude qualitative :
Nous complétons ici l'analyse qualitative des séries temporelles relatives à la visibilité qui a été faite au chapitre 4, section 3. Nous illustrons le comportment des prédictions au cours d'épisodes de neige. Nous montrons aussi quelques cas où les largeurs d'intervalles sont incorrectement réglées, des cas d'erreur systématiques (basse visibilité par beau temps) et des cas de mesures non représentatives.

Dans le chapitre 4, nous proposions des cas de brouillard. Nous commençons ici par des épisodes de neige. La figure A-9 représente deux épisodes de neige en plaine enregistrés les 3/11/2012 et 7/11/2012 par les caméras de Nancy et Entzheim.
Sur nancy1, l'ajustement paraît plus médiocre A-9.a, pendant que la visibilité remonte. Mais après vérification, le modèle ordonne les images correctement. Les écarts observés autour de 10 h sont plutôt dus à un défaut de représentativité de la mesure, ou plus probablement à l'imprécision de l'horodatage. En fin de journée, la baisse de la luminosité induit des erreurs

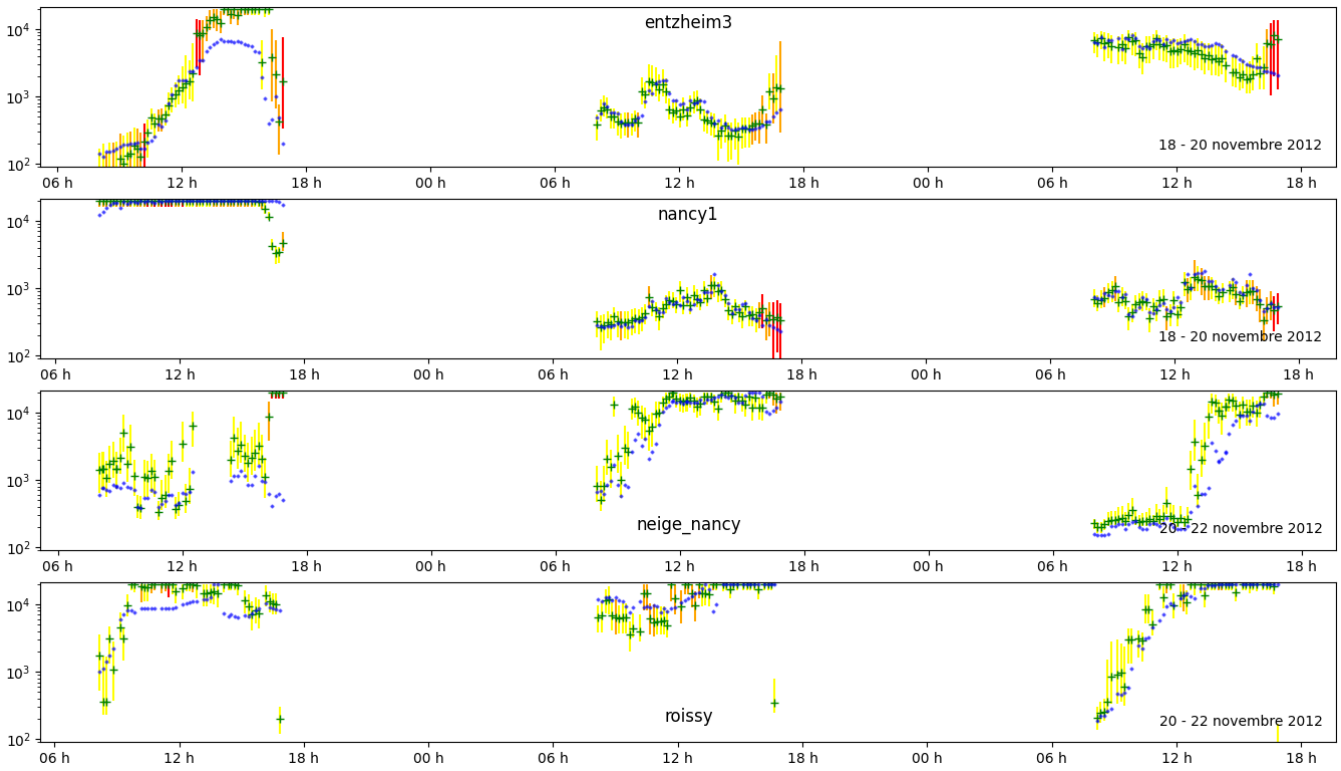


FIGURE A-8 – Comparaison entre prédictions et mesures sur quatre scènes du réseau TENEBRE. La visibilité (en mètres) est en ordonnée, l'heure UTC en abscisse. Les graphiques couvrent des périodes de trois jours, entre 8h et 17h ; ils couvrent au moins une situation de brouillard (visibilité < 1000 m) par scène. Les estimations ponctuelles (croix vertes) sont obtenues par l'équation (4.15), les bornes des intervalles par les équations (4.14 - 4.13). Les grands intervalles sont représentés par les couleurs orange et rouge (voir encadrés 4.2). Les croix bleues représentent les mesures instrumentales

de prudence après 16h (figures A-9.a-c).
Sur les scènes entzheim1 et entzheim3, les chutes de neige ont pollué l'objectif au moment où la visibilité était au plus bas. Les masques les plus opaques génèrent les intervalles les plus larges (figures A-9.b-d).

Sur les caméras nancy2 et entzheim1, le modèle propose aussi de plus larges intervalles pour les images dégradées. Cette aptitude est illustrée sur les figures A-10.a-c, lors de chutes de neige des 12/03/2013 et 20/12/2017. Cependant, sur les caméras neige_nancy et parc_entzheim, (figure A-10), cette aptitude est mise en défaut. Les gouttelettes défocalisées génèrent des visibilités inférieures à 1000 m sur des périodes de plusieurs heures.

Nous pouvons aussi comparer les largeurs d'intervalle d'un modèle à l'autre. Les deux scènes qui présentent les intervalles les plus larges sont dorans1 et portail_entzheim (voir figure A-11 et chapitre II). Cela n'est pas surprenant : ces scènes sont relativement peu propices à l'évaluation de la visibilité. La première correspond, comme pour entzheim1 à une scène à horizon bas, mais sans aucun élément au premier plan (figure A-11.a2). Cette ca-

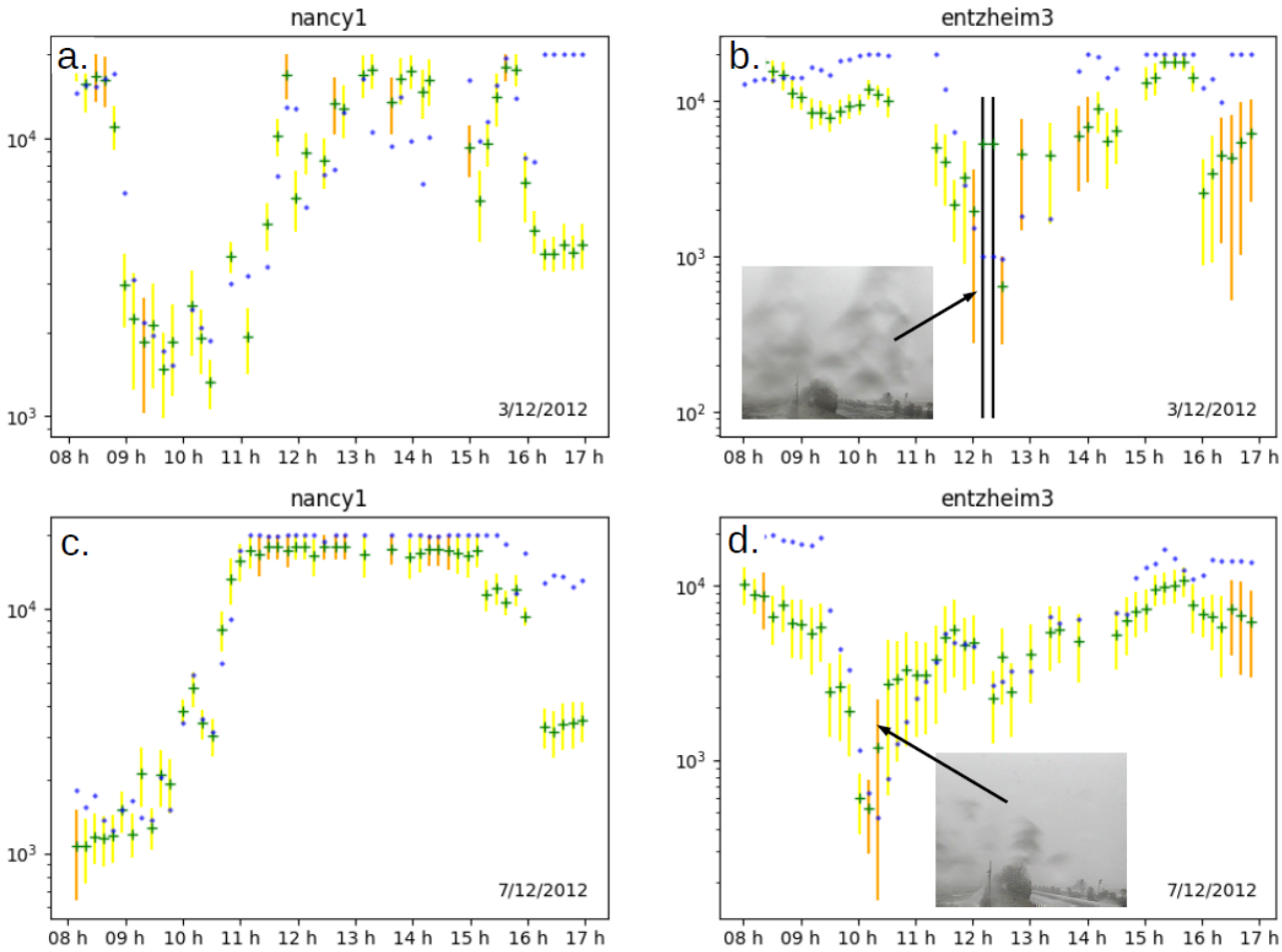


FIGURE A-9 – Comparaison entre prédiction et mesure pendant des épisodes de neige en plaine. Cas de chutes de neige intenses avec tenue de neige au sol. Les plus gros intervalles des figures b. et d. sont prédits pour les images les plus dégradées.

méra est de plus assez mal entretenue. Dans le cas de portail_entzheim, il n'y a pas de second plan. Seules les très basses visibilités ont un impact sensible sur l'image, comme sur la figure A-11. Les intervalles sont d'ailleurs plus larges pour les visibilités élevées.

Enfin, nous avons repéré deux cas d'erreurs récurrents. Le premier concerne les scènes entzheim1 et entzheim3, lorsque le brouillard se dissipe et fait place au beau temps (figure A-11.c1). Le modèle prédit une visibilité de plus de 10km alors que le brouillard ne s'est pas encore levé. La sur-estimation du 18/11 sur la caméra entzheim3 (figure A-8) intervient dans une sitatuation analogue : le temps est beau mais la visibilité horizontale est réduite.
Le deuxième cas concerne la caméra de Roissy. Lors du passage d'averses, la visibilité mesurée peut décroître brusquement pour remonter plus progressivement vers un palier (figure A-11.d). Lorsque l'averse ne passe pas dans le champ de vision de la caméra, la portée optique n'est pas affectée, et le modèle prédit à raison une visibilité élevée. C'est ce qui se produit sur la figure A-11.d.1. Sur la séquence d'images du 10 octobre, les portées optiques observées après 12 h sont stables. Il ne s'agit donc pas d'une erreur de prédiction, mais d'un

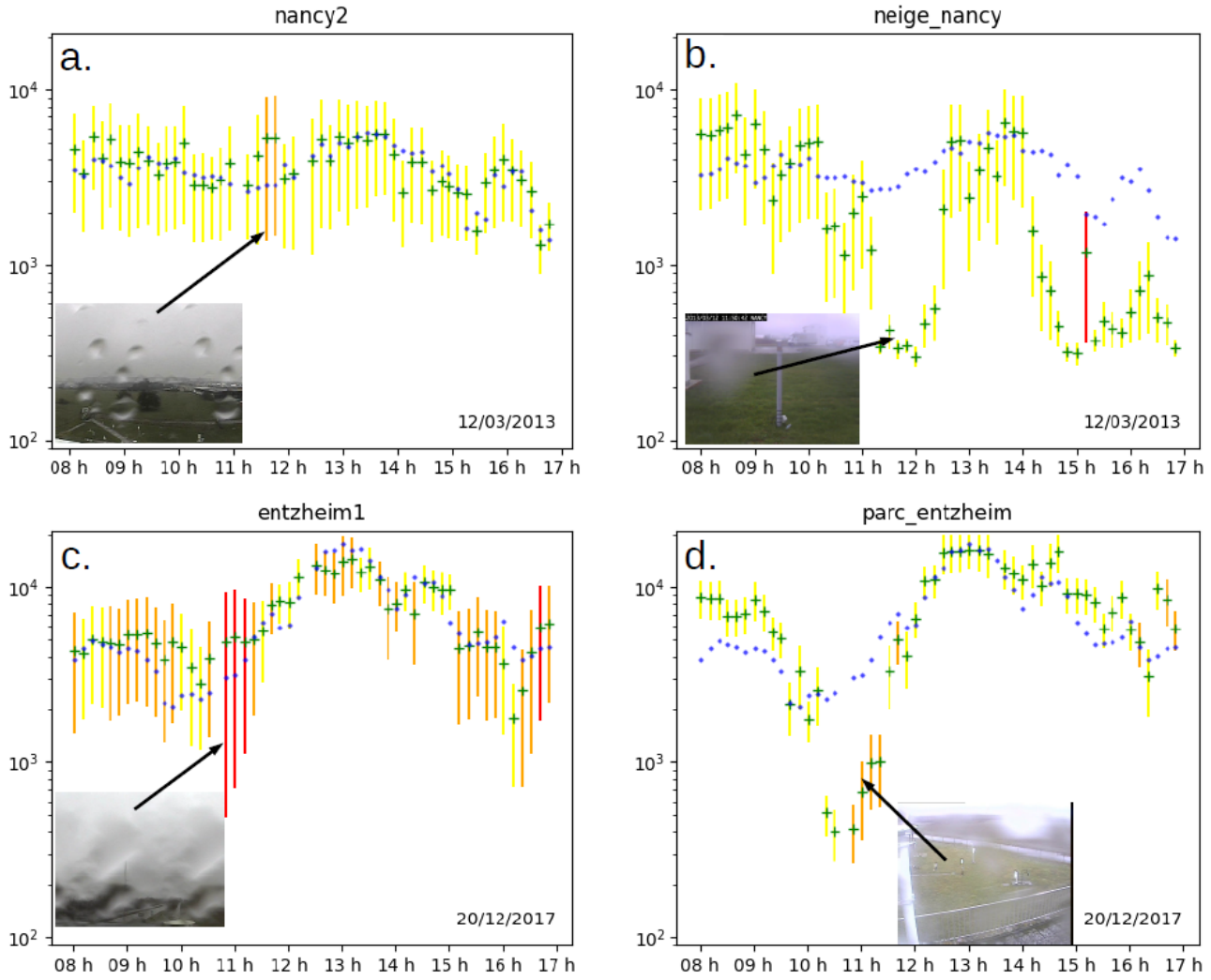


FIGURE A-10 – Comparaison entre prédictions et mesures pendant de faibles chutes de neige. Les prédictions sont correctes pour les scènes nancy2 et entzheim1. Les flocons (gouttelettes ?) défocalisés génèrent d'importantes erreurs de prudence en matinée sur les deux scènes neige_nancy et parc_entzheim, puis à partir de 14h sur neige_nancy.

défaut de représentativité de la mesure, déjà relevé au chapitre II pour la caméra de Roissy (section II.B.4.3).

Compléments à l'étude quantitative :

Enfin, pour compléter l'analyse quantiative du chapitre 4, nous avons cherché un lien entre la largeur des intervalles prédits et l'erreur commise sur l'estimation ponctuelle. Ce lien ne peut pas être déduit du cadre auquel nous nous sommes limités. Mais si l'on veut valider une interprétation probabiliste raisonnable des intervalles prédits, il faut au moins pouvoir vérifier que la dispersion des erreurs croît avec la largeur de l'intervalle.

Est-ce le cas sur sur TENEBRE ? Pour le vérifier, nous avons comparé sur chaque scène l'erreur relative médiane $e^{q50}_{med.}$ associée à l'ensemble des « petits » (voir encadré 4.2) avec l'erreur relative médiane $e^{q90}_{med.}$ associée à l'ensemble des intervalles les plus grands (inter-

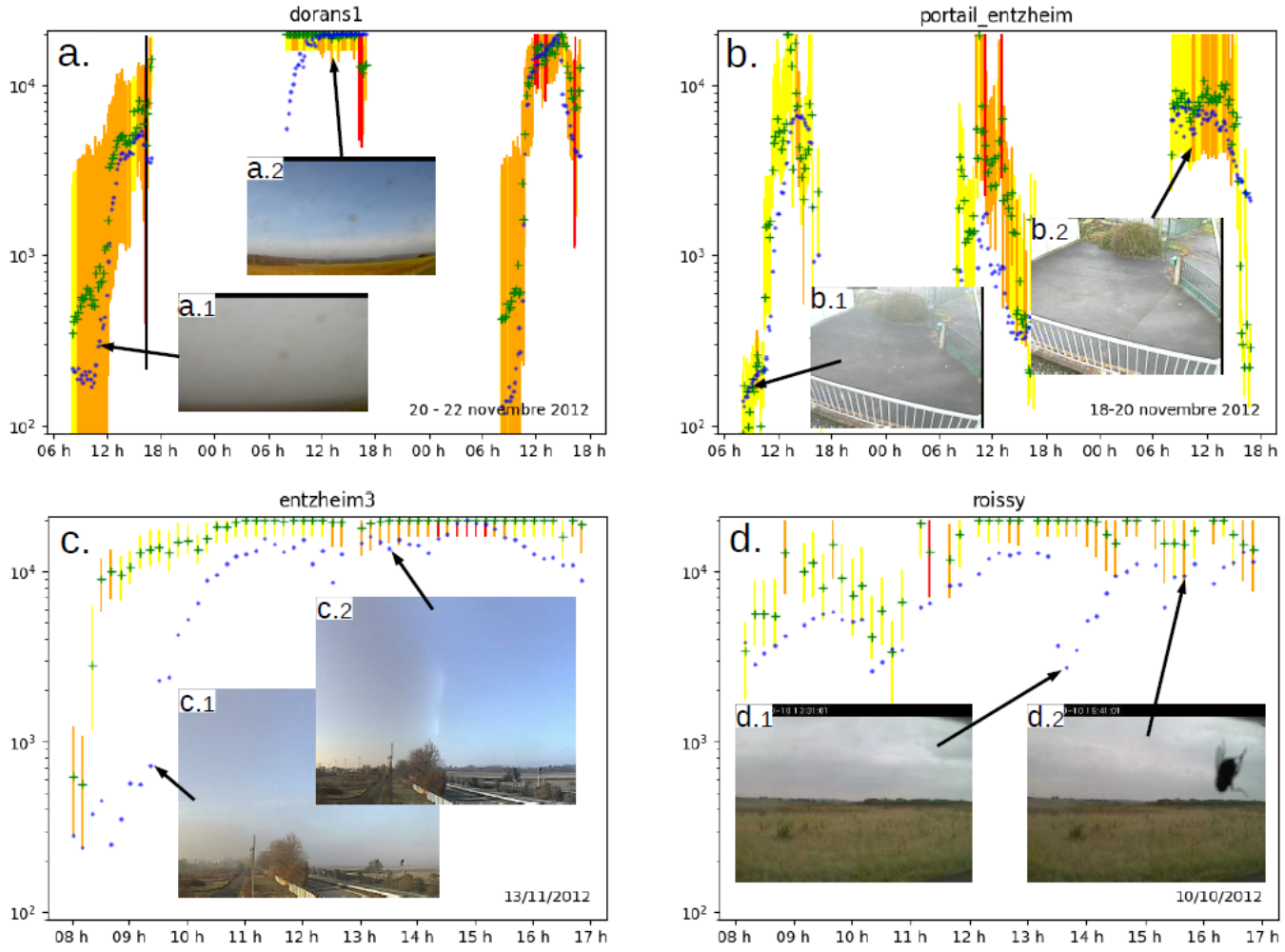


FIGURE A-11 – a.Comparaison entre prédictions et mesures pendant un épisode de brouillard sur la scène dorans1. Il s'agit d'une scène atypique, sans premier plan, à horizon bas (a.2). Pour des basses visibilités, il n'y a plus aucune information sur la structure de la scène (a.1).b. La caméra est rarement entretenue, et des traces sont visibles sur l'image pendant les trois premiers mois. Sur la scène portail_entzheim. c.Cas d'erreur (surestimation). Le brouillard se lève au matin du 13/11/2012 sur Entzheim. A 9h30 (c.1) la mesure n'est pas représentative de la scène mais la portée optique est encore limitée à quelques kilomètres. d.Défaut de représentativé des mesures sur Roissy. Des averses se sont succédées pendant la journée du 10/10/2012. Les fluctuations de la mesure à partir de 12 h n'apparaissent pas sur la séquence d'images (comparer d.1 et d.2).

valles représentés en jaune et rouge). Sur la figure A-12, nous affichons les différences entre ces erreurs médianes pour six modèles entraînés sur les paires incomparables, et un modèle « moyen » (vv_sl_mean_gr1_gr2). Les intervalles les plus larges sont généralement associés à des erreurs plus grandes. Les caméras pour lesquelles le lien entre largeur et erreur relative est le moins évident sont celles sur lesquelles les erreurs médianes sont les plus élevées et les taux d'appartenance les moins bons. C'est par ailleurs sur la caméra de Dorans, mal entretenue, que la différence est la plus importante.

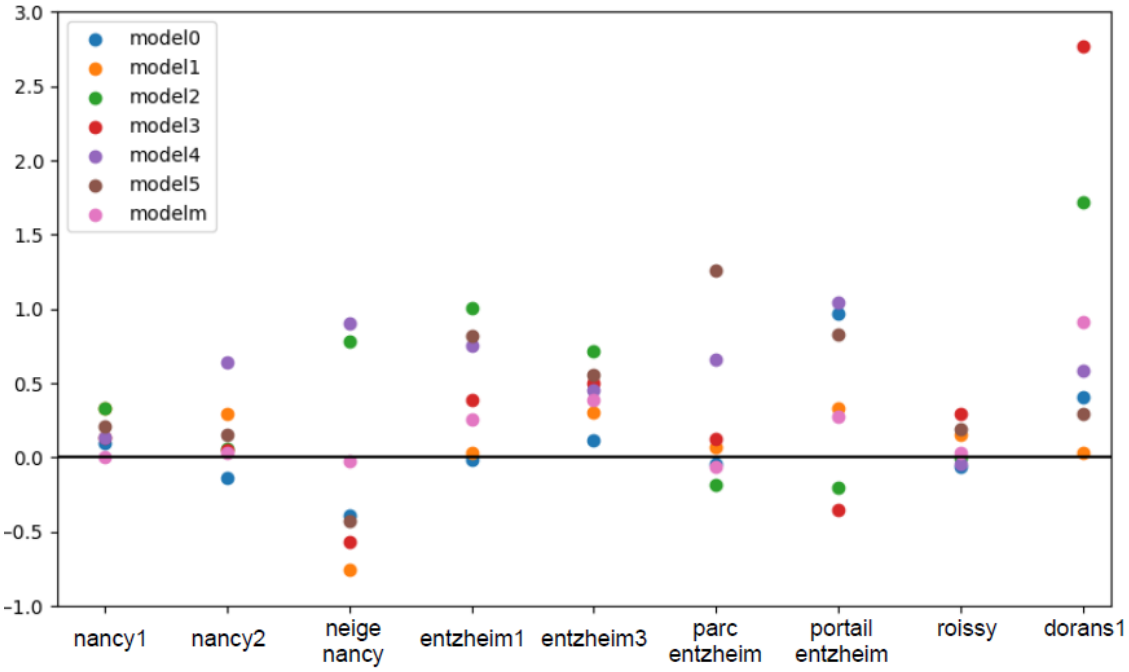


FIGURE A-12 – Différences $e^{q90}_{med.} - e^{q50}_{med.}$ pour différents modèles appris sur les paires incomparables. Les modèles ont tous été étalonnés avec les mesures colocalisées.

D.6 Jeux de données pour l'intercalibration

Dans le paragraphe qui suit sont listés les détails relatifs à la préparation du jeu de données et à l'entraînement pour l'apprentissage en « intercalibration ».
A partir du jeu TENEBREvvExt et des archives d'infoclimat et des DIRs, nous avons rassemblé 3.123 séquences webcam s'étendant chacune sur au moins 4 jours, et contenant au moins 50 images. Ces séquences comptent celles du jeu de test d'AMOSvv mais pas celles du jeu de validation ni celles de TENEBRE.
Au moment où ce jeu a été construit, seul le modèle vv_sl_due111.0 était disponible. C'est ce modèle qui a été utilisé pour trier les séquences.

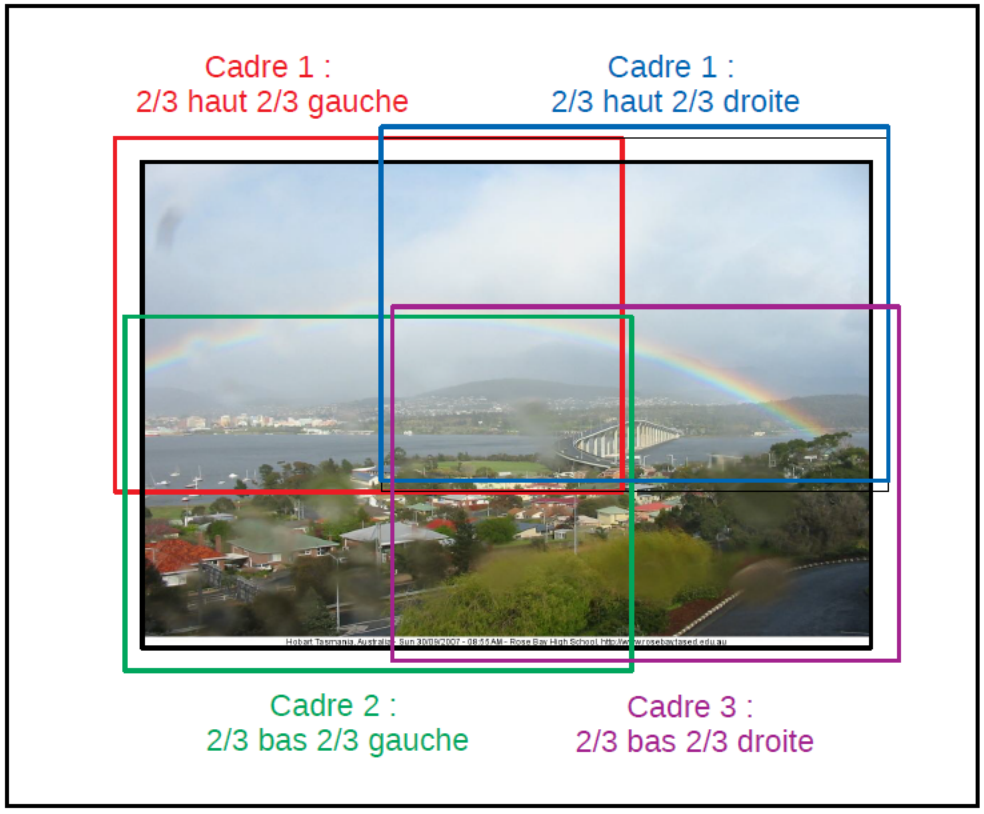


FIGURE A-13 – Cadres pour l'augmentation de données.

Avant d'appliquer le réseau nous recadrons chaque image suivant un patron prédéfini. Pour des images de largeur comprise entre 512 et 1024, nous définissons par exemple cinq cadres (voir A-13). Le cinquième cadre correspond aux dimensions d'origine. Pour ces scènes, le modèle est donc appliqué cinq fois à toutes les images. Cela permet de d'augementer la donnée par rognage sans sous-estimer les tailles des intervalles cibles.

Sur les séries de prédictions, nous avons retrouvées les cas d'erreur systématiques décrits dans la section 4.2.7. Les prédictions ont dû être post-traitées pour limiter l'accumulation

des erreurs et ses conséquences sur l'estimation des fonctions de répartition.
En particulier, les erreurs liées à la présence résiduelle d'images de nuit et au passage régulier du soleil dans le champ de la caméra ont été prises en compte. On retrouvera ces étapes dans le code (annexe numérique).
Nous avons utilisé les sorties post-traitées pour évaluer la fonction $\frac{\mathcal{R}^{s^-}_Z + \mathcal{R}^{s^+}_Z}{2}$ pour chaque série de prédictions . Ces quantités définissent les cibles de l'apprentissage.

Contrairement aux entraînements de la partie précédente, il ne s'agit pas d'apprentissage par paires. Les mini-lots sont formées à partir d'images simples (64 images). Nous avons utilisé l'erreur moyenne absolue (MAE) comme fonction de coût pour prévenir l'effet des valeurs extrêmes.
Au cours de l'entraînement, les séquences et les cadres sont tirées au hasard (tirage équiprobable). Au sein d'une même séquence, le tirage des images est pondéré de façon à privilégier les faibles visibilités (équilibrage). Après recadrage, les opérations d'augmentation de données autres que le rognage (transformation perspective, réflexion, rotation, etc) sont appliquées sans perte d'information.

Annexe E

Compléments sur le chapitre 5

E.1 Compléments sur AMOSssExt

Dans cette section, nous décrivons les grandes lignes de la construction du jeu AMOSssExt à partir du classifieur ss_pl_du11.1.

Cette construction a été réalisée suivant trois grandes étapes. La première étape a consisté à rassembler des séquences homogènes d'au plus 400 images sur lesquelles au moins un événement est détecté. Le classifieur est appliqué pendant la seconde étape sur toutes les paires d'images de chaque séquence (au plus 160.000 paires). On déduit des matrices de comparaison résultantes un ordre d'intervalle sur les quéquences par optimisation (algorithme varIO). Enfin, on débruite les séries résultantes par application de deux critères physiques (lenteur des variations et pas de croisssance possible par beau temps).

Première étape : séquences avec événements de neige

Dans un premier, nous avons raccordé les séquences d'AMOSvvExt de façon à disposer de séquence plus longues, avec davantage de niveaux d'enneigement différents. Les caméras de validation et de test sont écartées. Nous ramenons enfin les séquences les plus longues à une taille de 400 images. Cette opération est faite en prenant au moins 50 % d'images associées à un épisode d'enneigement (ces épisodes sont cherchés de la même façon que les épisodes de basse visibilité dans AMOSvvExt).

Deuxième étape : prédiction d'un ordre d'intervalle par optimisation

Nous utilisons ensuite le classifieur sur chaque séquence pour obtenir une matrice de comparaison (la sortie du classifieur sur la paire (x_i, x_j) définit l'intersection de la ligne i et de la colonne j). Cette matrice est symétrisée puis utilisée comme entrée de l'algorithme varIO (encadré D.1). Cet algorithme fournit un ordre d'intervalle pour chaque séquence. [1]

1. Nous n'avons jamais tenté de comparer cet ordre d'intervalle à une prédiction par fonction de rang sur le jeu de test. A posteriori, l'expérience aurait été intéressante.

En général, les ordres d'intervalles prédits sont trop prudents. Mais ils suivent bien l'enneigement et les erreurs sont rares. Pour limiter les effets de cet excès de prudence, nous n'avons conservé dans $\mathcal{U}_2^s$ que des arêtes qui sont prédites incomparables à la fois par varIO et par le classifieur.

Troisième étape : correction des prédictions

Nous avons aussi repéré certaines fautes de prudence facilement correctibles. D'une part, des variations opposées sur trois images d'affilée dans une période de 2 h (par exemple $x_i \prec x_{i+1}$ et $x_{i+2} \prec x_{i+1}$, sont très rares : le manteau varie trop lentement. Ces arêtes sont donc considérées comme erronées et placées dans $\mathcal{U}_2^s$. De même lorsqu'une croissance est prédite alors que le modèle vv_sl_due111.0 ne prédit pas d'épisode de basse visibilité.