Suivi multi-objets basé sur des tracklets dans un réseau de caméras

Yosra Dorai

École doctorale n° XXX : Sciences Pour ingénieur et Ecole doctorale de Sousse

Université de Clermont-Auvergne et Université de Sousse

THÈSE

pour obtenir le grade de docteur délivré par les écoles doctorales
Des Sciences Pour l'Ingénieur
Spécialité: "Électronique et système"

présentée et soutenue publiquement par

Yosra DORAI

le 25/06/2021

Suivi multi-objets basé sur des tracklets dans un réseau de caméras

Directeur de thèse : **Najoua ESSOUKRI BEN AMARA**
Directeur de thèse : **Frédéric CHAUSSE**
Co-encadrant de thèse : **Sami GAZZAH**

Jury

M. Abdelmalik TALEB-AHMED ,	Université Polytechnique des Hauts-de-France	Rapporteur
M. Chokri BEN AMAR ,	Ecole Nationale d'Ingénieurs de Sfax-ENIS	Rapporteur
Mme Fatma Ezzahra SAYADI,	Ecole Nationale d'Ingénieurs de Sousse-ENISo	Examinateur
M. Thierry CHATEAU,	Université Clermont Auvergne	Examinateur

LATIS-Laboratory of Advanced Technology and Intelligent Systems
Université de Sousse, École Nationale d'Ingénieurs de Sousse, Sousse 4023, Tunisia
Institut Pascal
Universite Clermont Auvergne, CNRS, SIGMA Clermont, F-63000 CLERMONT-FERRAND

THESE

Résumés

Résumé

Aujourd'hui, les caméras envahissent notre vie, elles sont de plus en plus utilisées dans plusieurs domaines et elles sont installées partout dans des milieux publics et privés, et plus particulièrement dans le contexte de vidéo-surveillance afin d'identifier des personnes ou des véhicules pas seulement par une caméra mais par le réseau entier. L'exploitation des données provenant d'un réseau de caméras devient une nécessité de nos jours pour faire face à des problèmes de sécurité ou même de simple contrôle. Ces systèmes de vision qui peuvent être basés sur des algorithmes d'intelligence artificielle présentent des défis d'actualité.

Cette thèse s'inscrit dans le contexte du suivi multi-objets et s'intéresse à la réidentification dans un réseau de caméras. Cette dernière consiste à déterminer la position d'un objet par rapport au champ de vue de chaque caméra. Ce défi devient particulièrement difficile face au changement de l'apparence de l'objet d'une caméra à une autre, la variation de luminosité, l'angle de vue... L'objectif de cette thèse est de proposer une solution fiable afin de réidentifier des objets dans un réseau de caméras assurant une robustesse aux différentes complexités de la réidentification.

Dans ce cadre, nous proposons d'exploiter l'historique de l'objet détecté et former un ensemble de nœuds de détection que nous avons appelé « tracklet ». Il s'agit d'une nouvelle approche inspirée des différents travaux de l'état d'art. Nos contributions basées sur des tracklets proposées durant cette thèse touche deux phases : le suivi dans une caméra et la réidentification dans un réseau de caméras.

Afin d'optimiser les performances des algorithmes de suivi et de réidentification, il est nécessaire de disposer d'un système de détection fiable. Dans la littérature, les réseaux de neurones convolutifs (CNN) ont eu beaucoup de succès dans la détection multi-objets. Nous avons utilisé une méthode d'apprentissage profond afin de détecter des objets. Bien que cette méthode présente un taux de détection élevé, elle présente aussi un nombre de faux positifs et des faux négatifs surtout lorsque la base de test est différente de la base d'apprentissage. Ce qui nous amené à proposer notre originalité de suivi à base de tracklet qui permet de corriger les défauts de détection et améliorer la qualité de suivi. En fait, notre stratégie se base d'abord sur la construction des tracklets à bases des détections par une comparaison de signature, puis de construire des trajectoires à partir de l'association des tracklets. Cependant, ces trajectoires peuvent présenter des coupures dues à la non-détection. Une étape de mise à jour permet de les combler grâce à un processus d'interpolation qui reconstitue les objets non détectés.

Quant à la réidentification, nos contributions se manifestent d'une part dans l'augmentation du volume des données d'apprentissage. En fait, le réseau de neurones qui effectue la réidentification d'une trajectoire dans chaque caméra nécessite, pour son entraînement, un volume important de données. Il se peut que pour certaines caméras ce volume soit trop faible d'où la nécessité de régénérer d'autres données. Notre contribution est de générer des nouveaux échantillons à partir des tracklets détectées dans une autre caméra par un auto-encodeur (GAN). Ce qui nous permet de ne transférer que des vrais positifs d'une façon automatique sans vérification. Au niveau du détecteur dans la ré-identification, les objets sont décrits par parties, ce qui permet de les reconnaître par la suite même s'ils n'apparaissent pas complètement dans une autre caméra. D'autre part, notre contribution se manifeste dans la comparaison des tracklets issues des différentes caméras du réseau.

Les améliorations proposées ont été évaluées sur des bases publiques et privées. Les résultats atteints par nos approches montrent des performances comparables à celles des systèmes existants.

Mots clés : Suivi, tracklet, Ré-identification, Apparence, Description globale, Systèmes de transport intelligents, Systèmes de vision, Vision par ordinateur, Intelligence artificielle, Traitement d'images, Classification, Réseaux de neurones convolutifs, Apprentissage profond, LSTM

Abstract

Today, cameras invade our life, they are more and more used in several fields and they are installed everywhere in public and private environments. Particularly, in the context of video surveillance in order to identify people or vehicles not only by a camera but by the network. Data exploitation from a camera network becomes a necessity nowadays to face security problems or even simple inspection. These vision systems, which can be based on artificial intelligence algorithms, present actual challenges.

This PhD deals with multi-object tracking and focuses on re-identification in a network of cameras. The challenge is to determine the position of an object relative to the field of view of each camera. This challenge becomes particularly complex due to the change of object appearance from one camera to another, the variation of luminosity, the angle of view... The objective of this work is to propose a reliable solution to re-identify objects in a network of cameras ensuring a robustness to the different complexities of re-identification.

In this context, we propose to exploit the history of the detected object and to form a set of detection nodes which we called "tracklet". This is a new approach inspired by various works in the state of the art. Our contributions based on tracklets proposed during this PhD covers two phases : tracking in a camera and re-identification in a network of cameras.

In order to optimize the performance of tracking and reidentification algorithms, a trusted detection system is needed. In the literature, convolutional neural networks (CNNs) have been very successful in multi-object detection. We used a deep learning method to detect objects. Although this method has a high detection rate, it also has a number of false positives and false negatives especially when the test base is different from the training base. This led us to propose our original tracklet-based tracking method which allows to correct the detection defects and improve the tracking quality. In fact, our strategy is based first on the construction of tracklets based on detections by a signature comparison, then to build trajectories from the association of tracklets. However, these trajectories can present breaks due to non-detections. An update step allows to fill them thanks to an interpolation process that reconstitutes the non-detected objects.

As for the reidentification, our contributions are manifested on the one hand in the increase of the volume of training data. In fact, the neural network that performs the reidentification of a trajectory in each camera requires, for its training, a large volume of data. It is possible that for some cameras this volume is too small, hence the need to regenerate other data. Our contribution is to generate new samples from tracklets detected in another camera by an auto-encoder (GAN). This allows us to transfer only true positives in an automatic way without verification. At the level of the detector in the re-identification, the objects are described by parts, which makes it possible to recognize them afterwards even if they do not appear completely in another camera. On the other hand, our contribution is manifested in the comparison of tracklets from the different cameras of the network.

The proposed improvements have been evaluated on a public basis. The results achieved by our approaches show performances comparable to those of existing systems.

Keywords : Tracking, Tracklet, Re-identification, Appearance, Global description, Intelligent transport systems, Vision systems, Computer vision, Artificial intelligence, Image processing, Classification, Convolutional neural networks, Deep learning, LSTM

Remerciements

Les années consacrées pour cette thèse me manque déjà sur le plan professionnel ainsi que humain.

Cette simple partie ne me permettra sûrement pas de remercier suffisamment tous ceux qui ont joué un rôle majeur durant ce long parcours.

Mes remerciements s'adressent tout d'abord aux membres du jury pour avoir accepté d'évaluer ce travail.

En particulier, je souhaite vivement remercier toutes les personnes qui ont contribué au succès de cette thèse et qui m'ont aidée lors de la rédaction.

Je tiens à témoigner toute ma reconnaissance aux personnes suivantes, pour leur aide dans la réalisation de cette thèse :

Madame Najoua ESSOUKRI BEN AMARA qui a encadré ma thèse. Elle a été d'un grand soutien dans l'élaboration de ma thèse et je la remercie aussi pour sa grande disponibilité, pour sa bienveillance et surtout ses judicieux conseils.

Je voudrais remercier également, mon directeur de thèse Monsieur Frédéric CHAUSSE pour sa patience, sa disponibilité qui ont contribué à alimenter ma réflexion.

Monsieur Sami GAZZAH qui a co-encadré cette thèse. je le remercie de m'avoir accordé sa confiance et une large indépendance dans l'exécution de missions valorisantes.

Je remercie également toute l'équipe pédagogique de l'université de Clermont-Auvergne (France) et l'université de Sousse (Tunisie) et les intervenants professionnels.

Je tiens aussi à remercier mes collègues dans les deux laboratoires Institut Pascal (France) et LATIS (Tunisie) pour le soutien qu'ils m'ont apporté et les moments (agréables et difficiles) partagés durant ma thèse.

Yosra

À mon père, ma mère, mon frères, ma sœurs, ma nièce et mon neveu
A mon mari (la cerise sur le gâteau)

Table des matières

Résumé iii

Remerciements i

Table des matières iii

Liste des figures v

Liste des tableaux ix

1 Introduction 1
- 1.1 Introduction . . . 2
- 1.2 Objectifs . . . 3
  - 1.2.1 La détection . . . 3
  - 1.2.2 Le suivi . . . 4
  - 1.2.3 La ré-identification . . . 5
- 1.3 Scénario d'étude . . . 6
- 1.4 Contributions . . . 7
  - 1.4.1 Contributions au niveau suivi . . . 7
  - 1.4.2 Contributions au niveau ré-identification . . . 8
- 1.5 Structure du document . . . 8

2 Etat de l'art : suivi et ré-identification 11
- 2.1 Introduction . . . 12
- 2.2 Suivi dans une caméra . . . 12
  - 2.2.1 Détection . . . 15
  - 2.2.2 Suivi . . . 17
  - 2.2.3 Discussion et positionnement au niveau suivi . . . 21
  - 2.2.4 Évaluation des méthodes de suivi . . . 24
- 2.3 Suivi dans un réseau de caméras . . . 27
  - 2.3.1 Ré-identification . . . 28
  - 2.3.2 Discussion et positionnement au niveau de la ré-identification . . . 35
  - 2.3.3 Évaluation des méthodes de ré-identification . . . 35
- 2.4 Conclusion . . . 36

3 Le suivi mono-caméra 37
- 3.1 Introduction . . . 38
- 3.2 Vue globale de la méthode de suivi . . . 38
- 3.3 Détection et extraction des caractéristiques . . . 40
  - 3.3.1 Réseau de neurones convolutif . . . 40
  - 3.3.2 Définition de la signature . . . 41

3.4 Le suivi . . . 44
3.4.1 Définition d'une tracklet . . . 45
3.4.2 Construction des tracklets . . . 45
3.4.3 Le stage global . . . 47
3.4.4 Étape de mise à jour . . . 47
3.4.5 Minimisation de l'architecture par RNN . . . 48
3.5 Expérimentation . . . 61
3.5.1 Base de données . . . 61
3.5.2 Métriques . . . 63
3.5.3 Résultats et discussions . . . 63
3.5.4 Analyse des erreurs . . . 67
3.6 Conclusion . . . 70

4 La ré-identification dans un réseau de caméras 71
4.1 Introduction . . . 72
4.2 La ré-identification des piétons dans un réseau de caméras . . . 72
4.3 Transfert de tracklets entre n caméras . . . 73
4.3.1 Principe du CycleGAN . . . 73
4.3.2 Transfert des tracklets . . . 75
4.4 Vue globale de l'architecture de ré-identification dans un réseau de caméras 77
4.4.1 Extraction des caractéristiques . . . 78
4.4.2 Extraction de l'information temporelle par LSTM . . . 79
4.4.3 Comparaison et décision . . . 80
4.5 Approche de ré-identification dans des conditions particulières . . . 81
4.6 Expérimentation . . . 84
4.6.1 Base de données . . . 85
4.6.2 Métrique . . . 88
4.6.3 Résultats et discussions . . . 88
4.7 Conclusion . . . 95

5 Conclusion et perspectives 97
5.1 Résumé des contributions . . . 98
5.2 Limites et perspectives . . . 99

Liste des figures

1.1 La vidéo-surveillance . . . 2
1.2 Une occultation partielle a eu lieu entre la personne dans le rectangle rouge et un objet de l'arrière plan. . . . 4
1.3 Exemple de variabilité inter-classe . . . 4
1.4 Le suivi dans n caméras conduit à la ré-identification dans le réseau de caméras . . . 6
1.5 Les différentes étapes du suivi dans le réseau de caméras . . . 7

2.1 Le suivi par détection, conteur et point d'intérêt . . . 13
2.2 Le calcul de l'histogramme des arêtes [KCM04] . . . 14
2.3 La composition d'un détecteur . . . 16
2.4 Une classification des algorithmes de suivi . . . 19
2.5 Exemple d'image des bases PETS S2L1 et PETS S2L2 . . . 25
2.6 Calcul des métriques Recall et Precision . . . 27
2.7 Un exemple de modèle de réseau profond . . . 28
2.8 Une architecture exemple de réseau Siamois triplet . . . 31
2.9 Une architecture de comparaison par paire d'images de [WZL+16] . . . 32

3.1 Vue globale de l'architecture de suivi dans une mono-caméra . . . 38
3.2 Architecture du Faster R-CNN [Gir15] . . . 41
3.3 Une représentation 3D de l'espace de couleurs HSV . . . 43
3.4 Un exemple de transformation de l'espace couleur RGB en HSV . . . 43
3.5 Vue globale de l'architecture de suivi dans une mono-caméra . . . 44
3.6 L'ensemble des nœuds constitue une tracklet . . . 45
3.7 Correction d'un objet non détecté par une interpolation . . . 48
3.8 La propagation de l'information par un passe-avant d'une couche i du RNN 49
3.9 La rétro-propagation du gradient par un passe-arrière d'une couche i du RNN 50
3.10 La différence de conservation d'information entre une couche de RNN et LSTM . . . 52
3.11 L'interaction entre les portes et les vannes qui contrôlent la propagation en passe-avant de l'information provenant de la mémoire interne, de l'entrée et de la sortie par le LSTM . . . 53
3.12 Visualisation de la propagation du gradient lors d'un passe-avant dans une couche LSTM . . . 54
3.13 Visualisation de la rétro-propagation du gradient lors du passe-arrière dans une couche LSTM . . . 58
3.14 Architecture de suivi mono-caméra . . . 59
3.15 Une unité de LSTM . . . 59
3.16 Le module de prédiction . . . 60
3.17 Le module de mise à jour . . . 60

3.18 Exemple de frame de la séquence S2L1 62
3.19 Exemple de frame de la séquence S2L2 62
3.20 Exemple de frame de la base ETHMS 62
3.21 Exemple de frame de la base LATIS-IP 63
3.22 Elimination des faux positifs issus de la détection par la méthode de suivi . 64
3.23 Les résultats de suivi sur la base PETS2009 S2L1 : (a)Frame 156 : le suivi est fait sans circonstances particulières; (b)Frame 168 : Croisement entre les piétons 7,3 et 1; (c)Frame 173 : Occultation entre le piéton 7 et un objet de l'arrière-plan; (d)Frame 185 : les piétons ont pu garder leur ids malgré l'occultation et le croisement 66
3.24 Les résultats de suivi sur la base PETS2009 S2L2 : (a)Frame 67; (b)Frame 95 67
3.25 Notre approche de suivi est validée par la base ETHMS (Sunny and Bahnhof) : (a)Frame 17 : La détection et le suivi des 3 premiers piétons apparaissant dans la vidéo. (b)Frame 19 : Suivi réussi malgré l'occultation entre l'objet 3 en bleu et 2 en vert. (c)Frame 70 : La poursuite du piéton 3 en présence de l'arbre. (d)Frame 90 : Ré-identification du piéton 3 après la disparition derrière l'arbre. (e)Frame 121 : Différentes tailles de piétons poursuivis avec succès. (f)Frame 216 : La dernière apparition du piéton 3 qui s'est montré pour la première fois dans la frame 1. (g)Frame 312 : Réussir à suivre des piétons apparaissant pour la première fois. (h)Frame 315 : Suivi de différents piétons avec différents angles de vue 68
3.26 Le suivi dans des conditions météorologiques dégradées 69
3.27 Une frame qui illustre le suivi des véhicules dans la base CIF 69
3.28 Le suivi des différents objets de la base "INDIA" 69

4.1 Notre architecture de ré-identification 73
4.2 La translation d'une image de domaine X à une autre image de domaine Y et l'inverse 74
4.3 L'architecture du discriminateur qui prend la décision après la reconstruction 74
4.4 L'architecture d'un générateur 75
4.5 La reconstruction des tracklets d'une caméra à une autre 76
4.6 La ré-identification dans un réseau de caméras 76
4.7 La ré-identification dans deux moments différents 77
4.8 Les tracklets sont l'entrée principale de l'architecture de la ré-identification 77
4.9 La comparaison se fait entre les tracklets issues de chaque caméra 78
4.10 La reconstruction du vecteur de caractéristiques 78
4.11 Une comparaison entre un bloc de réseau de neurones et un bloc de réseau de neurones résiduel 79
4.12 Une unité de LSTM 80
4.13 L'effet des conditions météorologiques sur l'apparence d'un objet 82
4.14 Exemple d'image de la base LATIS-IP 83
4.15 Exemple d'image de la base PRID 2011 85
4.16 Exemple d'image de la base Market-1501 86
4.17 Exemple d'image des personnes de la base de VIPeR 86
4.18 Exemple d'image de la base i-LIDS 87
4.19 Exemple d'image de notre sous-base 1 87
4.20 Exemple d'image de notre sous-base 2 88
4.21 Exemple de courbe de correspondance cumulée CMC 88
4.22 La dégradation des performance au-delà de 6 répartitions 90

4.23 Comparaison de nos approches par la courbe CMC appliquée à la base VIPeR 93
4.24 Comparaison de nos approches par la courbe CMC appliquée à la base PRID-2011 94
4.25 Comparaison de nos approches par la courbe CMC appliquée à la base i-LIDS 94

Liste des tableaux

2.1 Comparaison des méthodes de suivi 16
2.2 Classification des méthodes de détection selon leur taux de faux positifs . . 18
2.3 Comparaison des méthodes d'association 22
2.4 Comparaison des méthodes de ré-identification par des réseaux profonds DNN 30
2.5 Comparaison des méthodes de classification 34

3.1 Les moments statistiques et leurs formules de calcul 42
3.2 Les bases de données utilisées dans le suivi et leurs caractéristiques 61
3.3 Les taux de faux positifs avant et après le suivi 64
3.4 Comparaison de nos approches appliquées à la base PETS2009 S2L1 par les métriques MOTA, MOTP, Precision, Recall et FP 65
3.5 Comparaison de nos approches appliquées à la base PETS2009 S2L2 par les métriques MOTA, MOTP, Precision, Recall et FP 65
3.6 Comparaison de nos approches appliquées à la base ETHMS par les métriques MOTA, MOTP, Precision, Recall et FP 67

4.1 Les bases de données utilisées et leurs caractéristiques 85
4.2 Comparaison de notre approche avec une description globale et avec répartition des personnes par la courbe CMC (%) appliquée aux bases de données PRID-2011 et VIPeR 89
4.3 L'influence du nombre de répartition p sur la valeur CCR (%) appliquée à la base PRID-2011 et VIPeR. 90
4.4 Comparaison entre le transfert des tracklets et celui des échantillons sur la base LATIS-IP et Market-1501 selon la métrique CMC (%) 91
4.5 Comparaison de notre méthode avec d'autres travaux de l'état de l'art appliqués à la base de données VIPeR et utilisant la métrique CMC (%) 91
4.6 Une sélection de quelques travaux de l'état de l'art et leurs caractéristiques 92
4.7 Comparaison de notre approche avec une sélection de travaux de l'état de l'art appliquée sur les bases PRID 2011 et i-LIDS selon la métrique CMC (%) 93
4.8 Comparaison de notre approche avec l'état de l'art selon la métrique CMC (%) sur la base Market-1501 95

Chapitre 1

Introduction

« *...c'est en prenant conscience que les choses sont éphémères qu'on les apprécie à leur juste valeur et qu'on a envie d'en savourer chaque minute."* »

Gemma Malley

Sommaire

1.1 Introduction **2**
1.2 Objectifs **3**
1.2.1 La détection 3
1.2.2 Le suivi 4
1.2.3 La ré-identification 5
1.3 Scénario d'étude **6**
1.4 Contributions **7**
1.4.1 Contributions au niveau suivi 7
1.4.2 Contributions au niveau ré-identification 8
1.5 Structure du document **8**

1.1 Introduction

La vision par ordinateur a pour objectif l'analyse et l'interprétation d'une scène à partir du traitement d'images issues d'une ou plusieurs caméras. Elle permet de détecter, d'identifier ou de suivre un ou plusieurs objets contenus dans les images. La vidéo-surveillance, la robotique, la réalité augmentée ou l'imagerie médicale sont des applications importantes de ce domaine.

Dans l'exemple de la vidéo-surveillance, l'objectif est de détecter des situations problématiques. Initialement, ce sont des opérateurs qui analysent visuellement les images et réagissent à ce qu'elles renvoient. Ils analysent simultanément des séquences d'images provenant de plusieurs caméras mises en réseau (cf. figure 1.1). L'utilisation de la vision par ordinateur dans un tel contexte a pour but d'automatiser ce type de tâche. D'autre part, les résultats de l'analyse permettent d'apporter des solutions à des problèmes de sécurité, à ceux révélés par des statistiques de fréquentation routière, etc ... Ces applications sont si nombreuses que ce domaine intéresse particulièrement la communauté scientifique depuis plusieurs décennies.

Plus récemment, la multiplication des plateformes mobiles (smartphones, tablettes, drones, etc ...) qui embarquent forcément au moins une caméra ont donné accès à d'autres champs d'application où le partage d'informations est primordial. Des contraintes matérielles plus importantes que celles imposées par les caméras statiques sont apparues.

La problématique du suivi d'objet occupe une place particulière dans ces applications car elle nécessite d'implémenter en premier lieu une fonction de détection/identification pour ensuite intégrer les aspects dynamiques de la scène et pour ainsi dire de l'apparence des objets dans les images. Le suivi a pour but de décider si deux observations d'objets dans deux images capturées à deux instants différents correspondent ou non au même objet de la scène. Il s'agit donc d'un processus d'association de données. Ce problème se complique dans le cas où plusieurs caméras délivrent chacune une séquence d'images de la même scène vue à partir de positions très différentes et avec parfois des champs de vision totalement ou partiellement non recouvrant. Il s'agit donc de décider automatiquement si un objet identifié est suivi dans une séquence et s'il est le même, ou non, que celui qui apparaît ultérieurement dans une autre séquence. La problématique prend alors le nom de ré-identification.

Cette thèse intitulée "suivi multi-objets basé sur des tracklets dans un réseau de caméras" s'inscrit dans cette problématique. Elle aborde le cas de la détection/identification/suivi dans une séquence d'images prise par une mono-caméra et ensuite la ré-identification dans les séquences provenant de caméras mises en réseau. Elle a été préparée dans le

FIGURE 1.1 – La vidéo-surveillance

cadre d'une convention de cotutelle entre l'Institut Pascal de l'Université Clermont Auvergne (France) et le laboratoire LATIS (Laboratory of Advanced Technology and Intelligent Systems) de l'Université de Sousse (Tunisie).

1.2 Objectifs

Pour éviter tout abus de langage, ou plutôt pour faciliter l'emploi, la communauté scientifique remplace l'expression "traitement d'une séquence d'images issues d'une caméra" par "traitement dans une caméra" que ce traitement soit de la détection ou du suivi etc.

L'objectif de cette thèse est de proposer un cadre pour le suivi d'objets dans une seule caméra pour le faire ensuite évoluer afin de ré-identifier des objets dans un réseau de caméras. Pour se faire, les fonctionnalités suivantes ont été jugées comme nécessaires :

— la détection,

— le suivi,

— la ré-identification dans un réseau de caméras.

1.2.1 La détection

La détection est la première étape d'un processus de suivi et de ré-identification. Son objectif est de localiser, s'il existe, l'objet d'intérêt dans une image. La sortie se présente, en général, sous forme d'un rectangle englobant l'objet, tracé sur l'image en fonction de coordonnées 2D déterminées par l'algorithme de détection. La détection est à elle seule une problématique scientifique individuelle pour laquelle de très nombreuses contributions sont proposées chaque année. La détection et le comptage du flux d'objets dans le trafic urbain ou la localisation d'obstacles (piétons, véhicules, ...etc.) dans les systèmes ADAS (Advanced Driver Assistance System) ou encore la protection des zones sensibles sont des exemples d'applications de la détection d'objets.

Néanmoins, les performances des systèmes disponibles doivent être améliorées pour créer des systèmes donnant des résultats plus satisfaisants. Les difficultés proviennent de :

— l'occultation totale ou partielle des objets au cas où l'objet n'apparaît pas totalement dans le champ de vision de la caméra (cf. figure 1.2). Ce phénomène peut être engendré par des croisements entre des objets de même classe ou de classes différentes ainsi que par le positionnement des objets à détecter en arrière-plan. Dans ce cas, la détection de l'objet est rendue difficile puisque la majorité de ses caractéristiques révélées par l'image sont cachées.

— le taux de luminosité et le rendu flou résultent du mauvais réglage des paramètres optiques des caméras et de la grande variabilité des conditions d'éclairage, en particulier dans le cas de scènes extérieures.

— la variation inter-classe : dans une même classe d'objet, on trouve une grande variabilité d'apparence. L'exemple de la classe des véhicules deux roues (moto) (Cf. figure 1.3) est assez typique dans ce genre de problème.

Dans cette thèse, la solution retenue pour la détection est d'utiliser une méthode de classification d'objets par un réseau de neurone convolutif préalablement entraîné. Cette méthode est décrite en détails dans le chapitre 3, section 3.3.

FIGURE 1.2 – Une occultation partielle a eu lieu entre la personne dans le rectangle rouge et un objet de l'arrière plan.

FIGURE 1.3 – Exemple de variabilité inter-classe

1.2.2 Le suivi

L'association des objets détectés dans une séquence d'images conduit à construire la trajectoire, suivie par chacun d'eux, dans le repère 2D de l'image. Chacune des instances de l'objet, détectées et combinées au fil du temps, constitue un **nœud** de la trajectoire. Le suivi peut servir au contrôle du trafic routier, à l'analyse des comportements ou à l'usage des robots compagnons dans le cadre du système d'aide aux personnes âgées, à titre d'exemple.

Il existe de nombreuses solutions pour le suivi d'objets. Elles font face aux difficultés suivantes :

— Les paramètres de la caméra : le suivi est influencé par le capteur d'acquisition. En fait, la basse résolution, la distorsion des couleurs, la faible cadence de l'image, la lenteur de la dynamique par pixels ainsi que le bruit affectent les performances du suivi.

— L'objet : l'objet lui-même peut dégrader les performances du suivi. Le nombre d'objets dans la scène est un facteur de complexité pour le suivi : les scènes renfermant des foules sont pour ainsi dire les plus complexes. La taille de l'objet dans l'image varie selon la distance qui le sépare de la caméra. Les objets éloignés sont de petite taille et par conséquent plus difficiles à suivre. Le croisement des trajectoires et la non-rigidité de certains objets augmentent encore la difficulté.

— L'environnement : l'occultation affecte le suivi. Elle n'est pas causée par les objets eux-mêmes mais plutôt par l'environnement. La variation de l'éclairage provoque des différences d'apparence si nettes qu'elles peuvent considérablement gêner le

suivi.

— Les contraintes applicatives : le cahier des charges de l'application (le temps-réel, la sûreté du fonctionnement, etc...)

La principale conséquence est la fragmentation des trajectoires. Un même objet est correctement suivi sur quelques images puis perdu de façon à ce que la construction de sa trajectoire soit interrompue. Une nouvelle trajectoire est un peu plus tard construite à partir de ce même objet mais l'algorithme n'associe pas les deux. Chaque fragment de trajectoire est appelé **tracklet**.

Dans cette thèse, l'association des données est réalisée par le calcul d'un score de similarité entre les caractéristiques d'apparence et la cinématique des objets détectés. C'est ainsi que, pour une séquence d'images, un ensemble de tracklets est d'abord constitué. Dans un second passage, une signature caractéristique de l'objet est déterminée pour chaque tracklet. Les signatures sont ensuite comparées pour associer les tracklets.

Ce processus est expliqué dans le chapitre 3, dans la section 3.4.

1.2.3 La ré-identification

La ré-identification dans un réseau de caméras est envisageable quand un objet disparaît de l'une et apparaît dans l'autre. Les informations fournies par une caméra et transmises à une autre doivent être robustes afin de faciliter la reconnaissance de l'objet. Le but est de localiser cet objet dans plusieurs images issues de plusieurs caméras. La ré-identification permet d'ouvrir d'autres horizons à diverses applications de surveillance et de contrôle telles que le suivi des voitures, celui des joueurs ou du ballon dans un match de foot, etc ...

Viennent ensuite les défauts de la détection et du suivi, lesquels influencent la qualité de la ré-identification. Les difficultés, qui sont encore plus importantes, sont causées par :

— la pose de l'objet : la variation de pose des objets, due par exemple au changement du sens de déplacement ou d'attitude (debout/assis) ou à la déformation d'un objet non rigide, laquelle conduit à des apparences différentes du même objet.

— la pose de la caméra : l'objet est observé par les caméras avec différents angles de vue. Les parties de l'objet peuvent apparaître dans une caméra et être cachées dans une autre.

— la variation d'apparence : un même objet peut changer d'apparence d'une caméra à une autre (changement des vêtements d'une personne, déformation d'un véhicule due à un accident, ...)

— l'hétérogénéité des caméras : toutes les caméras d'un réseau ne sont pas forcément identiques, ce qui augmente la variabilité de l'apparence d'un même objet.

Dans cette thèse, après une étape de pré-traitement, afin d'augmenter le volume des données, nous utilisons spécialement un auto-encodeur, en vue de reconstruire les tracklets issues de chaque caméra de façon automatique non supervisée. Puis, le modèle d'apparence, qui sera un réseau profond, décompose l'objet en parties dans le but de refléter les changements locaux qu'un objet peut subir dans le réseau. Ensuite, une extraction de l'information temporelle se fera avec un réseau récurent. La prise de décision aura lieu par un réseau de calcul de similarité. Cette méthode sera plus détaillée dans le chapitre 4, dans la section 4.2.

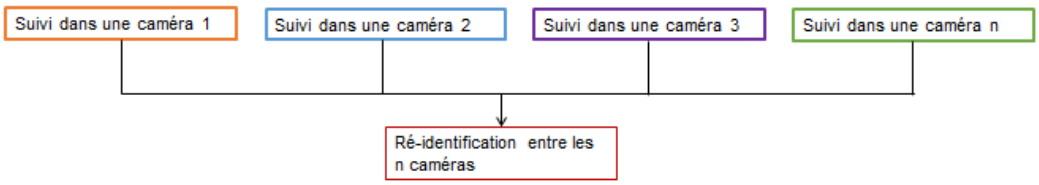


FIGURE 1.4 – Le suivi dans n caméras conduit à la ré-identification dans le réseau de caméras

1.3 Scénario d'étude

Cette thèse porte sur le suivi multi-objets basé sur des tracklets dans un réseau de caméras. Trois mots-clés sont pertinents : le suivi, les tracklets et le réseau de caméras. Par la suite, le lien entre ces trois concepts est détaillé.

Une stratégie est adaptée dans cette thèse afin d'atteindre le but de ré-identification dans un réseau de caméras et de faire le suivi dans une mono-caméra puis, étendre cette stratégie afin de faire le suivi dans le réseau de caméras. A ce niveau-là, il est nécessaire de donner une définition du mot "tracklet". En fait, une tracklet est une mini-trajectoire ; plus précisément, c'est l'association des détections sur une durée bien définie.

La problématique majeure du suivi d'après l'état de l'art est la fragmentation des trajectoires où le même objet, au lieu d'avoir une seule trajectoire, aura plusieurs mini trajectoires avec différents IDs. Ce découpage des trajectoires est dû au croisement entre les objets, aux occultations entre l'objet et l'environnement ou à d'autres facteurs. Dans cette thèse, ces fragmentations ne seront plus la problématique mais plutôt la solution. Ces mini-trajectoires appelées tracklets seront la première étape dans la chaîne de ré-identification. La première phase est la détection de l'objet dans la scène. Puis, une phase de construction des tracklets se manifeste par l'association de ces détections sur une durée bien déterminée. Une phase d'association de ces tracklets est appelée "stage global" où il y aura l'association des tracklets afin de permettre d'avoir des trajectoires complètes pour chaque objet depuis son apparition jusqu'à sa disparition dans une caméra.

A une échelle plus grande, les fragmentations (les tracklets) sont les trajectoires issues de chaque caméra. Le but est d'associer ces trajectoires afin d'identifier un objet et le ré-identifier dans les autres caméras. Par conséquent, le suivi dans le réseau de caméras s'opère implicitement. La figure 1.4 illustre que le suivi a eu lieu dans chaque caméra séparément avant la phase de ré-identification. La figure 1.5 met en valeur les étapes de suivi dans chaque caméra.

Le scénario visé par cette thèse est la ré-identification dans un réseau de caméras à l'aide des tracklets. A titre applicatif, la ré-identification pourra s'étendre sur une large zone avec des caméras positionnées de telle sorte que leur champ se couvre ou non pour la surveillance, pour le service de l'aide aux personnes ou pour des bâtiments intelligents. Les caméras peuvent être fixes ou mobiles. La topologie du réseau n'est pas prise en considération. Tout objet appartenant au trafic routier (véhicules ou piétons ...) est considéré comme un centre d'intérêt. Le suivi ou la ré-identification traite des scènes avec des foules au nombre important d'objets d'intérêt et où les occultations sont nombreuses et influencent la continuité des trajectoires. Les critères adaptés sont :

— **Au niveau mono-caméra :** le choix des techniques et des méthodes utilisées, pour faire le suivi dans une seule caméra, est basé essentiellement sur le compromis entre les bonnes performances et la résolution des problématiques liées à la détection et au suivi.

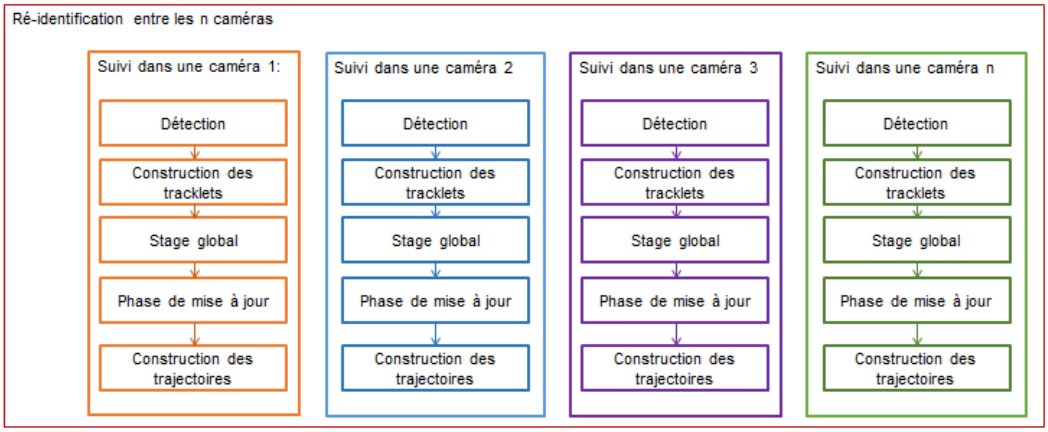


FIGURE 1.5 – Les différentes étapes du suivi dans le réseau de caméras

— **Au niveau multi-caméras :** le choix dans ce cas est basé sur des méthodes qui ne prennent pas en considération des problèmes de calibration spécifiques ni de topologies et qui peuvent garantir en contrepartie l'identifiant de chaque objet d'une caméra à une autre indépendamment de la quantité de l'information et du changement de l'environnement ou de pose dans le réseau de caméras. Des méthodes, qui facilitent la reconnaissance de l'objet, diminuent la complexité et ne prennent pas en considération les défauts de détection.

1.4 Contributions

Les contributions proposées dans cette thèse concernent le suivi et la ré-identification. Aucune contribution significative n'est présente dans l'étape de détection qui utilise une méthode de l'état de l'art disponible à l'époque du début de la thèse (2016).

1.4.1 Contributions au niveau suivi

Méthodologique :

— Le suivi utilise au départ les résultats de la détection. Ceux-ci sont imparfaits avec l'apparition de faux positifs ou des cas de non-détection. La stratégie de suivi proposée associe les détections pour former les tracklets. Les faux positifs ne sont naturellement pas associés, ce qui permet de les filtrer.

— Une trajectoire reconstituée à partir de l'association des tracklets peut présenter des "trous" dus à des non-détections. Une étape originale de mise à jour permet de les combler grâce à un processus d'interpolation qui reconstitue les objets non détectés. Il est à préciser qu'il n'y a pas à posteriori de vérification dans l'image de cette hypothèse d'interpolation.

— Le principe de l'association des signatures des tracklets gère aussi la naissance, la fin ou l'interruption d'une trajectoire (causée par exemple, d'une occultation ou d'un croisement). A cet effet, il n'est pas nécessaire d'ajouter de briques algorithmiques supplémentaires pour réaliser cette gestion.

Applicative : l'approche de suivi fonctionne de jour comme de nuit (avec éclairage public). Elle présente aussi de bonnes performances dans le cas de scènes avec un nombre important d'objets filmés par des caméras statiques ou mobiles.

1.4.2 Contributions au niveau ré-identification

Méthodologiques :

— Augmentation du volume des données d'apprentissage : le réseau de neurones qui effectue la ré-identification d'une trajectoire dans chaque caméra nécessite, pour son entraînement, un volume important de données (c'est-à-dire de tracklets). Il se peut que pour certaines caméras ce volume soit trop faible. La transformation d'apparence d'une classe d'objets entre deux caméras est effectuée par un réseau de neurones auto-encodeur (GAN). Ainsi, les tracklets de la base d'apprentissage de toute caméra du réseau peuvent être transcodées vers n'importe quelle autre caméra.

Il faut remarquer que le transfert se fait non pas par échantillons mais par tracklets et ce parce que ces dernières garantissent le fait de ne transférer, d'une caméra à une autre, que de vrais positifs.

— Description profonde et partielle des objets : la description des objets est faite par l'extraction des informations les plus profondes et pertinentes du réseau convolutif de l'étape de détection (carte des caractéristiques produites à partir des opérations successives des produits de convolution).

Au niveau du détecteur, les objets sont décrits par parties, ce qui permet de les reconnaître par la suite même s'ils n'apparaissent pas complètement en raison d'une occultation par exemple. Il est montré expérimentalement que cette approche de description par parties augmente les performances de ré-identification.

— Comparaison des tracklets issues des différentes caméras du réseau : la comparaison des tracklets permet de tenir compte explicitement de la cohérence temporelle de la trajectoire, ce qui n'est pas possible dans le cas de la comparaison d'objet échantillon (nœuds de la trajectoire). Ceci améliore la performance de la ré-identification.

Applicative : L'approche de ré-identification est robuste face à des conditions d'éclairage très variables. Ceci est prouvé lors de l'application de l'algorithme sur une base filmée pendant le jour et la nuit.

Puis, l'architecture est testée sur un nombre important de caméras dans un réseau afin de prouver que le changement de pose et l'apparition partielle des objets n'ont pas d'impact sur les performances de ré-identification de notre algorithme.

1.5 Structure du document

Cette thèse est structurée en deux parties, l'une pour le suivi dans une seule caméra et l'autre pour la ré-identification dans le réseau :

— Chapitre 1 : dans ce chapitre, une introduction générale est présentée, après laquelle l'objectif de la thèse ainsi que les facteurs de complexité qui limitent les performances sont précisés. Ces problématiques vont être détaillées sur trois niveaux (détection, suivi et ré-identification). Nous présentons également dans ce chapitre nos contributions au niveau de l'approche et de l'application pour les deux niveaux : suivi et ré-identification.

— Chapitre 2 : nous présentons dans ce chapitre les méthodes et les travaux issus de l'état de l'art au niveau du suivi. Nous y détaillons précisément les méthodes de suivi basées sur l'extraction des caractéristiques, sur les contours et les silhouettes

ainsi que sur la détection. Après une discussion et une comparaison de ces trois approches, nous présentons les différentes méthodes de détection et les stratégies possibles d'association de données dans un algorithme de suivi. Les méthodes d'évaluation, à partir de différentes bases de données publiques utilisées dans la littérature et les métriques, sont présentées par la suite. Le chapitre progresse avec une étude bibliographique de la problématique du suivi dans un réseau de caméras, c'est-à-dire de la ré-identification en se focalisant uniquement sur les méthodes basées sur des réseaux de neurones profonds. L'évaluation de ces méthodes sera ainsi exposée avec, de prime abord, la présentation des métriques, puis avec celle des différentes bases de données utilisées dans la plupart des travaux de l'état de l'art.

— Chapitre 3 : notre méthode de suivi est détaillée dans ce chapitre. Nous commençons par une vue globale de la méthode ; puis, nous traitons l'étape de détection et de l'extraction des caractéristiques afin de définir une signature pour chaque objet détecté. Dans la partie qui suit, notre approche de suivi est présentée. La notion de tracklets y est définie tout comme la manière de les construire. Ensuite, nous détaillons les différentes étapes d'association de tracklets et de correction de trajectoires. Nous proposons aussi dans ce chapitre une seconde approche dans laquelle la construction des tracklets, leur association et leur correction sont effectuées par un réseau récurrent. Dans la partie expérimentation, nous expliquons les différentes bases de données utilisées ainsi que les métriques. Les résultats sont discutés ultérieurement dans une partie où nous démontrons L'intérêt des tracklets sur les faux positifs ainsi que la validation de notre approche sur des bases statiques et mobiles.

— Chapitre 4 : nous y exposons l'approche de suivi étendue à des problématiques plus complexes liées à la ré-identification. Après une introduction qui illustre et définit la ré-identification, l'approche est détaillée, avec l'indication qu'il y aura une phase de pré-traitement et une architecture de ré-identification. Le pré-traitement est présenté sous forme d'une architecture de transfert de tracklets d'une caméra à une autre tout en présentant l'impact de cette phase sur la ré-identification. Dans l'étape suivante, nous développons le reste de l'architecture en commençant par l'extraction des caractéristiques liées à l'apparence et à l'information temporelle. Nous mettons en valeur la partie classification faite par un réseau de recherche de similarité entre les tracklets. Nous traitons aussi la ré-identification dans des conditions d'éclairage difficile avec le passage du jour à la nuit. Enfin, nous présentons l'étude expérimentale divisée en deux parties : la première met en valeur l'impact de nos contributions et la deuxième effectue la comparaison avec l'état de l'art. Nous clôturons ce chapitre par une analyse des erreurs.

— Chapitre 5 : nous présentons dans ce chapitre conclusif un résumé de nos contributions ainsi que des perspectives.

Chapitre 2

Etat de l'art : suivi et ré-identification

« *Le passé est de l'histoire, c'est au présent qu'il faut vivre* »

Henri-Frédéric Amiel

Sommaire

2.1 Introduction 12
2.2 Suivi dans une caméra 12
2.2.1 Détection 15
2.2.2 Suivi 17
2.2.3 Discussion et positionnement au niveau suivi 21
2.2.4 Évaluation des méthodes de suivi 24
2.3 Suivi dans un réseau de caméras 27
2.3.1 Ré-identification 28
2.3.2 Discussion et positionnement au niveau de la ré-identification . . . 35
2.3.3 Évaluation des méthodes de ré-identification 35
2.4 Conclusion 36

2.1 Introduction

Récemment, plusieurs modifications ont été apportées à l'intelligence artificielle et ce à plusieurs niveaux. Le suivi d'objets dans un réseau de caméras est classé parmi les applications les plus classiques et les plus importantes en vision par ordinateur. Il s'agit d'une thématique très active dans le domaine de la recherche qui présente plusieurs défis à relever : ceux liés à la détection, au suivi, à l'association des données, à la ré-identification, aux bases de données, à la classification, aux scores et à ceux liés aux métriques. Cette thématique est utilisée dans plusieurs applications liées à la robotique, à la vidéo-surveillance, aux assistances à la conduite et aux véhicules autonomes...

Cette revue de la littérature est axée sur deux grandes thématiques en relation directe avec la thèse : le suivi mono-caméra et la ré-identification dans un réseau de caméras. Dans la première section, une étude bibliographique est détaillée sur le suivi dans une caméra. Une présentation des différentes briques constituant la chaîne du suivi y est donnée. Une étude de l'évolution des détecteurs au cours des dernières années est présentée en premier lieu. Puis, les différentes méthodes de suivi sont mises en valeur par une étude bibliographique. Ensuite, dans les deux dernières sections de cette partie, les bases de données les plus utilisées dans le suivi mono-caméra et les métriques d'évaluation utilisées pour valider les résultats sont étudiées.

L'état de l'art en ré-identification est présenté dans la deuxième partie de ce chapitre avec une mise en relief des méthodes de ré-identification les plus utilisées dans les réseaux de caméras. Les bases et les métriques liées à cette thématique sont également étudiées avec leurs défis et leurs caractéristiques. Enfin, une conclusion clôture le chapitre.

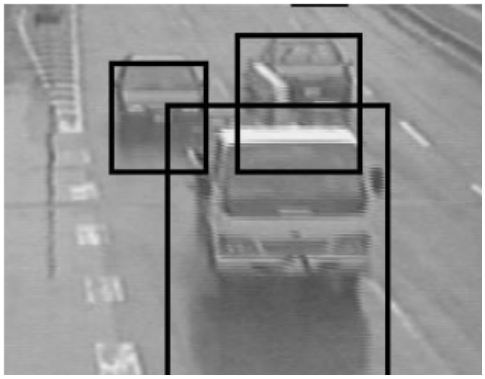
2.2 Suivi dans une caméra

Le suivi est une tâche complexe qui permet d'estimer des trajectoires $T = T_1, T_2, ..., T_n$ d'une ou plusieurs cibles $O = O_1, O_2, ..., O_n$. Ces cibles peuvent être des piétons, des voitures, des motos... Ces objets évoluent dans l'espace 2D de l'image. L'acquisition des données est faite par un capteur (camera, radar, sonar...) qui détermine des hypothèses sur la position $P = P_1, P_2, ..., P_n$ des objets dans la scène. Ces positions sont aussi appelées observations ou mesures. Les retours du capteur ne sont jamais parfaits ; les erreurs sont liées à l'imperfection des capteurs ou au bruit existant dans l'environnement. D'autres défauts du capteur peuvent se manifester par un objet non détecté. Les algorithmes de suivi ont pour objectif la construction des trajectoires des différents objets $T = T_1, T_2, ..., T_n$ à partir des observations issues des capteurs $P = P_1, P_2, ..., P_n$ au fil du temps.

Les algorithmes de suivi sont nombreux, nous en citons quelques exemples tels que le suivi basé sur des points caractéristiques, celui des compteurs et celui par détections. La figure 2.1 illustre les résultats d'application de ces méthodes sur différents objets.

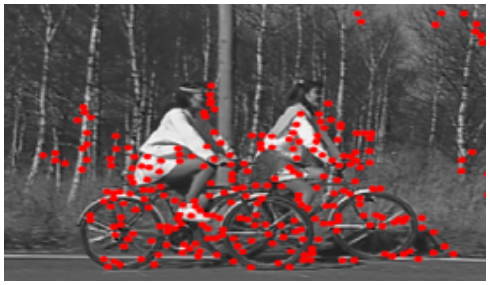
Le suivi par des points caractéristiques : Les méthodes de suivi basées sur les points caractéristiques représentent l'objet par des descripteurs locaux. Ces derniers sont calculés autour du point cible sur une petite région. Les points d'intérêt sont déterminés par des méthodes comme par exemple celle de flots optiques [WKSL11], SIFT (Scale-Invariant Feature Transform) [GBT07], SURF (Speeded Up Robust Features) [HYLL09] et FAST (Features from Accelerated Segment Test) [MHB+10]. Le suivi par la méthode des points caractéristiques, dépend des mesures de similarité entre l'objet détecté précédemment et les descripteurs locaux à l'état courant [GCT+14], [YLY13].

Par exemple, Yang et al. [YLY13] utilisent une méthode basée sur les points caractéristiques SIFT pour modéliser l'objet à suivre. Ils utilisent deux modèles d'apparence : le



(a) Le suivi par détection

(b) Le suivi par les conteurs



(c) Le suivi par détection

FIGURE 2.1 – Le suivi par détection, conteur et point d'intérêt

premier est un modèle de couleur et de texture alors que le deuxième repose sur la structure. Le premier utilise des patchs aléatoires établis sur les histogrammes de couleur RGB (Rouge, vert, bleu) et un descripteur LBP (Local Binary Patterns). Le deuxième, basé sur la structure, repose sur les points de SIFT et encode un histogramme spatial global. Les points d'intérêts détectés sur une région sont appairés avec la région détectée précédemment. Les points appairés constituent un ensemble qui présente l'histogramme spatial de l'objet. La similarité est calculée entre l'objet et la région par une comparaison entre les caractéristiques RGB et LBP locales et l'histogramme des points SIFT. Dans ce travail, l'algorithme de suivi exploite trois types de caractéristiques liées à la distribution des couleurs, à la texture et à la distribution géométrique globale. Le modèle de structure ne tient compte que des informations récentes car l'histogramme spatial ne prend en considération que les points appairés avec l'objet de la trame précédente.

En fait, cet inconvénient ne concerne pas le travail de Guo et al. [GCT+14] lequel essaye de mémoriser les points d'intérêts dans un modèle qui subit des mises à jour lors du suivi. La structure de l'objet est aussi représentée par un graphe qui contient le modèle d'apparence. Une description synthétique, qui tient compte de la variation de point de vue et de l'échelle, est ajoutée à la description de chaque point caractéristique. La comparaison entre les points d'intérêts et le modèle détermine la cible sur la trame courante. Cette méthode a démontré une stabilité pour le suivi des objets dynamiques. Cependant, la relation entre la stabilité et la simplicité des calculs est inversement proportionnelle par la méthode RANSAC (RANdom SAmple Consensus).

Le suivi des contours ou des silhouettes : Le suivi par le modèle d'apparence, basé sur les contours, fournit une représentation précise de la cible qui se manifeste par une bordure indiquant les frontières exactes de l'objet. Ce type de suivi a pour but d'estimer la position exacte de la cible sur chaque trame à partir des contours existant sur les trames précédentes. En effet, l'histogramme de couleur à l'intérieur du contour [KCM04] par exemple est utilisé afin d'extraire des caractéristiques et définir un objet. Dans cette stra-

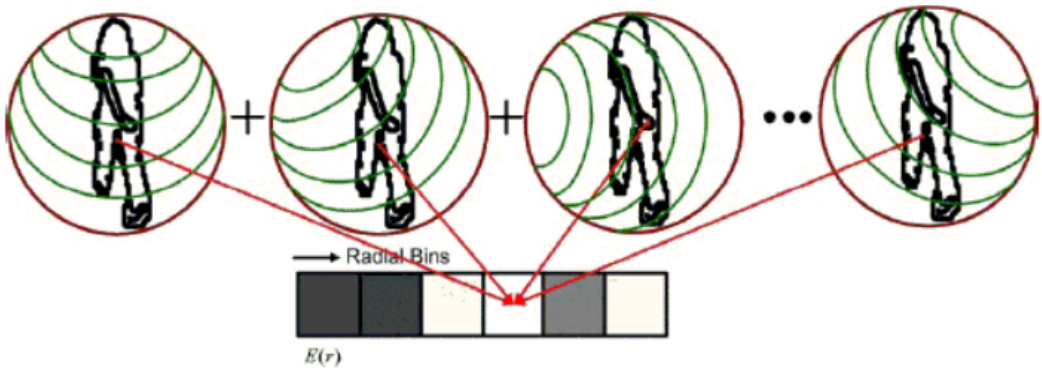


FIGURE 2.2 – Le calcul de l'histogramme des arêtes [KCM04]

tégie de suivi, la recherche d'objets se fait par évolution de contours [LCZD01], [KCM04] ou par appairage des formes [YLS04], [CRH01]. Dans la méthode d'appariement des formes, le modèle subit des mises à jour d'une trame à une autre afin de réinitialiser le contour qui subit des translations au fil du temps. Un appariement des formes est proposé dans [KCM04] et les silhouettes sont modélisées par les histogrammes de couleurs et les arêtes calculées dans des cercles englobants qui couvrent la silhouette et qui ont des rayons différents. La figure 2.2 de l'article [KCM04] illustre le calcul de l'histogramme d'arêtes. Le plus petit cercle est constitué en contenant la silhouette. Des points de référence sont choisis uniformément sur le cercle afin de définir un ensemble de cercles de rayons différents. Chaque point de contrôle correspond à une composante de l'histogramme de la silhouette. De même, un histogramme global de couleur est constitué par la même configuration. La similarité entre deux trames consécutives se mesure par le calcul de distance de Bhattacharya. En fait, ce modèle est invariant au changement d'échelle dû à la normalisation des histogrammes et de même à la rotation grâce aux points de contrôle car si la silhouette subit une rotation ce sera comme si les points de référence étaient permutés. Les algorithmes d'évolution de contours gèrent le problème du suivi en faisant évoluer le contour de la trame précédente à la trame courante. L'évolution s'effectue selon des modèles liés à l'espace d'état qui modélise l'évolution du mouvement et de la forme du contour par le filtre de Kalman [Pet99] ou par minimisation d'énergie par la descente de gradient [Man02]...

Il faut rappeler que le choix du suivi par silhouette dépend de la volonté d'avoir la région exacte et complète de l'objet. Ces algorithmes sont utilisés souvent pour les objets de formes complexes ou déformables mais, dans les environnements non contraints, ils représentent des limites non négligeables. Ce type de suivi fait appel à des mécanismes externes afin de détecter le changement comme la soustraction de l'arrière-plan mais ce n'est pas toujours évident avec les arrière-plans dynamiques. En effet, la stratégie d'évolution de contour nécessite qu'une partie de l'objet dans la trame courante soit couverte par une partie de l'objet de la trame précédente. Cette hypothèse n'est pas toujours valide à cause d'un faible taux d'images issues de la caméra ou d'un déplacement important de la cible. Le suivi par compteur ou silhouette est basé sur les modèles d'apparence holistique. Ces modèles présentent les caractéristiques de la cible de façon simplifiée car ils les calculent de façon globale, ce qui ne prend pas en considération les changements partiels et locaux qu'un objet peut subir. En outre, la disparition d'une partie de la cible, par croisement ou occultation, peut dégrader ou paralyser le suivi.

Le suivi par détection : il consiste à employer un détecteur de la classe des objets sui-

vis pour estimer les positions des cibles à chaque nouvelle image. Cette méthode est composée de deux parties : la détection et le suivi. L'étape de détection sert à localiser la cible dans l'image et à en extraire les caractéristiques. L'étape de suivi combine ces détections afin d'avoir les trajectoires au fil du temps. Ces deux étapes peuvent avoir lieu conjointement ou séparément. Pour la première catégorie où la détection se fait séparément, elle s'effectue sur chaque image et l'algorithme de suivi sert à relier les détections successives pour construire les trajectoires. Dans la deuxième catégorie, le détecteur s'exécute lorsqu'un objet apparaît dans la scène pour la première fois. Par conséquent, le détecteur permet de relancer l'algorithme de suivi. Dans les deux cas la détection est considérée comme un pré-traitement.

Les méthodes de suivi basées sur la détection sont généralement choisies selon l'évolution des détecteurs et des technologies. Certaines méthodes, comme par exemple [ZPS12], ont choisi d'exploiter le détecteur de [DT05] qui utilise la méthode de classification par SVM (Support Vector Machines). Ce type de classifieur fonctionne avec un modèle linéaire vis-à-vis de certains critères sans trop pénaliser l'erreur et avec une marge souple. D'autres travaux de suivi, comme par exemple [DABP14], montrent que les auteurs ont utilisé l'histogramme de gradient et non pas SVM pour la classification. Cette méthode de ACF (Aggregate Channel Features) repose sur une technique de classification basée sur un arbre de décision. Cette technique est appliquée avec une méthode de Boosting. Son avantage, par rapport aux précédentes, est qu'elle traite la différence d'échelle des objets détectés grâce au calcul rapide des caractéristiques visuelles; ceci se fait par l'approximation de certaines d'entre elles à partir des échelles proches. Plus récemment, plusieurs chercheurs se sont référés, dans leur suivi, aux détecteurs avec des méthodes d'apprentissage profond [DCGA17]. En fait, ce type de suivi, même s'il propose de hautes performances, reste néanmoins dépendant de la performance du détecteur et le défaut de ce dernier affecte le suivi. Plus précisément, les fausses détections et les cibles non détectées sont les défauts du détecteur qui influencent le suivi.

Discussion

D'une manière générale, l'état de l'art du suivi multi-objets connaît une hausse d'utilisation remarquable par ces différentes représentations au cours des dernières années. Les algorithmes sont de plus en plus capables de coder la structure de l'objet et de tenir compte des changements locaux qu'un objet peut subir.

Le tableau 2.1 présente les points forts et faibles des trois méthodes de suivi.

Dans cette thèse, nous soutenons que le suivi par détection offre une modélisation de structure robuste et stable plus que les autres méthodes. Cette robustesse découle du fait que cette méthode est basée sur les détecteurs qui ont vu une amélioration remarquable ces dernières années. Nous présentons, dans ce qui suit,les différents détecteurs utilisés dans une sélection de l'état de l'art.

2.2.1 Détection

Généralement, la phase de détection est constituée de trois étapes (cf. Figure 2.3) : la proposition des régions d'intérêt, puis l'extraction des caractéristiques de ces régions et enfin la décision s'il s'agit d'un objet ou non ainsi que la détermination de sa classe.

Après le détecteur de Viola et Jones [VJS05], plusieurs algorithmes ont vu le jour. Dalal et Triggs proposent l'histogramme des gradients orientés connu par HOG [DT05] et qui se révèle efficace pour la tâche de détection. Ainsi il est devenu le détecteur le plus classique. Dans [KHK07] les auteurs ont appliqué le PCA (Analyse en composantes principales) associée au HOG afin de sélectionner les caractéristiques les plus pertinentes. L'ajout du

TABLEAU 2.1 – Comparaison des méthodes de suivi

Méthodes	Références	Caractéristiques	Limites
Suivi par les points caractéristiques	[GCT+14], [YLY13]	+Comparaison par des modèles qui se mettent à jour au cours du suivi +Stabilité remarquable des objets dynamiques	-Calculs complexes.
Suivi des contours ou des silhouettes	[YLS04], [CRH01], [KCM04]	+Présentation simple des caractéristiques +Invariant à la rotation +Invariant au changement d'échelle par la normalisation.	-Nécessite un chevauchement des contours entre deux instants successifs -Incapable de représenter efficacement les changements locaux et partiels -Occultation partielle influence énormément les performances
Suivi par détection	[DCGA17] [ZPS12]	+L'entrée et la sortie des objets sont contrôlables +Les modèles d'apparences sont propres à chaque objet et reposent sur la détection pour leur mise à jour, ce qui engendre une ré-initialisation automatique	-Les défauts du détecteur affectent les performances

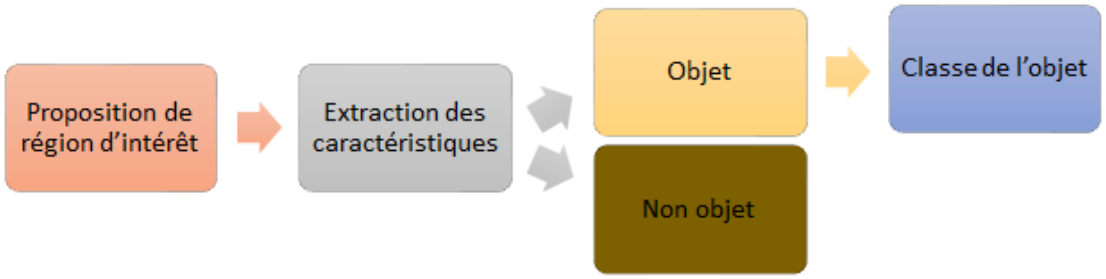


FIGURE 2.3 – La composition d'un détecteur

PCA permet de réduire le vecteur caractéristiques sans diminuer les performances. Ensuite, dans [FGMR10], est proposée la méthode DPM (Deformable Part Model) avec le HOG. Cette méthode est robuste à l'occultation et la variation des poses mais ces performances restent limitées. Ce qui a motivé les auteurs de [DABP14] à ajouter l'histogramme de couleur "LUV" et à optimiser les orientations du gradient. Depuis, d'autres travaux essayent de développer d'autres méthodes qui exploitent les histogrammes de couleurs et les filtres. Par exemple [NDH14], où les auteurs utilisent des filtres afin de limiter les corrélations des canaux de couleurs. Dans [ZBC14], les auteurs proposent un modèle de l'objet fait à base de différents filtres appelés rectangles de HAAR ; alors que d'autres travaux utilisent d'autres caractéristiques extraites par le flot optique, LBP ou les matrices de covariances. Le nombre de filtres augmente avec les années et diffère d'un travail à un autre jusqu'à l'arrivée des réseaux neurones qui ont remplacé le choix manuel des filtres.

Les réseaux profonds ont révolutionné la vision par ordinateur. Les méthodes basées sur les réseaux convolutifs (CNN) ont donné un niveau élevé de performances sur plusieurs classes et plusieurs applications en même temps telles que la classification [BJWW15], la détection [RHGS17], la segmentation [LSD15]. Dans [BJWW15], les auteurs proposent le MCCNN (Multichannel Convolutional Neural Network) afin d'effectuer la reconnaissance multimodale de l'état émotionnel des personnes. Ils utilisent les caractéristiques profondes pour déchiffrer l'état émotionnel avec des performances élevées. D'autres méthodes appliquent l'apprentissage profond pour extraire des caractéristiques telles que [CPLP16] et [SKCL13] où les auteurs utilisent un CNN spécialement un modèle non supervisé basé sur CSC (Convolutional Sparse Coding) pour la détection des personnes. Ensuite, pour limiter l'effet du changement d'apparence, dans [LTWT14], ils ajoutent d'autres couches au réseau de neurones convolutif. Ces couches sont construites avec la Restricted Boltzmann Machine (RBM). Dans [AKV+15], les auteurs profitent de cette évolution et utilisent les cascades afin d'améliorer la vitesse de la détection et ils ont atteint 15 trames par secondes. Récemment, dans [AAR+17], [KSH12] deux réseaux, CifarNet et AlexNet, sont présentés pour la détection et ont prouvé leurs performances. Pour l'apprentissage de AlexNet, ils utilisent R-CNN (Region with CNN features) et ils prouvent que le prétraitement avec un volume de données plus important augmente les performances. Les méthodes d'apprentissage profond utilisent souvent les réseaux de neurones convolutifs afin d'effectuer la classification. Cependant, ces réseaux sont extrêmement lents à fonctionner en mode de fenêtre glissante ; par conséquent, la plupart des méthodes optent pour l'utilisation des modules de propositions de régions d'intérêt.

Le tableau 2.2 présente quelques algorithmes de détection avec leurs caractéristiques et le pourcentage des faux positifs par la métrique "Miss-Rate". En fait, d'après ce que nous avons déjà exposé, le suivi par détection est limité par les défauts du détecteur, spécialement par les pourcentages des faux positifs ; ce qui est désigné dans le tableau par les Miss-Rate .

2.2.2 Suivi

Le suivi sert à créer des trajectoires pour chaque objet. Ce dernier possède un identifiant (ID) unique qui lui est assigné grâce à un algorithme de suivi. En effet, la similarité entre deux objets situés sur deux trames successives constitue la trajectoire. L'algorithme de suivi cherche cette similarité afin de combiner les résultats de détection dans le cas des algorithmes de suivi par détection où le détecteur s'effectue sur toutes les trames de la séquence. Dans d'autres algorithmes, le suivi nécessite l'initialisation de la détection lorsque l'objet entre pour la première fois sur scène. Ensuite, le suivi et la détection s'effec-

TABLEAU 2.2 – Classification des méthodes de détection selon leur taux de faux positifs

Algorithmes	Caractéristiques	Classificateurs	Miss-Rate	Bases
[VJS05]	Haar	AdaBoost	94.73%	Inria
[SM07]	Shapelet	AdaBoost	91.37%	Inria
[SKCL13]	Pixels bruts	Deep Learning	77.20%	Inria
[DT05]	HOG	SVM	68.46%	Inria
[WS08]	HOG +Haar +Shape Context+ Shapelet(MultiFtr)	AdaBoost	68.26%	Inria
[WHY09]	HOG+LBP	SVM	67.77%	Inria
[FGMR10]	HOG	LatentSVM	63.26%	Inria
[WMSS10]	MultiFtr+ Couleur	SVM	60.89%	TUD-Motion
[DBP10]	Channel	AdaBoost	57.40%	Inria
[WMSS10]	MultiFtr+ Couleur+ Mouvement	SVM	50.88%	TUD-Motion
[BMTVG13]	Channel	AdaBoost	50.17%	Inria
[DABP14]	Channel	AdaBoost	44.22%	Caltech-USA
[GLVP18]	HOG	SVM	43.42%	Caltech-USA
[OW13]	Gradient+Couleur	Deep Learning	39.32%	Caltech-USA
[YZL+13]	HOG	SVM	37.64%	Caltech-USA
[PZRD13]	HOG +mouvement (gradient temporel)	SVM	37.34%	Caltech-USA
[BMTVG13]	Channel	AdaBoost	34.81%	Caltech-USA
[ZBC14]	Haar optimisé	AdaBoost	34.60%	Caltech-USA
[OW13]	CifarNet	SVM	28.4%	Caltech-USA
[ZLX+14]	AlexNet	SVM	23.3%	Caltech-USA
[TLWT15]	TA-CNN	Calcul de probabilité	25.64%	Caltech-USA
[HOBS15]	SpatialPooling+	Softmax	21.9%	Caltech-USA

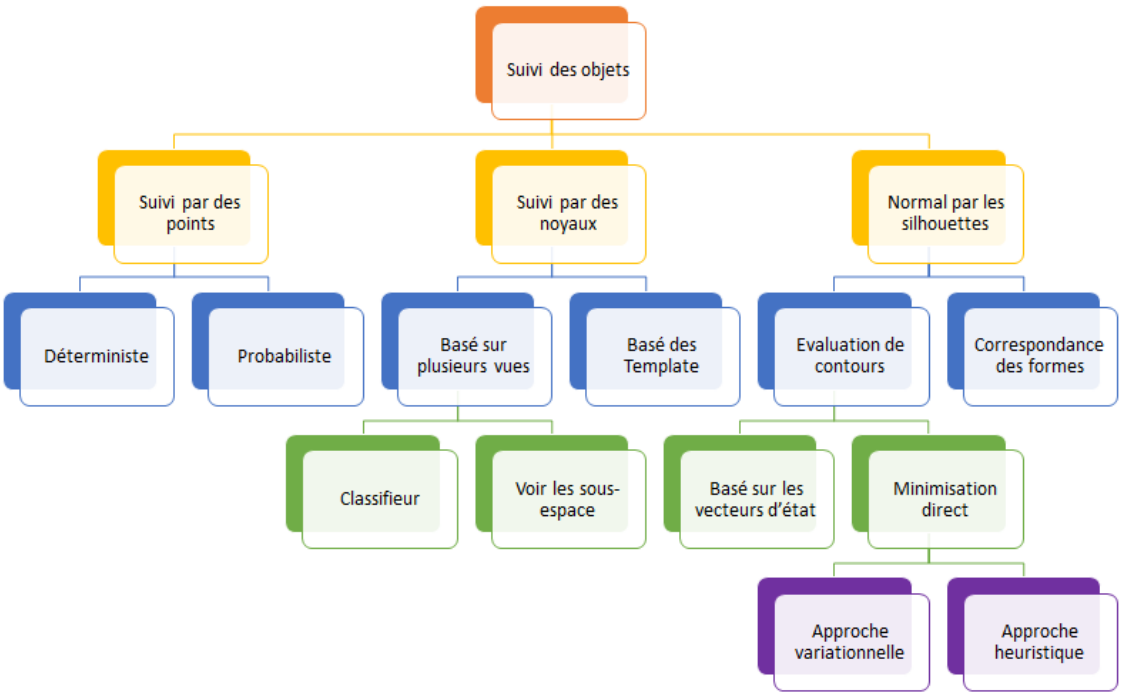


FIGURE 2.4 – Une classification des algorithmes de suivi

tuent en même temps. Les algorithmes de suivi des objets sont répartis en trois groupes comme le montre la figure 2.4 selon [YJS06].

Dans [AMGC02], les auteurs introduisent le filtre de Bayes qui se réfère aux lois probabilistes afin de faire du suivi un mono-objet où la cible est présentée par un point. En fait, les filtres de Bayes sont approximés pour les systèmes linéaires Gaussiens avec un filtre de Kalman et en général avec un filtre à particules. Cependant, afin de suivre plusieurs points qui présentent plusieurs objets, il faut une méthode d'association. Ces méthodes servent à relier les observations issues d'un détecteur. Elles peuvent être déterministes (exemple, la théorie des graphes [TMC06]) ou basées sur des statistiques (exemple JPDA (Joint Probabilistic Data Association) [JMCB06] ou MHT (Multiple Hypothesis Tracker)[Bla04]).

Les mesures générées par le détecteur pendant le suivi représentent les nœuds du graphe et une arête introduit le coût lié au déplacement de la cible entre deux nœuds. Les trajectoires sont définies par le fait d'avoir un chemin entre deux nœuds qui minimise des fonctions de coût. Les méthodes existantes sont variées et se distinguent par les dimensions de la fenêtre temporelle ainsi que par le choix de la méthode d'optimisation et de la fonction de coût.

Dans [KUS03], les auteurs introduisent une application spéciale de la théorie des graphes GNN (Global Nearest Neighbor) qui assure une solution optimale entre deux instants successifs pour une fenêtre temporelle. Autrement dit, à l'aide d'une matrice de coût permettant de relier les mesures aux pistes, le GNN aide à trouver une solution qui sert à minimiser le coût global des associations. En fait, un filtre de Kalman est utilisé dans ce cas afin de prédire et mettre à jour les pistes. C'est la fonction de Mahalanobis qui est utilisée pour calculer le coût. Grâce à l'aspect déterministe de la méthode, le nombre de paramètres reste limité. Ce qui facilite l'utilisation de la méthode dans divers scénarios de suivi. La complexité de la méthode augmente exponentiellement avec le nombre de mesures. Pour obtenir une complexité polynomiale, certaines contraintes relatives aux caractéristiques des cibles sont utilisées par des solutions sous optimales. Dans le cas de GNN, le filtrage de Kalman est utilisé afin d'intégrer un aspect probabiliste. Afin de modé-

rer les interactions entre cibles et les faux négatifs ou positifs, on opte pour les méthodes statistiques utilisant l'aspect probabiliste. Par exemple, la méthode JPDA [HRMZ+15] permet de mettre à jour l'état de la cible en prenant en considération toutes les mesures à l'instant courant. En fait, lors de la mise à jour, chaque mesure est pondérée par un poids qui correspond à la probabilité liée à l'association cible-piste. Par conséquent, cette stratégie assure une robustesse contre les faux positifs. Cependant, cette méthode impose que le nombre d'objets soit fixe et connu. En contrepartie, la méthode MHT propose des hypothèses d'association et les utilise au fil du suivi. Le score des hypothèses propagées permet de prendre une décision optimale. Néanmoins, le nombre d'hypothèses augmente de façon exponentielle, ce qui engendre une complexité importante de la méthode. Ceci est considéré comme un inconvénient majeur mais permet pourtant, comme la théorie des graphes, de modérer l'effet des occultations grâce à la fenêtre temporelle.

La méthode basée sur le filtre de Bayes évolue pour trouver une solution au suivi multi-objets sans avoir recours à des méthodes d'association. Ceci engendre une estimation des positions et du nombre de cibles en parallèle avec la modélisation des mesures et des pistes par la méthode RFS (Random finit Set). En effet, cette méthode est l'ensemble de vecteurs ou de variables aléatoires. Elle modélise le nombre d'éléments par l'intermédiaire d'une densité de probabilité discrète ainsi que l'ensemble des éléments par la densité de probabilité conjointe. Cette extension a généré les filtres PHD (Probability Hypothesis Density) dans [MTC08] et CPHD dans [LCB13] qui dépendent du calcul de l'approximation de la densité multi-objets. Le filtre PHD est basé sur le principe de la loi de Poison, ce qui signifie que seul le moment statistique du premier ordre est propagé durant le suivi. Par contre, pour la méthode de CPHD, toute la densité est propagée et le nombre d'objets ne se limite pas à la loi de Poison et se présente par une densité discrète. L'inconvénient de ces méthodes probabilistes est lié à la distribution statistique des données dont le nombre de paramètres est élevé en comparaison avec les méthodes déterministes.

Afin de gérer les objets de formes géométriques paramétrables, les auteurs de [PT05] proposent une méthode de recherche de correspondance entre les blocs par (Template Matching). Pour chaque objet, l'algorithme cherche l'apparence de la région la plus proche du vecteur représentant l'objet par calcul de distance. Cette méthode s'avère efficace pour le suivi mono-objet ; par contre, elle nécessite l'utilisation d'une méthode d'association pour le multi-objets.

La méthode CAMSHIFT [ZQFX09] nécessite la détection pour l'initialisation et chaque cible est représentée par l'histogramme de couleurs du rectangle qui englobe la cible. La méthode vise à déterminer le maximum local de la densité de probabilité, de manière itérative, afin de faire converger la largeur, le centre de masse et la hauteur du rectangle vers la moyenne des pixels dans le rectangle. L'avantage qu'a cette méthode par rapport aux autres est qu'elle permet de localiser facilement la cible. Par contre, son inconvénient réside dans la perte de la cible au cas où il n'y a pas de chevauchement entre deux trames successives.

Les filtres à particules [KL09], contrairement au filtre de Kalman, permettent un large choix de caractéristiques dans le cas où le bruit n'est pas Gaussien et que l'évolution n'est pas linéaire. Le principe de ces filtres se manifeste par la représentation de la densité de probabilité d'une cible par l'ensemble des particules. Ces derniers subissent des mises à jour de correction et de propagation au fil du temps. En contrepartie, les particules se caractérisent par une dimension élevée car la densité de probabilité est liée à l'information d'apparence ainsi qu'à l'information spatiale. Par conséquent, la complexité augmente de façon exponentielle avec le nombre et la dimension de la particule.

L'évolution des méthodes d'apprentissage profond et leurs performances a permis de les utiliser pour le suivi multi-objets ; spécialement pour les réseaux de neurones récurrents (RNN) [GK96] qui incluent une boucle entre l'entrée et la sortie. Cela permet, non seulement de stimuler l'effet mémoire, mais aussi de faire la correspondance entre les séquences d'entrées et celles de sorties, à condition que l'alignement des séquences et les dimensions d'entrée et de sortie soient connus à l'avance.

Les chercheurs apprennent l'association des détections par des réseaux profonds. Dans [GGZC15], les auteurs peuvent par des RNNs prédire la position de l'objet dans chaque frame tout en la comparant aux mesures de la détection ; ce qui réduit le nombre de faux positifs. De même, dans [NZH+17], le détecteur YOLO a été utilisé afin de détecter les positions et les RNNs pour l'association. La détection dans chaque frame rend le suivi multi-objets plus flexible.

Le LSTM (Long short-term memory), cas particulier de réseau récurrent, est aussi utilisé dans le cas de suivi. Dans [VSL+16], le LSTM est utilisé afin d'intensifier la capacité de discrimination profonde par la mémorisation de l'information spatiale. D'ailleurs, dans les méthodes classiques, il existe de nombreux cas où les modèles de suivi, et spécialement lors de suivi par mouvement linéaire, ne peuvent pas traiter les occultations à long terme. Pour remédier à un tel problème, les auteurs de [SAS17] proposent un modèle de mémoire à base de LSTM qui permet de prédire des modèles de mouvement. C'est une approche entièrement basée sur les données qui peuvent gérer les détections bruitées.

Les méthodes d'association dans l'étape de suivi classique ou celles à base de réseaux profonds présentent des avantages et des inconvénients. Le tableau 2.3 présente une comparaison entre les différentes méthodes d'association citées dans cette section.

2.2.3 Discussion et positionnement au niveau suivi

Le suivi dans une seule caméra est essentiellement basé sur trois étapes : la détection, l'extraction des caractéristiques et les méthodes d'association liées au suivi.

— **La détection :** Les méthodes de détection sont nombreuses et différentes. En fait, la détection peut être effectuée par des méthodes basées sur des filtres ou des histogrammes et peut aussi avoir lieu avec des méthodes dépendant de l'apprentissage. Le choix de notre méthode de détection dépend de deux critères : un taux de performance élevé et un faible taux de faux positifs.

D'après le tableau 2.2, les méthodes de détection ont considérablement évolué au fil des années alors que le taux de faux positifs a nettement diminué. Il est passé de 94.73% dans les travaux de [VJS05] en 2005 à 21.9% après 10 ans en 2015 dans [HOBS15]. D'ailleurs, les méthodes basées sur l'apprentissage présentent le taux le plus faible en terme de faux positifs [HOBS15], [TLWT15], [ZLX+14]...

Les méthodes basées sur l'apprentissage définissent l'objet comme une classe et permettent de le détecter avec des caméras fixes et mobiles. Les performances dépendent de plusieurs paramètres et sont influencées par la base de données de l'apprentissage.

La plupart des méthodes basées sur l'apprentissage utilisent des bases de données de grande taille ; par exemple le détecteur Faster RCNN [Gir15] est pré-traîné avec la base Imagnet qui contient des centaines de milliers d'images avec une variété de classes. Ces réseaux présentent aussi une fonctionnalité d'entraînement qui définit le réseau avec un nombre de classes bien précis et d'autres bases ou images qui permettent de spécialiser le réseau et d'augmenter les performances.

TABLEAU 2.3 – Comparaison des méthodes d'association

Méthodes	Références	Caractéristiques	Limites
GNN	[KUS03]	+Solution optimale entre deux instants successifs +Le nombre de paramètres reste limité	-Complexité polynomiale
JPDA	[HRMZ+15], [JMCB06]	+Robuste contre les faux positifs et négatifs	-Impose le nombre d'objets : Il doit être fixe et connu
MHT	[Bla04]	+Modère les occultations +Les hypothèses se mettent à jour au fil du temps	-Complexité exponentielle
RFS	[LCB13], [MTC08]	+Le moment statistique d'ordre 1 est propagé au fil du temps	-Nombre de paramètres très élevé
Templete matching	[PT05]	+Gère les formes géométriques paramétrables	-Des lacunes avec le ssssss
CAMSHIFT	[ZQFX09]	+Simplicité dans la localisation de l'objet	-Perte de l'objet en absence de chevauchement entre deux images successives
Filtre à particules	[KL09]	+Large choix de caractéristiques	-Dimension élevée et complexité exponentielle
RNN	[SAS17], [VSL+16]	+Stimule l'effet mémoire avec les couches cachées +Mémorise le modèle de mouvement +Prédire les positions et limite les faux positifs	-Nécessite un apprentissage -Nécessite des couches de décision

Les méthodes basées sur des réseaux profonds sont nombreuses. Une des architectures des réseaux profonds est le réseau de neurones convolutif (CNN), qui est le plus utilisé jusqu'à nos jours dans les tâches de vision par ordinateur. Les CNNs [LBH15] introduisent les couches de convolution dans les réseaux qui sont capables, à partir d'une image d'entrée, d'apprendre ces caractéristiques et ce à l'aide de plusieurs filtres tels que ceux de couleurs ou ceux de la détection de la présence ou de l'absence d'une ligne à partir des cartes de caractéristiques. En appliquant des filtres, qui sont des matrices de valeurs scalaires, on ajuste la valeur de chacun des poids. Chaque couche de convolution utilise ces cartes de caractéristiques de manière similaire à une transformation non linéaire et elles sont apprises à l'aide d'une descente de gradient avec une rétro-propagation. Le CNN introduit aussi les couches de pooling. Ces couches permettent de réduire le calcul et d'augmenter la robustesse en divisant uniformément la carte des caractéristiques en régions ; et il est à préciser qu'elles ne retournent que les valeurs d'activation les plus élevées.

Grâce au nombre important de cartes des caractéristiques de chaque couche de convolution pour chaque entrée, les CNNs sont particulièrement bien adaptés au traitement des données provenant de plusieurs canaux, tels que les images en couleurs qui possèdent trois canaux de couleurs [LBH15]. Quatre architectures de CNN ont été définies et leurs performances sont calculées par rapport au coût et à l'utilisation de la mémoire par rapport à la précision. Ces architectures incluent AlexNet, VGG, GoogleNet et ResNet et contiennent entre 7 à 152 couches. Une approche commune permettra d'apprendre la tâche de classification et calculer les poids avec l'une des quatre architectures traînées par une base de données publique afin d'initialiser les paramètres et les re-traîner par une base spécifique. En fait, le CNN a pu démontrer un succès remarquable dans la classification. Deux approches pour la détection d'objet sont particulièrement mises en valeur dans la littérature. La première est basée sur le R-CNN où l'entrée est divisée en plusieurs boîtes de tailles différentes et ce en utilisant un algorithme de ségrégation pour chaque région passant par le CNN. Puis, le Fast R-CNN introduit le module qui propose la région d'intérêt et qui est basé sur la dernière carte des caractéristiques afin de diminuer la dimension de la région. Finalement, le réseau Faster R-CNN propose un réseau RPN (Region Proposal Network) pour la recherche des zones d'intérêt. La deuxième approche de détection est YOLO. Elle divise l'image en grille et chaque cellule de la grille sert à classifier et définir la taille de la zone d'intérêt. Ce détecteur est moins précis que Faster R-CNN à cause des grilles qu'il utilise mais il est plus rapide [RDGF16].

Pour conclure, la méthode Faster R-CNN est adéquate pour notre contexte et permet un fonctionnement dans différents scénarios et environnements avec le multi-objets grâce à sa précision.

— **Extraction des caractéristiques :**

L'extraction des caractéristiques se fait par un seul filtre ou par la combinaison de deux ou plusieurs dans le but de décrire l'objet. Ces caractéristiques sont liées à l'apparence, au mouvement ou aux deux à la fois. Cette représentation à partir des caractéristiques sera l'entrée du classifier.

Les caractéristiques liées au mouvement nécessitent la position et la taille de l'objet. Grâce à un modèle de mouvement, l'accélération et le mouvement peuvent être déduits. Le vecteur d'état, qui contient l'information temporelle, est généralement de faible dimension et ne prend pas trop d'espace dans la mémoire. En effet, le filtre de Kalman a prouvé des performances non négligeables et a été utilisé dans plusieurs travaux. Puis, avec l'évolution des méthodes d'apprentissage profond, les

RNNs ont été utilisés afin de déduire l'information temporelle.

En contrepartie, l'extraction de l'information liée à l'apparence de l'objet nécessite plus de calcul et de complexité. Cette information peut être extraite par deux stratégies différentes. La première nécessite la position de l'objet et peut dépendre des histogrammes de couleurs, de gradient orienté ou de template de couleurs... Cette information est ainsi extraite par l'application d'un ou plusieurs filtres ou par celle d'un descripteur.

La deuxième requiert un apprentissage pour extraire les informations de l'objet. Cette méthode dépend des réseaux profonds, spécialement les CNNs. L'extraction des caractéristiques se fait à partir des couches totalement connectées.

Elle peut surtout avoir lieu sur la totalité de l'objet ou sur des parties précises lors du traitement des piétons.

Nous estimons que la combinaison de deux informations sur le temps et l'apparence permet d'avoir une représentation robuste de l'objet.

— **Le suivi :**

Dans la littérature, plusieurs méthodes de suivi sont détaillées et présentent des résultats non négligeables. Les méthodes de suivi sont divisées en deux, celles basées sur l'apprentissage profond et les autres. En fait, les méthodes classiques sont : GNN [KUS03], MHT [Bla04] ou JPDA [HRMZ+15]. Elles présentent des solutions optimales mais la complexité reste polynomiale et exponentielle. Les méthodes, à base de réseaux profonds, comme par exemple le RNN dans [SAS17], [VSL+16], stimulent l'effet mémoire des couches cachées de ces modèles afin de définir une liaison entre les détections et nécessitent un apprentissage.

Cependant, dans la majorité des cas de suivi, peu importe les méthodes, des problèmes persistent encore telles que la signature unique et spécifique de chaque objet, la vérification de la qualité du suivi et la correction des trajectoires. Bref, dans l'état de l'art, tous les travaux ont été focalisés sur le fait d'associer les détections à court terme sans prendre en considération les défauts qui proviennent déjà du détecteur. En outre, après un croisement ou une occultation, la majorité des méthodes de suivi ne vérifient pas les IDS des fragments qui appartiennent au même objet et qui risquent donc d'avoir des IDS différents ; ce qui nous encourage à suivre une stratégie basée sur ces fragmentations.

2.2.4 Évaluation des méthodes de suivi

Dans cette partie, les protocoles d'évaluation des méthodes de suivi sont présentés. Ils incluent les bases de données publiques et les métriques les plus utilisées.

Les bases de données

La littérature présente une diversité de bases de données dédiées spécialement au suivi multi-objets due au nombre important d'applications et de travaux associés. De nouvelles bases ont vu le jour avant et au cours de cette thèse. Le choix entre ces bases de données se fait selon les bases les plus utilisées par les travaux les plus proches de travaux présentés dans cette thèse. Les bases les plus populaires dans le suivi multi-objets sont : PETS [FS09], ETH [ELVG07], ParkingLot [SDO+12], TUD [ARS09], TownCenter [BR11]...

Pour quelques séquences d'images, les détections sont disponibles. Dans nos travaux, nous avons utilisé trois séquences PETS S2L1, PETS S2L2 et ETH, considérées comme les bases les plus populaires pour le suivi multi-objets.

(a) Frame 198 de la base PETS S2L1 (b) Frame 2 de la base PETS S2L2

FIGURE 2.5 – Exemple d'image des bases PETS S2L1 et PETS S2L2

Les deux séquences PETS S2L1 et PETS S2L2 sont des vidéos issues des scènes fixes les plus proches des applications de vidéo-surveillance. Ces deux séquences présentent un flux différent du niveau de densité et de fréquence des personnes. Des exemples d'images de deux scènes sont présentées dans la figure 2.5.

La base ETH est très utilisée puisqu'elle présente plusieurs challenges. Étant une base prise par une caméra mobile présentant plusieurs états de croisement et d'occultation, plus de détails sur ces bases de données sont dans le chapitre 3 section 3.5.1

Les vérités de terrain de suivi des bases de tests sont en libre accès, ce qui facilite la comparaison avec les autres travaux. Les vérités de terrain présentent les trajectoires des objets en identifiant leurs positions et leurs IDs dans chaque trame.

Métriques

L'évaluation des méthodes de suivi nécessite des bases de données et des métriques d'évaluation pour permettre la comparaison avec d'autres travaux. Le choix des métriques ou le fait de trouver une mesure unique pour calculer les performances du suivi multi-objets s'avère un peu difficile à cause de plusieurs facteurs qui peuvent affecter les résultats. En fait, les erreurs liées à la détection sont parmi les facteurs qui influent sur le suivi. D'autre part, les autres facteurs sont liés à l'association des données et à l'application qui utilise les résultats. En effet, l'application peut influencer les performances par le maintien des identifiants des objets sans changement de l'ID, par la limite de la fragmentation et par le fait qu'un seul objet ne peut avoir qu'une seule trajectoire ainsi que par la localisation précise des cibles. De ce fait, la majorité des travaux de l'état de l'art ont essayé de trouver des métriques qui prennent en considération toutes les erreurs. Dans [MSR13], les auteurs présentent une synthèse des métriques liées au suivi. La plupart des travaux utilisent les métriques CLEARMOT pour évaluer leurs travaux. L'algorithme d'évaluation de CLEARMOT repose sur l'appariement entre les hypothèses issues de l'approche et les mesures de l'état de l'art.

Dans le but d'apparier les hypothèses des trajectoires des méthodes de suivi avec celles de la vérité-terrain, il faut associer à chaque trame de la vidéo les positions hypothétiques des objets associés. Cette condition n'est pas évidente dans le cas où les objets suivis sont proches les uns des autres. Soit $H = H_1, H_2, ..., H_n$ l'ensemble des hypothèses liées au suivi, $O = O_1, O_2, ..., O_n$ l'ensemble des objets de la vérité terrain et F_t la frame à l'instant t. Les étapes d'affectation hypothèses-cible avec CLEARMOT sont :

— A l'instant $t-1$, soit H l'hypothèse associée à l'objet O. F_{t-1} la frame à l'instant

$t-1$, si H et O sont présents et suffisamment proches tels que $d_{\text{CLEARMOT}}(\text{H},\text{O}) \leqslant th_{\text{CLEARMOT}}$. Par conséquent, ils seront aussi considérés comme associés à l'instant t dans la frame F_t tant que la distance $d_{\text{CLEARMOT}}(\text{H},\text{O})$ n'a pas dépassé le seuil limite d'appariement th_{CLEARMOT}.

— S'il existe à l'instant t des H et O qui n'étaient pas associés à l'instant $t-1$, un appariement est donc nécessaire pour minimiser la somme de $d_{\text{CLEARMOT}}(\text{H},\text{O})$ entre les hypothèses et les cibles tout en considérant que seul le couple H et O ayant une distance inférieure au seuil peut être associé ; et le calcul d'appariement optimal se fait à l'aide de la méthode hongroise. En fait, cette méthode d'optimisation combinatoire permet de résoudre des problèmes d'affectation avec la contrainte de temps polynomial. Elle est capable de trouver dans un graphe biparti, un couplage parfait du poids maximum.

Ce protocole entre les hypothèses et les cibles permet l'association entre les résultats d'une approche et la vérité-terrain. Dans le suivi 2D, le calcul de la distance d_{CLEARMOT} se fait par le calcul du rapport entre l'aire de la partie de chevauchement entre les deux rectangles englobant l'objet et l'aire de l'intersection. Le seuil limite d'appariement est fixé à $\frac{1}{2}$ d'après des travaux d'état d'art exemple [MSR13]. Grâce à cette définition, un calcul des métriques est possible. Par la suite, nous présentons les métriques les plus utilisées en CLEARMOT.

Les CLEARMOT font appel à d'autres métriques usuelles qui sont déterminées par l'appariement des hypothèses avec la vérité-terrain tel que le nombre de faux positifs (FP), de faux négatifs (FN) et de changements d'IDs (Identity Switches :IDS). Toute hypothèse non associée à une cible est comptée comme un faux positif et l'inverse est considéré comme un faux négatif et ce pour toutes les cibles de la vérité-terrain qui n'ont pas été associées à une hypothèse de trajectoire. Le changement d'ID est pris en compte si une cible est associée à une trajectoire H_i et dont la trajectoire dépend des hypothèses de H_j avec $\text{H}_j \neq \text{H}_i$. A partir de ces trois paramètres, plusieurs métriques sont définies.

La métrique MOTA : "Multiple Object Tracking Accuracy", elle est définie par l'équation 2.1 où P_o est l'ensemble des positions dans toutes les images que la cible a occupée :

$$\text{MOTA} = 1 - \frac{\text{FP} + \text{FN} + \text{IDS}}{\text{P}_o} \tag{2.1}$$

La métrique MOTP : "Multiple Object Tracking Precision" reflète la valeur moyenne de la distance entre les hypothèses et les positions de la cible liées à la vérité-terrain, la distance étant précédemment définie par $d_{\text{CLEARMOT}}(\text{H},\text{O})$. Ces métriques (MOTA, MOTP, FN, FP et IDS) reflètent la qualité du suivi. En fait, MOTP évalue la qualité de la localisation de l'objet et MOTA prend en considération plusieurs types d'erreurs comme FN, FP et IDS lesquels sont engendrés par la détection et l'appariement.

A partir de ces appariements, deux autres métriques sont définies "Precision" et "Recall". La figure 2.6 illustre le calcul de ces deux métriques. En fait, "Precision" calcule le rapport entre les faux négatifs et la somme des vrais et des faux positifs. La métrique "Recall" calcule le rapport entre les vrais positifs et la somme des vrais positifs et faux négatifs.

Enfin, un résumé des métriques les plus utilisées dans les travaux des états de l'art sont détaillés par la suite tout en les indiquant avec des symboles (↑ et ↓) pour définir quand les résultats sont considérés comme meilleurs.

— IDS ↓ : indique le changement d'ID et IDS $\in [0, +\infty]$.

— FN ↓ : indique le nombre de faux négatifs et FN $\in [0, +\infty]$.

— FP ↓ : indique le nombre de faux positifs et FP $\in [0, +\infty]$.

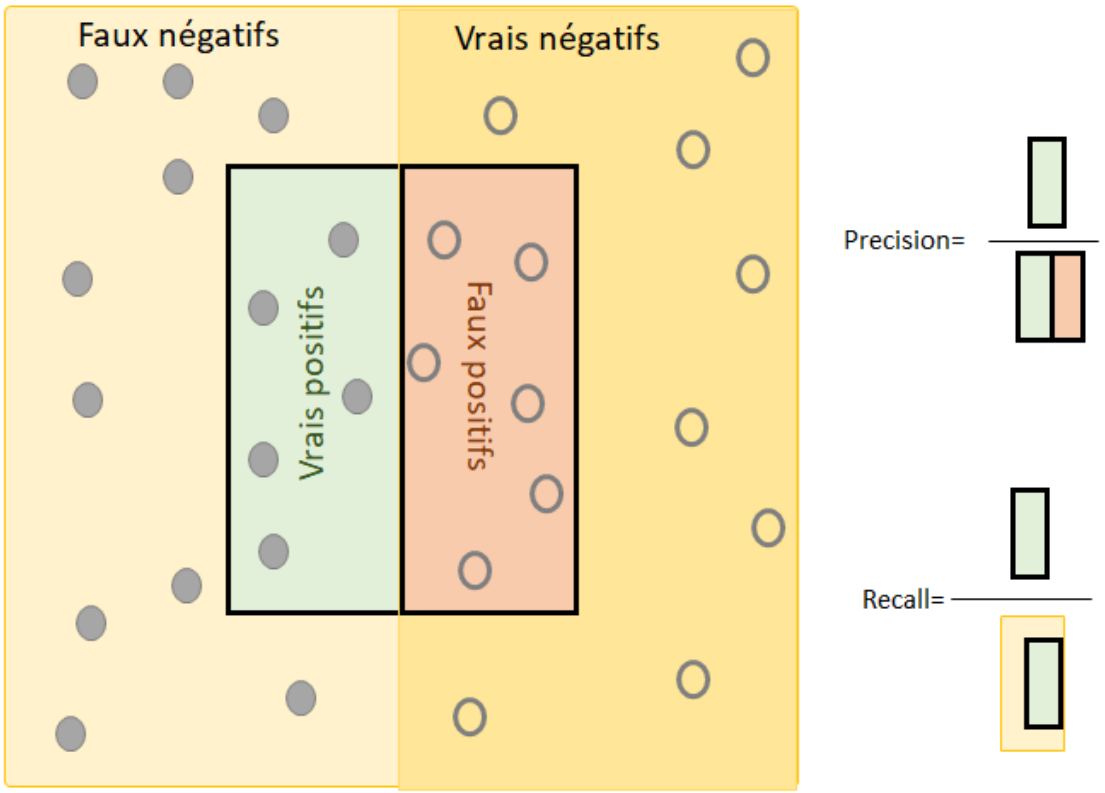


FIGURE 2.6 – Calcul des métriques Recall et Precision

— MOTA ↑ : $1 - \frac{\mathrm{FP} + \mathrm{FN} + \mathrm{IDS}}{\mathrm{P}_o}$ et MOTA $\in [-\infty, 1]$.

— MOTP ↑ : $d_{\mathrm{CLEARMOT}}(\mathrm{H}, \mathrm{O})$ et MOTP $\in [\frac{1}{2}, 1]$ avec $th_{\mathrm{CLEARMOT}} = \frac{1}{2}$.

— Precision ↑ : les faux négatifs par rapport aux vrais positifs et faux positifs et $\mathrm{P}recision \in [-\infty, 1]$.

— Recall ↑ : les vrais positifs par rapport la somme des vrais positifs et faux négatifs et $\mathrm{R}ecall \in [-\infty, 1]$.

2.3 Suivi dans un réseau de caméras

La ré-identification sert à suivre un objet dans un réseau de caméras : il s'agit de suivre l'objet dans une caméra et le ré-identifier dans d'autres. Le suivi présente plusieurs problématiques qui deviennent encore plus complexes dans la ré-identification. Il est à préciser que les défis de la ré-identification sont liés à la variation de l'apparence de l'objet qui a subi une variation de pose, d'angle de vue et plusieurs autres. Le défi se résume dans la façon de décrire l'objet de manière robuste afin qu'il soit identifié à partir de n'importe quelle vue des différentes caméras. Les caractéristiques globales sont utilisées dans la littérature pour décrire les objets dans la plupart des applications de ré-identification [GJLY21]. Les réseaux profonds dominent aussi le domaine de ré-identification et dans cette étude de l'état de l'art nous nous limitons à des approches basées sur des réseaux profonds puisque la plupart des travaux récents se focalisent sur cette technologie.

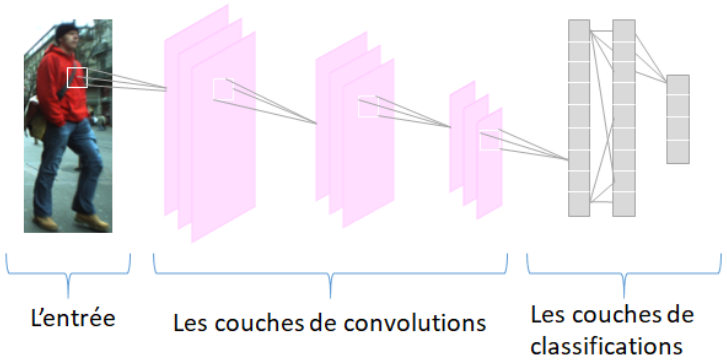


FIGURE 2.7 – Un exemple de modèle de réseau profond

2.3.1 Ré-identification

Dans cette partie, nous présentons une sélection de travaux de l'état de l'art basés sur des réseaux profonds. Les architectures les plus récentes sont développées afin de remédier à des problèmes de ré-identification des personnes que les architectures classiques n'ont pas pu résoudre ainsi qu'à l'amélioration des performances. En général, les architectures peuvent être classées en deux catégories : des modèles de classification afin de résoudre des problèmes de ré-identification et des modèles basés sur des réseaux siamois de comparaison par paires et par triplets.

Les modèles de ré-identification souffrent toujours du manque de données pour l'apprentissage. Ceci résulte du fait que les bases de données disponibles contiennent peu d'échantillons de chaque individu comme par exemple la base VIPeR [GBT07] qui contient deux paires d'échantillons pour chaque individu. Ce problème engendre des modèles non robustes et lors de la phase de test, il augmente le risque d'avoir des problèmes de sur-dimensionnement ("overfitting"). Par conséquent, une comparaison par paires a été proposée avec une classification basée sur un réseau Siamois dans [LZXW14]. Le modèle prend en entrée une paire d'images pour chaque personne. Les réseaux siamois peuvent être une solution dans le cas d'un volume restreint de données [ZYH16]. Par la suite, nous commentons quelques travaux de ré-identification basés sur des réseaux profonds poursuivis par une présentation des fonctions d'erreurs et les techniques d'augmentation du volume des données.

La ré-identification par des réseaux profonds

Les réseaux de neurones profonds à base de classification pour la ré-identification ont typiquement la même structure présentée par la figure 2.7. Dans ces modèles, le réseau prend en entrée l'image de l'individu pour chercher la probabilité d'appartenance. Dans le cas de la base VIPeR, ce type de réseau peut facilement échouer à cause du problème de surdimensionnement.

Dans [WCL+16], les auteurs proposent un réseau profond en parallèle avec une architecture classique pour extraire les caractéristiques sous forme d'un histogramme. Le réseau prend en entrée une seule image de dimension $224 \times 224 \times 3$ qui passe par les couches de convolution, d'une part, et par les filtres d'extraction de couleur et de texture de l'autre après une division horizontale. Les résultats des deux architectures sont fusionnés par une couche intégralement connectée. La couche de fusion est nécessaire pour remédier à la différence de dimension entre les caractéristiques issues de l'architecture classique et celle du réseau profond. Une couche de softmax prend la sortie de la couche

totalement connectée pour minimiser la fonction d'erreur "cross-entropy loss". La taille de cette couche dépend des caractéristiques issues de l'architecture classique. L'apprentissage dans le réseau se fait à travers l'application de minis-batchs stochastiques du gradient "mini-batch stochastic gradient" pour la rétropropagation.

Puis, dans [WSvdH17], les auteurs proposent une seule architecture au lieu de deux en parallèle et remplacent l'architecture classique par un descripteur SIFT et un module d'extraction de l'histogramme de couleurs. Le vecteur caractéristique issu de ces deux modules est réduit par un ACP. Ensuite, il entre dans des couches totalement connectées par l'encodage vectoriel ("Fisher vector encoding") dans le but de produire un vecteur caractéristique. En fait, chaque vecteur encodé par "Fisher vector encoding" est calculé à partir du descripteur SIFT et LAB color. Finalement, après les couches totalement connectées, les auteurs emploient LDA "linear discriminative analysis" afin d'élargir la marge entre les classes.

Dans [XLOW16], les auteurs proposent un apprentissage profond à partir de plusieurs bases de données utilisant un réseau de neurones convolutifs (CNN) afin de découvrir l'effet de chaque apprentissage sur les données. Ils proposent un modèle robuste à partir de l'apprentissage avec les différentes bases de données et la classification se fait par une couche de softmax loss. Après un passe-avant pour tous les échantillons de différentes bases, ils calculent pour chaque neurone son impact sur la fonction objective. Quelques neurones ne sont efficaces que pour des données spécifiques et pas pour les autres. Ceci est dû à la différence de poids. Par exemple, la base i-LIDS contient des personnes avec des bagages et les neurones qui capturent cette information seront inutiles pour détecter des personnes dans d'autres bases : ils sont incapables d'identifier toute personne sans bagage.

Dans [SLZ+17], un réseau profond est proposé afin d'extraire des caractéristiques robustes des personnes à partir des détections de poses. Le réseau est constitué de deux sous-réseaux : le premier basé sur des couches de convolution pour apprendre les caractéristiques globales de l'image originale. Le deuxième est utilisé pour apprendre des caractéristiques locales à partir des images divisées en six parties du corps de la personne. Les deux sous-réseaux sont combinés à la fin dans une couche de fusion et partagent des paramètres tels que les poids durant l'apprentissage. La sortie du réseau est utilisée comme signature d'image pour évaluer les performances de ré-identification de la personne par le calcul de la distance euclidienne. Cette architecture permet explicitement d'apprendre des caractéristiques robustes des parties du corps des personnes et les adapter pour la recherche de la similarité sauf que la différence de poids entre les deux réseaux peut causer des problèmes lors de la ré-identification.

Puis, dans [LCZH17], les auteurs adoptent le même principe de division de l'échantillon en parties et extraient des caractéristiques de chaque partie. De plus, dans ce travail, chaque partie subit une extraction des caractéristiques en empilant des convolutions sur plusieurs échelles dans chaque couche. Au lieu d'utiliser des parties rigides du corps des piétons, ils proposent d'apprendre à localiser les parties du piéton à travers un réseau de transformateurs spatiaux avec de nouvelles restrictions spatiales. En raison des variations de l'arrière-plan, qui engendrent des difficultés de représentation des parties des piétons, l'apprentissage du corps complet du piéton a été intégré dans le module d'apprentissage des parties du corps.

TABLEAU 2.4 – Comparaison des méthodes de ré-identification par des réseaux profonds DNN

Méthodes	Références	Caractéristiques	Limites
DNN + Filtre de couleur et texture	[WCL+16]	+ Des caractéristiques précises pour les personnes	- Description globale des personnes
DNN + SIFT + LAB color	[WSvdH17]	+Robuste comme caractéristiques	-Complexité de calcul
DNN + description par parties	[XLOW16]	+Robuste contre les changements de pose	-Des neurones inutiles qui ne traitent que des cas spéciaux
DNN + détection de pose	[SLZ+17]	+Robuste à la variation entre les caméras	-La différence de poids entre la détection de pose et des personnes en pose globale dans plusieurs cas des problèmes
DNN + un contexte multi-échelle	[LCZH17]	+Diminuer les corrélations entre les différents noyaux de convolution	-Complexité de calcul

La ré-identification par des réseaux siamois

Les modèles de réseau siamois ont été utilisés dans les tâches de ré-identification de personne en raison du manque des données dans ce domaine de recherche. Ce type de réseau contient généralement deux ou plus sous-réseaux. Ces derniers partagent la même architecture, les mêmes paramètres et les mêmes poids. Le réseau siamois est typiquement employé en paire, quand il s'agit de deux sous-réseaux ou en triplet, quand il s'agit de trois. La sortie du réseau est un score de similarité. Soit $X = x_1, x_2, ..., x_n$ et $Y = y_1, y_2, ..., y_n$ sont un ensemble d'images de personnes; et pour distinguer les paires des négatifs, on adopte l'équation 2.2 :

$$I_s(x_i, x_j) = \begin{cases} positif & si \quad y_i = y_j, \\ négatif & si \quad y_i \neq y_j \end{cases} \tag{2.2}$$

Pour les réseaux de triplet, une fonction d'apprentissage est utilisée afin de créer une marge de distance entre les paires positives et celles négatives. A la sortie du réseau, une couche de softmax est utilisée. La fonction triplet d'erreurs permet d'obtenir une distance moins grande entre les paires positives qu'entre les paires négatives. Si $O_i = (I_i, I_i^+, I_i^-)_{i=1}^N$ soit un ensemble de trois images, avec I_i, I_i^+, faisant référence à la même personne et I_i, I_i^- qui se réfèrent à deux personnes différentes. La figure 2.8 illustre un exemple d'architecture classique basée sur un réseau Siamois triplet.

Les modèles paires : Dans [ZKWM14], le réseau Siamois prend en entrée une paire d'images. Chacune de ces images est connectée à un sous-réseau et elles sont toutes directement connectées à des couches de convolution. Un SVM à la fin du réseau est employé au lieu d'utiliser une couche de softmax dans le but de mesurer la similarité des paires d'images.

Dans [YLLL14], les auteurs divisent les images d'entrée en parties. Le réseau Siamois est construit de façon à apprendre à chercher la similarité entre les paires. Cependant, Chaque partie passe par un réseau indépendant et la fusion se fait au niveau des couches totalement connectées.

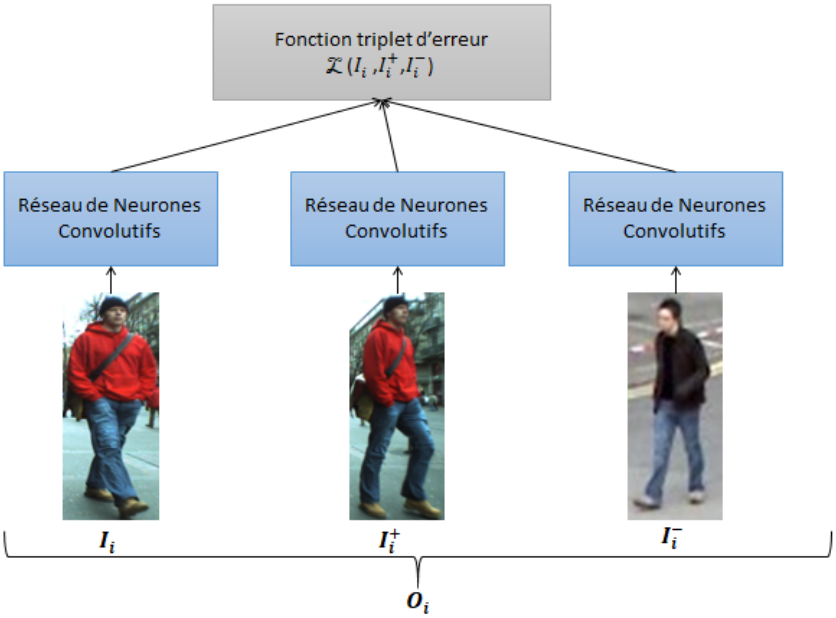


FIGURE 2.8 – Une architecture exemple de réseau Siamois triplet

Ensuite, dans [LZXW14], les auteurs ajoutent un filtre profond pour coder la transformation dans les différentes vues des caméras. Une couche corrective d'appariement est ajoutée par la multiplication des cartes des caractéristiques issues des couches convolutives.

Plus tard, dans [AJM15], les auteurs prouvent aussi l'effet des réseaux Siamois pairs. Le réseau prend en entrée une paire d'images et transmet en sortie des probabilités de similarité entre les images d'entrée. Le modèle commence avec deux couches convolutives qui prennent la paire d'images comme entrée. Les cartes des caractéristiques subissent un traitement afin de diminuer leurs tailles.

Les auteurs de [WWG+17], présentent une architecture CNN basée sur plusieurs échelles et plusieurs parties du piéton. L'architecture prend en entrée deux images de personnes, les divise en parties supérieures et centrales et les convertit en plusieurs échelles. La sortie est un score de similarité. Cette architecture est composée en particulier de quatre sous-réseaux. Le premier prend en entrée l'image complète de taille 200 X 100 et le second prend une image de taille plus petite : 100 X 50. Les deux autres réseaux prennent le haut et le centre de l'image d'entrée. Les quatre sous-réseaux sont composés de deux couches de convolution, deux couches de max pooling, une couche totalement connectée et une autre de normalisation. Chaque réseau retourne une représentation de la personne et un score final est calculé à la fin à partir des comparaisons entre les sorties des quatre sous-réseaux.

Dans [WZL+16], les auteurs ont développé une architecture CNN afin d'apprendre une représentation SIR (single-image representation) pour chaque image et une représentation CIR (cross-image representation) pour l'intermédiaire entre deux. Cette approche est utilisée dans des comparaisons de paires et de triplets d'images. Chaque modèle est composé de sous-réseaux SIR et CIR comme le montre la figure 2.9. En ce qui concerne la comparaison des paires d'images, le SIR utilise la distance euclidienne comme fonction pour le calcul d'erreurs. Quant au CIR, il utilise, lui, une classification binaire par SVM standard pour le calcul d'erreurs. Les auteurs utilisent une combinaison des deux fonctions de calculs d'erreurs. Ensuite, ils partagent les différents paramètres entre les réseaux.

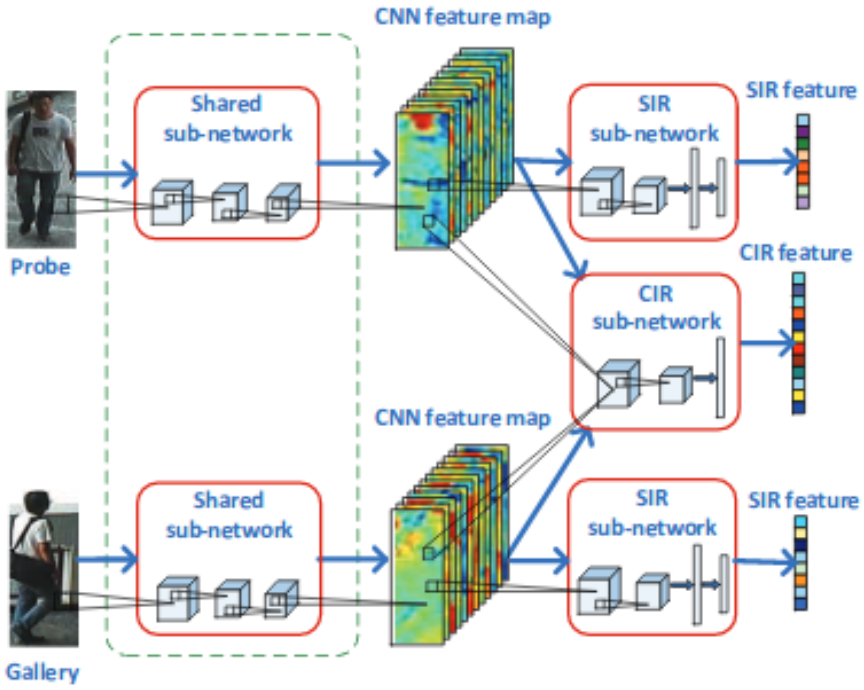


FIGURE 2.9 – Une architecture de comparaison par paire d'images de [WZL+16]

Toutes les architectures utilisées et mentionnées dans cette partie ont typiquement la même structure et sont basées sur des sous-réseaux et différentes fonctions de calcul de similarité. En particulier dans [VSL+16], une nouvelle architecture de réseau siamois est employée à base de LSTM qui sert à extraire l'indépendance contextuelle afin de renforcer les capacités de discrimination des caractéristiques locales puisque l'image est divisée en parties et que chaque partie passe par un LSTM. En fait, chaque image d'entrée est divisée en six parties et est connectée à un sous-réseau. Chaque partie subit une extraction de caractéristiques par deux descripteurs LOMO et Color Names. Puis, chaque vecteur entre dans une LSTM et toutes les LSTMs partagent des paramètres comme les poids. La sortie des LSTMs est combinée et la distance entre les différentes sorties est calculée par une fonction de calcul d'erreurs. Ce réseau basé sur des paires fait l'entraînement avec un algorithme de mini-batch de la descente de gradient stochastique.

Les modèles triplets : Dans [ZLZ+15], les auteurs présentent un réseau supervisé pour la ré-identification dans des vues de caméras disjointes. L'architecture profonde est utilisée pour produire des codes avec la matrice des poids et ce en prenant une taille d'image d'entrée 250 x 100 des personnes. L'objectif de ce codage est de projeter chaque échantillon dans un vecteur binaire. Le réseau est utilisé avec un apprentissage des triplets afin de les forcer à avoir un codage proche en cas de similarité. Pour chaque triplet, les auteurs essayent de maximiser la marge entre les triplets non appariés. Ce réseau profond utilise le Alexnet [KSH12] pré-traîné composé de dix couches : les six premières couches sont des couches de convolution, de pooling et d'activation. Ils utilisent 32, 64 et 128 comme noyaux avec une taille de 5 x 5 pour les trois premières couches de convolution respectivement. Une couche de pooling, de taille 2 X 2, est ajoutée par la suite. Les quatre dernières couches sont composées de deux couches totalement connectées : une couche tangente pour générer la sortie sous forme de code et une dernière couche pour manipuler la taille et le poids du code. Les fonctions d'activation sont des ReLu (Unité de Rectification Li-

néaire) sauf pour la couche de la génération du code qui est à la base de la fonction tangente.

Les auteurs de [CGZ+16] proposent aussi un réseau de comparaison des triplets qui prend en entrée une présentation globale optimale de piétons ainsi qu'une présentation locale par parties. La fusion de deux types de présentations est l'entrée du réseau. La couche de convolution dans le réseau est divisée en quatre parties similaires et chaque partie forme la première couche du réseau indépendant pour apprendre et extraire des caractéristiques d'une partie du corps du piéton. Les quatre sous-réseaux correspondant aux quatre parties du corps sont entraînées sans partage de paramètres. Les sorties des sous-réseaux sont concaténées dans un seul vecteur dans une couche totalement connectée.

D'autres travaux comme [CGC+18] utilisent un algorithme laplacien de graphe structuré dans un réseau profond. Le réseau conçu apprend, au sein d'un réseau siamois du triplet avec la fonction softmax, à maximiser les variations inter-classes d'un piéton à l'autre. Tandis que l'algorithme laplacien de graphe structuré est utilisé pour minimiser les variations inter-classes. Comme les auteurs l'ont souligné, le réseau conçu n'a pas besoin de branche réseau supplémentaire, ce qui rend le processus de formation plus efficace.

Dans [BYH+17], les auteurs proposent un modèle pour générer des caractéristiques globales et locales des personnes. Chaque image des triplets entre dans un sous-réseau CNN pour produire des caractéristiques avec un partage de paramètres entre les sous-réseaux. Les caractéristiques obtenues entrent dans des couches de LSTMs afin d'engendrer de nouvelles représentations avec d'autres caractéristiques. Les LSTMs sont utilisés en raison de leur capacité discriminante pour présenter l'information contextuelle. La sortie de LSTM est introduite dans une autre branche du réseau comportant un average pooling global, une couche entièrement connectée et une couche Softmax pour l'apprentissage global des personnes.

Une approche d'apprentissage profond a été introduite dans [LRL+17], laquelle permet de couvrir l'ensemble des caméras réseau. L'approche proposée vise à chercher la correspondance optimale globale par rapport aux caméras. Les caractéristiques profondes sont générées sur toute la globalité des parties du corps dans un cadre de comparaison des triplets. Chaque image du triplet présente une vue de réseau de caméras. Une fois que les caractéristiques profondes sont produites, la similarité par la fonction de cosinus est utilisée pour obtenir les scores de similarité. Puis, la rétropropagation est adoptée pour obtenir une association finale, globale et optimale.

Les couches de prises de décisions : Même s'il y a plusieurs fonctions utilisées dans la ré-identification des piétons, nous nous limitons aux trois plus utilisées dans l'état de l'art : les fonctions à base de distance euclidienne, le calcul de similarité à base de fonction de cosinus et la fonction softmax.

La distance euclidienne est couramment utilisée comme métrique de calcul de distance. Cette fonction a été adoptée dans plusieurs travaux, exemple [CGZ+16] ; elle est notée par $d(\mathrm{W},\mathrm{O}_i)$ avec $\mathrm{W}=\mathrm{W}_i$; correspond au paramètre du réseau et $\mathrm{F}_w(\mathrm{I})$ représente la sortie du réseau de l'image I. La différence entre les distances est calculée entre les paires appariées et non appariées d'un triplet, par exemple avec l'équation 2.3 :

$$d(\mathrm{W},\mathrm{I}_i) = \parallel \mathrm{F}_w(\mathrm{I}_i) - \mathrm{F}_w(\mathrm{I}_i^+) \parallel^2 - \parallel \mathrm{F}_w(\mathrm{I}_i) - \mathrm{F}_w(\mathrm{I}_i^-) \parallel^2 \tag{2.3}$$

En contrepartie, la fonction de calcul à la base de la fonction cosinus utilisée souvent dans les réseaux Siamois pairs [LRL+17], sert à maximiser sa valeur pour les paires positives et à la réduire pour les cas inverses ainsi qu'à minimiser la valeur de cosinus pour les paires négatives lorsque la valeur est inférieure à la marge (équation 2.4).

TABLEAU 2.5 – Comparaison des méthodes de classification

Fonctions	Références	Fonctions	Caractéristiques
La distance euclidienne et la fonction cosinus	[CGZ+16], [LRL+17]	Calcul de distance : $d(W, I_i)$ et $I(x_1, x_2, y)$	*Apprentissage semi-supervisée *Le nombre des classes n'est pas prédéfini
La fonction softmax	[CGC+18]	Calcul de probabilité : $s(x_i)$	*Apprentissage supervisé *Les classes sont prédéfinies et assignées manuellement

$$I(x_1, x_2, y) = \begin{cases} max(0, cos(x_1, x_2) - m) & si \quad y = 1, \\ 1 - cos(x_1, x_2) & si \quad y = -1 \end{cases} \tag{2.4}$$

La fonction softmax [CGC+18] est aussi utilisée pour calculer la probabilité d'association des paires positives et négatives. Elle est utilisée essentiellement dans les tâches de classification. Elle utilise plusieurs neurones et engendre des sorties normalisées afin de créer une classification $s(x_i)$ par probabilité, par l'équation 2.5 (avec $s \in [0, 1]$).

$$s(x_i) = \frac{exp(x_i)}{\sum exp(x_i)} \tag{2.5}$$

Augmentation de volume : l'augmentation du volume des données dans la tâche de ré-identification est un mécanisme crucial pour l'apprentissage dans les réseaux profonds. Le manque de données d'apprentissage engendre un manque d'échantillons positifs (pairs appariés) et négatifs (pairs non appariés) [HYW+20]. Par conséquent, le réseau ne parvient généralement pas à résoudre le problème de sur-apprentissage. Pour réduire le risque de ce problème, l'augmentation des données est effectuée afin de générer des échantillons artificiels en vue d'augmenter le nombre d'échantillons d'entraînement. Lors d'une ré-identification des personnes, cette opération est généralement effectuée par une simple transformation d'une image de personne de taille $H * W$ en plusieurs distributions uniformes situées dans la plage $[-\alpha H, \alpha H] * [-\alpha W, \alpha W]$ par une valeur aléatoire de α pour chaque image de piétons. Dans [ZZK+17], les auteurs ajoutent une valeur aléatoire au rectangle qui englobe l'objet. Cette stratégie limite les conséquences des problèmes d'occultation. D'autres stratégies ont été adoptées dans [MDRM15] où les auteurs utilisent une transformation linéaire et l'arrière-plan afin de produire d'autres échantillons. De plus, dans [ZKFT17], les auteurs ajoutent de vrais positifs dont l'apparence est similaire aux objets pour augmenter le volume des données d'apprentissage à partir d'autres bases de données.

Actuellement, le réseau GAN [GPAM+14], un style de transfert particulier attire plus l'attention des chercheurs par rapport à d'autres styles de transfert. En fait, ce réseau permet le transfert des échantillons d'un domaine à un autre de façon non supervisée. Ce réseau a montré, depuis son apparition en 2014, des performances élevées.

2.3.2 Discussion et positionnement au niveau de la ré-identification

Les méthodes de ré-identification sont diverses et nous nous sommes limités dans l'état de l'art à ceux qui se réfèrent aux réseaux profonds car elles sont les plus utilisées ces dernières années et présentent les performances les plus élevées.

Les méthodes (exemple [GBT07]) qui considèrent que chaque objet est une classe, souffre du manque de robustesse. Plusieurs travaux ont ajouté d'autres descripteurs ou filtres afin d'extraire plus d'informations. Dans [WCL+16], ils utilisent un filtre de couleur et de texture, dans [WSvdH17], les auteurs utilisent, avec le réseau profond, le descripteur SIFT ; et dans [SLZ+17], les auteurs ont ajouté la détection de pose. Dans une telle stratégie, le réseau peut apprendre des cas particuliers dans certaines classes, ce qui ne permet pas, par conséquent, de trouver le même objet dans d'autres conditions.

Ces méthodes de ré-identification souffrent du manque de volume de données dans des caméras particulières, ce qui engendre une perte de données, redéfinie une autre trajectoire et attribue un nouveau ID à une même personne. Plusieurs auteurs ont utilisé le réseau Siamois soit par paire [ZKWM14], [LZXW14] soit par triplet [ZLZ+15], [CGZ+16] et ce afin de remédier au manque de données. Cependant, ces architectures sont plus complexes car elles présentent deux ou trois fois la même architecture.

Même si la migration vers les réseaux Siamois peut être une solution au manque de données, il est à préciser qu'elle n'est pas la seule. L'augmentation du volume de données, déjà détaillée dans le chapitre précédent, peut avoir lieu par l'ajout d'une valeur aléatoire au rectangle qui englobe l'objet afin de construire d'autres échantillons [ZZK+17] ou d'en ajouter d'autres positifs à partir d'autres bases qui présentent des objets ayant une apparence similaire à celle des objets de la base d'origine [ZKFT17].

En fait, toutes ces méthodes basées sur les réseaux siamois ou d'autres comparent les échantillons un par un aux changements qu'un objet peut subir d'une caméra à une autre, ce qui augmente le risque de l'échec de l'identification.

Par la suite, l'extraction des caractéristiques devient une étape plus critique dans le suivi dans un réseau de caméras car l'objet subit beaucoup de changements d'une caméra à une autre, qui influent directement sur son vecteur de caractéristiques et risque aussi d'augmenter l'écart. L'extraction des caractéristiques se fait de façon globale [ZKWM14] ou partielle [YLLL14]. La première souffre de perte d'informations et la deuxième risque de tomber sur des zones vides ou redondantes. De même, l'information d'apparence s'avère non suffisante ni pour le suivi ni pour la ré-identification et la majorité des travaux [ZKWM14], [YLLL14] et [LZXW14] ne cherchent pas à vérifier l'information temporelle.

Pour conclure, une méthode d'augmentation de données est utile pour la ré-identification et l'extraction de l'information qui doit prendre en considération les informations liées à l'apparence et au mouvement.

2.3.3 Évaluation des méthodes de ré-identification

Mesures de performances : Pour mesurer les performances des architectures de ré-identification des piétons, la courbe de correspondance cumulée (CMC)est la métrique la plus utilisée. Elle permet de calculer le taux d'identification correct des individus. En d'autres termes, la courbe CMC est définie par la probabilité d'identification correcte dans les premiers rangs r ($r = 1, 2, ..., n$) avec n le nombre total d'images modèles utilisées dans la phase de test. La courbe atteint 1 pour $r = n$. Cette métrique n'a de valeur que lorsqu'elle est appliquée sur une base de données.

Les bases de données de ré-identification de personnes sont nombreuses. Certains facteurs doivent être pris en compte pour atteindre un taux de reconnaissance élevé.

Cette tâche est confrontée à des défis dus aux occultations comme par exemple dans la base i-LIDS, et aux changements de luminosité (qui se trouvent dans presque toutes les bases). En outre, la distinction entre l'arrière-plan et l'objet s'avère être une tâche difficile, raison pour laquelle certaines bases (ETHZ, VIPeR, CAVIAR...) fournissent des régions d'intérêt déjà segmentées. Plusieurs bases de données issues de l'état de l'art sont préparées pour évaluer la tâche de ré-identification. VIPeR est la plus difficile. Les bases de données VIPeR, CAVIAR et PRID sont utilisées lorsque deux vues de caméras fixes seulement sont données pour évaluer les performances des méthodes de ré-identification de personnes.

Une brève description est détaillée sur l'ensemble des bases de données les plus populaires.

i-LIDS : Elle contient 476 images de 119 personnes. L'acquisition de la base est faite dans un hall d'aéroport avec des caméras à champs de vue sans chevauchement. La base présente plusieurs défis comme l'occultation, le changement de pose et le changement de luminosité. Un minimum de 2 images et en moyenne 4 images est présenté pour chaque piéton.

CAVIAR : Elle contient 72 personnes, deux vues et seules 50 personnes apparaissent dans les deux. Chaque personne est présentée en 5 images par vue avec variation d'apparence due au changement de résolution, variation de luminosité, occultation et différentes poses.

VIPeR : Elle est constitué de 632 personnes et chaque personne est présentée par deux images issues de deux vues de caméras avec un changement de pose et de luminosité ; les images des personnes sont coupées à 128 X 48 pixels. Elle est considérée comme la base la plus utilisée et la plus connue pour la ré-identification des personnes.

PRID : Cette base de données est spécialement capturée pour la tâche de ré-identification. Elle contient deux lots d'images de 385 et 749 personnes respectivement capturés par deux caméras A et B. Uniquement 200 personnes passent dans les deux caméras.

Market-1501 : Cette base de données est considérée comme la plus large pour la ré-identification puisqu'elle contient 32 643 rectangles de 1501 personnes. Chaque personne est capturée par six caméras au maximum.

2.4 Conclusion

Dans ce chapitre, une étude de la littérature a été présentée. Il est structuré en deux axes : le suivi dans une seule caméra et le suivi dans un réseau de caméras. Dans la première partie, les méthodes de suivi ont d'abord été présentées. Puis, une étude des détecteurs a été réalisée suite au choix d'une méthode de suivi basée sur la détection. Et enfin, une étude de méthode d'évaluation des différentes bases de données a été décrite dans la dernière partie de cette section. De même, le suivi dans un réseau de caméras a été détaillé avec une présentation des différentes méthodes de ré-identification ainsi que les méthodes d'évaluation et les bases de données les plus utilisées pour cette tâche.

La littérature montre que le suivi peut avoir lieu avec plusieurs algorithmes. Le choix de ceux-ci est lié essentiellement aux performances les plus élevées et aux scénarios applicatifs les plus proches du réel. Nous détaillons le suivi dans une seule caméra dans le chapitre 3 et la ré-identification dans le chapitre 4.

Chapitre 3

Le suivi mono-caméra

« *Combien reste impénétrable dans chaque destinée le noyau véritable de l'être, la cellule plastique d'où jaillit toute croissance!* »

Stefan Zweig

Sommaire

3.1 Introduction 38
3.2 Vue globale de la méthode de suivi 38
3.3 Détection et extraction des caractéristiques 40
3.3.1 Réseau de neurones convolutif 40
3.3.2 Définition de la signature 41
3.4 Le suivi 44
3.4.1 Définition d'une tracklet 45
3.4.2 Construction des tracklets 45
3.4.3 Le stage global 47
3.4.4 Étape de mise à jour 47
3.4.5 Minimisation de l'architecture par RNN 48
3.5 Expérimentation 61
3.5.1 Base de données 61
3.5.2 Métriques 63
3.5.3 Résultats et discussions 63
3.5.4 Analyse des erreurs 67
3.6 Conclusion 70

3.1 Introduction

Le suivi mono-caméra est une tâche préliminaire dans le cadre de la ré-identification dans un réseau de caméras. L'objectif du suivi est, dans notre cas, basé sur la détection. Il s'agit de localiser puis associer ces détections pour construire une trajectoire complète tout au long de l'apparition de l'objet dans une caméra. Le suivi présente un sujet crucial dans la vision par ordinateur lequel peut être utilisé dans divers domaines. Un exemple où le suivi intervient, est la comptabilisation et le calcul de flux dans le secteur automobile, à la recherche des informations dans les vidéos indexées dans le domaine de reconnaissance des textes ou de l'assurance de la protection des zones à accès restreint précisément dans le domaine de la vidéo-surveillance. Une signature à chaque objet est attribuée à chaque instant et est recherchée par la suite dans toute la séquence dans le cadre du suivi mono-caméra ou dans un réseau de caméras pour la ré-identification.

Suite à la démonstration faite dans le chapitre précédent, la plupart des algorithmes de suivi souffrent de fragmentation des trajectoires. Par conséquent, différents IDs sont attribués à un même objet. Les causes de ce phénomène sont les occultations entre l'objet et l'environnement, le croisement entre les objets eux-mêmes, la similarité, les défauts de détection, voire le changement de pose ou de luminosité.

L'objectif de notre travail est de définir une signature unique pour chaque objet. Chaque signature est exprimée par un vecteur de caractéristiques qui contient des informations sur son état lesquelles peuvent être les coordonnées de sa position, celles de sa vitesse, de sa taille et de son rapport de taille, etc. L'association de ces détections aboutit à la construction de tracklets dont l'association forme les trajectoires.

Dans ce chapitre, nous détaillons notre méthode de suivi basée sur la détection, la construction de tracklets et leur association. Nous commençons par une section de vue globale du système de suivi dans une une seule caméra. Puis, nous présentons notre méthode de détection basée sur un réseau de neurones convolutif. Ensuite, nous développons notre méthode d'association et de construction de tracklets. Nous commençons par la définition et l'étape de construction des tracklets. Et enfin, nous proposons une étude expérimentale dans le but d'éprouver la robustesse de notre méthode de suivi.

3.2 Vue globale de la méthode de suivi

Dans cette section, une vue globale du système de suivi est présentée. Le suivi peut corriger les erreurs issues de la détection ; et en retour, un bon détecteur peut contribuer à donner de bonnes performances de suivi. Par la suite, nous présentons une nouvelle

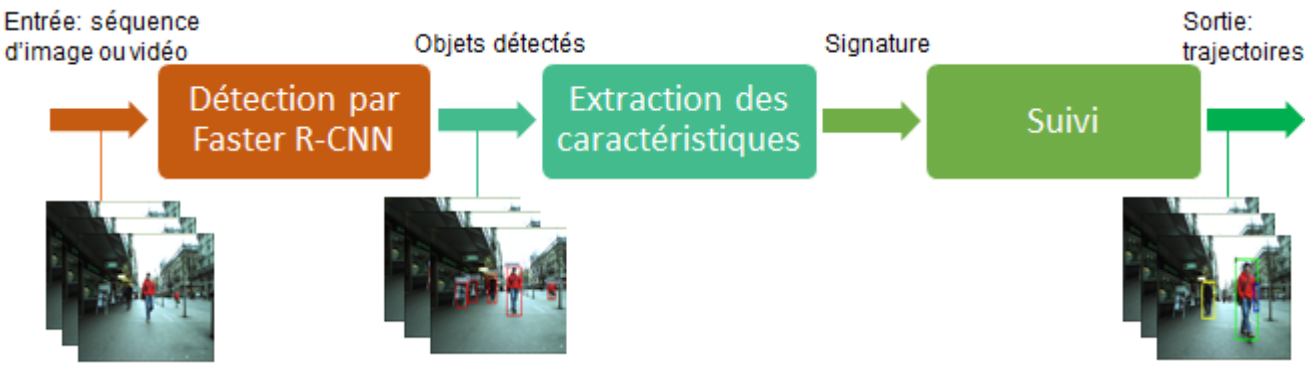


FIGURE 3.1 – Vue globale de l'architecture de suivi dans une mono-caméra

approche du suivi basée sur les tracklets ainsi que sur un réseau de neurones convolutif.

Dans notre thèse, le suivi dépend d'une méthode de suivi par détection. La figure 3.5 illustre les étapes du suivi dans une seule caméra.

L'entrée de cette architecture peut être une séquence d'images ou une vidéo. Cette entrée passe par un réseau CNN qui retourne des rectangles englobant l'objet ou un faux positif. En fait, comme tout système, le réseau convolutif retourne de vrais positifs qui sont les rectangles englobant les objets à détecter, de faux positifs représentés par les rectangles englobant non seulement l'objet mais également des parties de l'arrière-plan et de faux négatifs qui sont des objets à détecter mais qui ne l'ont pas été. La sortie du réseau passe par le bloc d'extraction des caractéristiques afin d'extraire des informations liées à la taille, la position et la vitesse... Puis, ces détections sont associées pour construire des mini-trajectoires intitulées "tracklets". Ces dernières, ayant des signatures similaires, sont à leur tour associées à l'étape que nous appelons "stage global". Une fois que les trajectoires sont construites, une étape de mise à jour aura lieu afin de corriger ces trajectoires. L'algorithme 1 illustre les différentes étapes de cette architecture de suivi.

Notre contribution dans cette partie se résume en trois points :

— Une contribution au niveau de l'approche : la définition d'une signature unique pour chaque objet détecté et l'ajout d'une phase de mise à jour afin de corriger les erreurs de construction.

— Une contribution pour minimiser l'architecture : l'utilisation d'un réseau RNN qui nous permet d'avoir la tâche de construction et de mise à jour en parallèle.

— Une contribution au niveau de l'application : une architecture robuste pour faire le suivi avec une caméra fixe ou mobile pendant le jour ou la nuit.

Fonction Suivi multi-objets basé sur CNN et tracklets(I : **Frame**) : **Trajectoires**
résultat : **trajectoire**
[l :numéro de frame dans une vidéo]
[M :numéro d'objet détecté]
Répéter
[Détection]
Pour (k=1 ; k ≤ l; k++) **faire**
[détection d'objet dans N frames]
Fin Pour
[Extraction des caractéristiques]
Pour (i=1 ; i ≤ M ; i++) **faire**
[extraction des caractéristiques]
[définition d'une signature pour chaque objet]
Fin Pour
[Suivi]
Construction des tracklets Attribution des ID Association des tracklets
jusqu'à ce que (Construction des trajectoires ;)
Retourner résultat ;
Fin

Algorithme 1 – Algorithme de suivi dans une mono-caméra

3.3 Détection et extraction des caractéristiques

La détection est une tâche primordiale dans le suivi des objets. Souvent, il s'agit de la première information que nous pouvons extraire. Cette information peut être sous forme de boîte englobant l'objet cible ou de points caractéristiques... L'algorithme retourne la segmentation de l'objet par rapport à son arrière-plan ou à sa position.

Les performances varient énormément d'un détecteur à un autre comme il a été prouvé dans le chapitre de l'état de l'art. En effet, cette dégradation est influencée par le nombre des classes de l'objet à détecter (voiture, bus, piéton...). Dans une même classe, elle est influençable par la variété inter-classe qui engendre une variété de formes, de textures et de couleurs. Et dans une même catégorie, les performances sont influençables par le changement de luminosité, d'échelle, ou de position dans l'image et l'angle de vue. Par conséquent, trouver un détecteur qui puisse surmonter tous ces défis est une tâche très complexe. Avec l'évolution des algorithmes de détection et à partir de nos comparaisons, nous allons nous référer à un détecteur CNN qui présente des performances élevées et des défauts que nous estimons pouvoir corriger.

3.3.1 Réseau de neurones convolutif

Les réseaux de neurones convolutifs présentent les meilleures performances selon nos comparatifs de l'état de l'art. Il est constitué de deux parties, une de convolution et une autre de classification.

D'ailleurs, la convolution est un outil mathématique simple très utilisé dans le domaine de traitement d'images, adoptée pour résoudre les problèmes de reconnaissance d'images. La convolution a un effet de filtrage qui parcourt la totalité de l'image par une fenêtre glissante avec un pas précis. A la sortie, une carte des caractéristiques est créée avec des dimensions plus petites que celles de l'image d'entrée. Cette procédure est répétée plusieurs fois et l'entrée de chaque niveau est la sortie du niveau précédent avec une diminution de calculs. À la sortie de cette partie convolutive, un vecteur contenant les caractéristiques les plus pertinentes est créé et chacune des valeurs est connectée à un neurone de la partie de classification. Cette dernière contient des neurones entièrement connectés et agit comme un réseau de neurones "MLP" qui permet de classifier les images.

L'entraînement des réseaux de neurones convolutifs consiste à calculer des poids. En effet, le CNN prend en entrée une image et prédit en sortie la classe, à savoir que les classes sont connues et prédéfinies avec les images d'apprentissage. Durant l'apprentissage, l'entraînement du CNN s'est fait avec des images dont les classes sont déjà connues. Par conséquent, les poids sont mis à jour selon la véracité des résultats, ce qui est appelé la rétro-propagation du gradient de l'erreur. Lors de la phase de validation, des images que le système ne connaissait pas, sont identifiées et un taux de bonne classification est déterminé afin de calculer la précision du modèle.

La littérature présente une diversité de réseaux de neurones convolutifs. Dans cette thèse, le détecteur Faster-RCNN introduit dans le papier [Gir15] est utilisé pour détecter nos objets. Exceptionnellement, ce réseau a un module "Region Proposal Network" (RPN) qui permet de proposer des régions d'intérêt (ROI) à l'intérieur du réseau. L'image d'entrée entre dans le réseau. Le RPN propose les ROIs qui sont traités au niveau de la cinquième couche de convolution et sont classifiés par la deuxième partie du réseau (cf.figure 3.2). Le Faster R-CNN est un concept capable d'être appliqué sur plusieurs architectures CNN. Nous utilisons dans cette partie le réseau VGG16 [HMD15]. L'apprentis-

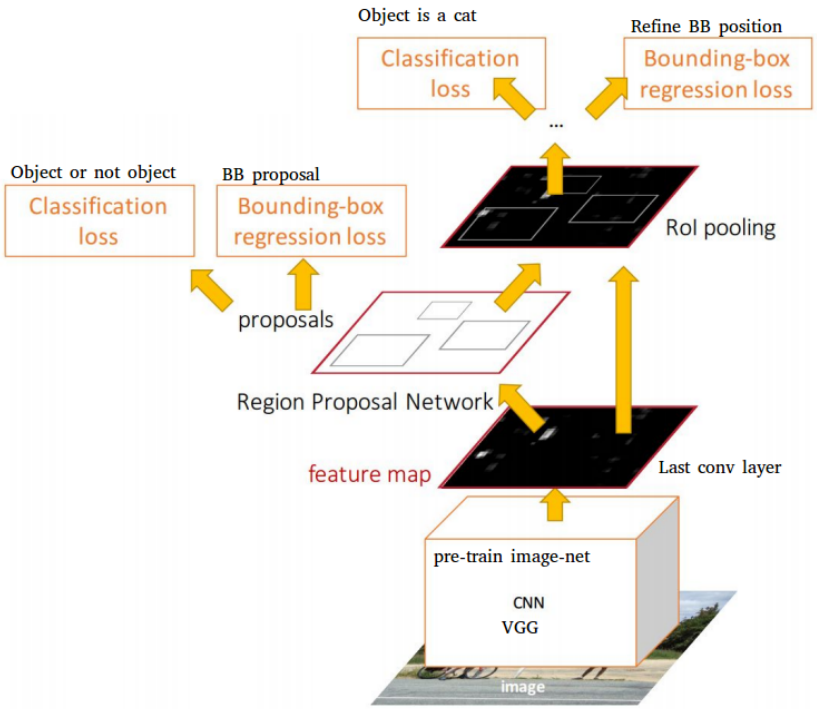


FIGURE 3.2 – Architecture du Faster R-CNN [Gir15]

sage est effectué sur 10000 itérations par une descente de gradient stochastique avec un taux d'apprentissage constant égal à 0.001. Les poids sont pré-entraînés avec les données et les classes d'ImageNet [KSH12].

3.3.2 Définition de la signature

Le but de notre algorithme de suivi est d'associer les détections issues d'images successives pour construire une trajectoire complète tout en cherchant une cohérence dans l'identifiant de chaque détection. Suite à nos comparaisons, l'algorithme de suivi exploite les informations liées à l'apparence et celles liées au mouvement. Chaque objet détecté est représenté par un vecteur qui exprime son état.

La sortie du détecteur est un rectangle englobant que nous pouvons exploiter afin de déterminer la position et la taille de l'objet détecté. Il peut être aussi utilisé pour extraire de l'information liée à la vitesse ou prédire la prochaine position. Les caractéristiques d'apparence sont aussi extraites de ce rectangle.

Extraction des caractéristiques d'apparence

La description globale d'une image est un vecteur unique issu de la combinaison de toutes les caractéristiques de l'image qui a une taille fixe. Plus précisément, les caractéristiques peuvent être une combinaison de vecteurs localement extraits ou des statistiques globales de l'image entière.

Les mesures statistiques de chaque pixel permettent de décrire et de présenter une image de façon globale. Les histogrammes RGB, HSV ou YCbCr des différents espaces de couleurs sont normalisés afin de limiter l'impact de la variation de l'éclairage. D'autre part, les moments de l'image peuvent en donner une information globale. La variance ou

TABLEAU 3.1 – Les moments statistiques et leurs formules de calcul

Moments statistiques	Formules
Énergie	$E = \sum_n p^2(n)$
Entropie	$En = -\sum^n p(n) log p(n)$
Contraste	$\frac{max(n) - min(n)}{max(n) + min(n)}$
Moyenne	μ_1
Moment d'ordre k	$\mu_k = \sum^n n^k p(n)$

la moyenne peut être directement calculée pour l'image ou pour l'image filtrée. Le tableau 3.1 illustre quelques moments statistiques et leurs formules.

Une description globale est développée dans cette partie de la thèse basée sur l'histogramme de couleurs. En réalité, l'histogramme sert à mesurer la distribution des couleurs dans une image. La présentation des couleurs pour définir un objet est faite selon plusieurs paramètres qui peuvent être la couleur des cheveux, des vêtements, des chaussures ou leur combinaison. Ces paramètres sont très sensibles au changement de l'angle de vue et de la luminosité ; par exemple, la tête d'une personne ne présente pas les mêmes caractéristiques, vue de face, de derrière ou de côté. Pour lutter contre ces variations, nous recherchons l'information la plus pertinente que le changement de vue n'affecte pas. Les paramètres les plus résistants aux changements de vue sont les couleurs du haut et du bas des vêtements. Par conséquent, les histogrammes de couleurs prennent en considération la globalité de la personne.

L'histogramme de couleurs est un représentatif de la distribution des couleurs d'une image. Plusieurs espaces de couleurs sont utilisés dans l'état de l'art et, après comparaison, l'espace couleur HSV (teinte, saturation et valeur) et l'espace RGB (rouge, vert et bleu) donnent les meilleurs résultats en test. Les équations 3.1, 3.2 et 3.3 suivantes montrent la conversion RGB à HSV :

$$V = max(R, G, B); \tag{3.1}$$

$$S = \begin{cases} \frac{V - min(R,G,B)}{V} & = si V \neq 0 \\ 0 & = sinon \end{cases} \tag{3.2}$$

$$H = \begin{cases} \frac{60(G-B)}{V - min(R,G,B)} & = si V = R \\ \frac{120 + 60(B-R)}{V - min(R,G,B)} & = si V = G \\ \frac{240 + 60(R-G)}{V - min(R,G,B)} & = si V = B \end{cases} \tag{3.3}$$

Avec les deux valeurs de S (saturation) et V (valeur) qui varient de 0 à 255 et la valeur H (teinte) qui varie de 0 à 360, les trois niveaux de couleurs de l'histogramme HSV peuvent être observés dans la figure 3.4.

Les trois dimensions de l'histogramme HSV sont calculées sur 15 bins. Chaque histogramme contient 15^3 ou 3375 éléments. Ces histogrammes sont normalisés selon le nombre de pixels détectés dans l'arrière-plan. Cette étape critique permet au système de maintenir sa robustesse par rapport à la différence de taille entre les objets détectés. Un exemple d'image dans l'espace HSV à partir d'une image RGB est illustré dans la figure 3.4.

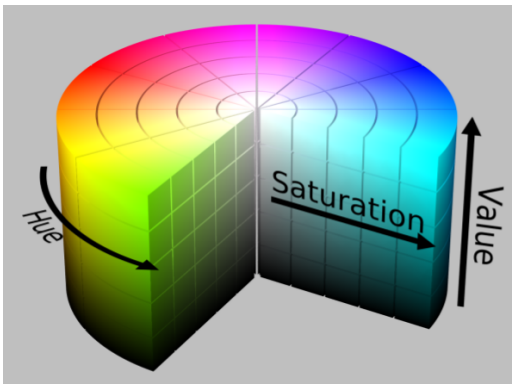


FIGURE 3.3 – Une représentation 3D de l'espace de couleurs HSV

FIGURE 3.4 – Un exemple de transformation de l'espace couleur RGB en HSV

Extraction des caractéristiques par filtre de Kalman

Une comparaison entre le filtre kalman et le filtre Bayes montre que ce dernier présente plusieurs complexités au niveau des systèmes linéaires entachés par un bruit Gaussien : par conséquent, le filtre Kalman nous semble la solution la plus optimale. Dans notre travail, la matrice de covariance de la Gaussienne représente son état. Quant à la moyenne de celle-ci, elle exprime le vecteur d'état de l'objet. Ce filtre permet de faire évoluer la Gaussienne de l'objet au cours du temps et l'évolution se manifeste en deux étapes (la prédiction et la mise à jour) :

L'étape de prédiction décrite par l'équation 3.4 présente la dynamique de l'objet. Cette équation est basée sur le modèle Gaussien de l'évolution linéaire.

$$x_{k|k-1} = \mathrm{F}_k x_{k-1|k-1} + w_{k-1} \tag{3.4}$$

Avec : $x_{k-1|k-1}$: le vecteur d'état avant prédiction de taille $n * 1$.

$x_{k|k-1}$: le vecteur d'état après prédiction de taille $n * 1$.

F_k : la matrice de transition de taille $n * n$.

w_{k-1} : le vecteur de taille $n * 1$ du bruit Gaussien de moyenne nulle et de covariance Q_k de taille $n * n$.

A partir de l'état de l'objet $x_{k-1|k-1}$ et sa covariance $\mathrm{P}_{k-1|k-1}$, l'état $x_{k|k-1}$et la covariance $\mathrm{P}_{k|k-1}$sont prédits selon les deux équations 3.5 et 3.6.

$$x_{k|k-1} = \mathrm{F}_k x_{k-1|k-1} \tag{3.5}$$

$$\mathrm{P}_{k|k-1} = \mathrm{F}_k \mathrm{P}_{k-1|k-1} \mathrm{F}_k^{\mathrm{T}} + \mathrm{Q}_k \tag{3.6}$$

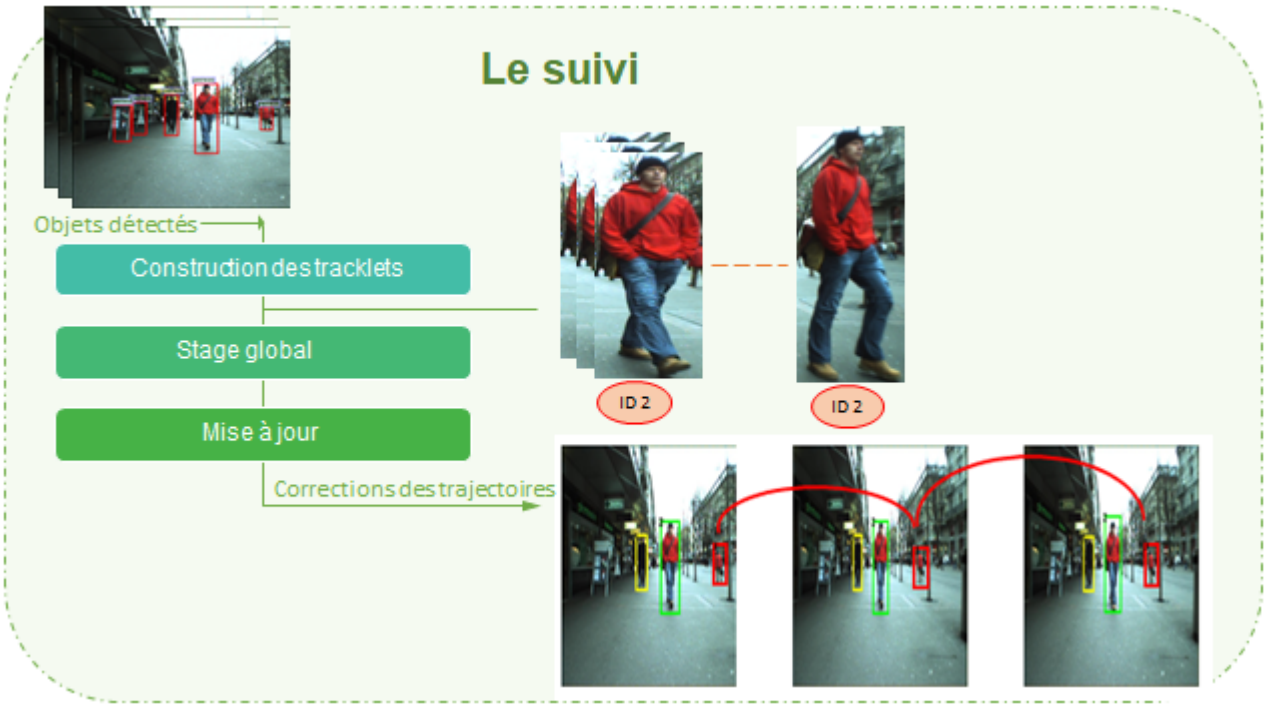


FIGURE 3.5 – Vue globale de l'architecture de suivi dans une mono-caméra

Ensuite, l'étape de mise à jour sert à corriger la covariance $P_{k|k-1}$ et l'état $x_{k|k-1}$ à partir des observations issues de la détection selon l'équation 3.7 :

$$z_k = H_k x_k + v_k \tag{3.7}$$

Avec z_k : vecteur d'observation de taille $m * 1$
H_k : matrice d'observation de taille $m * n$
v_k : vecteur de bruits issus des erreurs de mesure. C'est un bruit Gaussien de moyenne nulle et de matrice de covariance R_k, de taille $m * m$.

Par la suite, les équations 3.8 et 3.9 permettent de corriger l'état $x_{k|k}$ et la covariance $P_{k|k}$ calculés précédemment.

$$x_{k|k} = x_{k|k-1} + K_k(z_k - \widehat{z_k}) \tag{3.8}$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1} \tag{3.9}$$

3.4 Le suivi

Après la phase de détection et d'extraction des caractéristiques, nous lions les détections afin de construire nos tracklets. Puis, dans une phase globale, les tracklets sont associées afin de construire des trajectoires qui sont à leur tour corrigées dans une phase de mise à jour (cf. figure 3.5).

En fait, ce suivi basé sur des détections permet de contrôler l'entrée et la sortie de chaque objet; ce qui nous permet, par la suite, de manager l'attribution des IDs. Mais, comme tous les détecteurs, le Faster R-CNN produit de faux positifs et de faux négatifs. Dans notre nouvelle approche de suivi, nous proposons des solutions qui permettent de minimiser et de restreindre les défauts du détecteur afin d'avoir un suivi robuste et, en

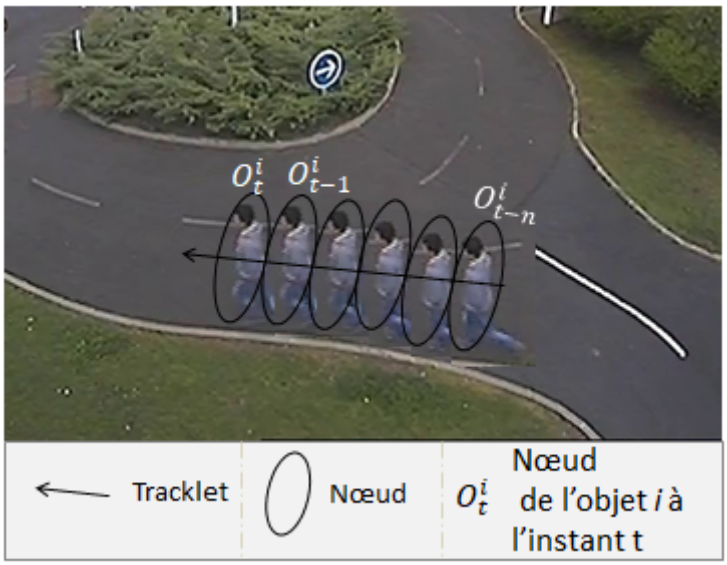


FIGURE 3.6 – L'ensemble des nœuds constitue une tracklet

même temps, de limiter les faux positifs et prédire la position d'objets à détecter et qui ne l'ont pas été.

3.4.1 Définition d'une tracklet

Avant d'entamer les étapes du suivi, nous définissons la notion des tracklets. En effet, chaque détection est considérée comme un nœud O^i où i est l'ID de l'objet (cf. figure 3.6) et une tracklet est l'ensemble de ces détections. Chaque objet i apparaît sur N frames successives avec le même ID du premier moment où il apparaissait t_s^i jusqu'au moment de sa disparition t_f^i. De plus, chaque nœud O_t^i est une détection au moment t avec $t \in [t_s^i, t_f^i]$ et il est défini par un vecteur $X_t^i \in \mathbb{R}^N$ avec $N = 4$ et $X_t^i = (x, y, w, h)$ qui sont les quatre paramètres définissant le vecteur d'état issu du détecteur. En outre, la taille de la tracklet est influencée par les occultations et les objets non détectés. D'ailleurs, la tracklet peut être fragmentée lorsqu'il n'y a pas de détection sur une ou plusieurs images. De même, l'occultation peut aussi perturber la construction de la tracklet, ce qui engendre plusieurs tracklets avec différents IDs. Pour remédier à tous ces problèmes, nous mettons en considération ces hypothèses lors de l'association, qu'il s'agisse de détections ou de tracklets.

Ainsi, une tracklet T^i est définie par l'apparence, la forme et le mouvement : $T^i = A^i, S^i, M^i$.

3.4.2 Construction des tracklets

Dans cette étape, nous collectionnons des détections afin de construire une tracklet. Notre méthode d'association est basée sur le chevauchement des détections et le calcul de la valeur moyenne d'affinité. D'abord, nous fixons le nombre de frames nécessaire à la définition d'une tracklet. Ensuite, pour vérifier le chevauchement entre les différents nœuds, nous suivons l'équation 3.10 :

$$Overlap_{(t,t-1)} = \frac{O_t^i \cap O_{t-1}^i}{Area(O_t^i)} \tag{3.10}$$

O_t^i réfère à l'objet détecté à l'instant t et O_{t-1}^i réfère à l'objet détecté à l'instant $t-1$.

Nous utilisons la comparaison par chevauchement car elle permet de distinguer les vrais positifs des faux positifs. Plus précisément, les tracklets de faibles dimensions sont négligées car nous supposons qu'il ne s'agit pas d'un objet mais plutôt de fausses alarmes.

D'autre part, une comparaison des apparences est utilisée pour associer les détections. Nous utilisons l'histogramme de couleurs HSV qui présente des performances élevées et faciles à implémenter. La similarité entre deux détections est calculée par la méthode de distance Bhattachayya [WN07]. La similarité est notée $d(f_{\mathrm{HSV}_i}, f_{\mathrm{HSV}_j})$ et le vecteur de caractéristiques par $h(O_t^i, O_{t-1}^j)$. Pour déterminer le score d'apparence entre deux nœuds, l'équation 3.11 est adaptée :

$$h(O_t^i, O_{t-1}^j) = d(f_{\mathrm{HSV}_i}, f_{\mathrm{HSV}_j}) \tag{3.11}$$

Cette équation permet de prendre deux nœuds et de vérifier s'il s'agit du même objet.

En fait, lors de l'apparition de l'objet dans la scène $v^i(t) = 1$; sinon, $v^i(t) = 0$. Par conséquent, si $v^i(t) = 1$, alors chaque objet est noté par : $O_t^i = (p_t^i, s_t^i, v_t^i)$

avec : p_t^i, la position que l'objet occupe à l'instant t.

s_t^i : la taille de l'objet.

v_t^i : la vitesse.

Ensuite, une tracklet est définie par $T^i = x_k^i \mid v_t^i = 1.1 \leqslant t_s^i \leqslant k \leqslant t_f^i \leqslant t$. Le score d'une tracklet est calculé par l'équation 3.12 :

$$s(T^i) = \frac{1}{L} \sum_{k \in [t_s^i, t_f^i], v_k^i = 1} \wedge(T^i, z_k^i) * max((1 + \beta log \frac{(L-w)}{L}), 0) \tag{3.12}$$

Avec : β : Valeur de contrôle de la performance du détecteur.

L : La longueur de tracklet.

w : Le nombre de frames où il n'y a pas de détection avec $w = t_f^i - t_s^i + 1 - L$.

$\wedge(T^i, z_k^i)$ "Average affinity score" : Le score moyen d'affinité est calculé entre une tracklet et la mesure d'un détecteur.

En effet, la mesure d'affinité $\wedge(X,Y)$ permet d'associer les tracklets et tracklet-détection et elle est calculée selon l'équation 3.13 avec X et Y peut être détection ou tracklet :

$$\wedge(X,Y) = \wedge^A(X,Y) \wedge^S(X,Y) \wedge^M(X,Y) \tag{3.13}$$

Avec $\wedge^A(X,Y)$: l'affinité d'apparence calculée à partir de l'histogramme de couleur par l'équation 3.14 :

$$\wedge^A(X,Y) = \frac{f_{\mathrm{HSV}}(x).f_{\mathrm{HSV}}(y)}{\| f_{\mathrm{HSV}}(x) \| \| f_{\mathrm{HSV}}(y) \|} \tag{3.14}$$

$\wedge^S(X,Y)$: L'affinité de forme calculée par les coordonnées de chaque objet détecté par l'équation 3.15 :

$$\wedge^S(X,Y) = exp(-\frac{h_x - h_y}{h_x + h_y} + \frac{w_x - w_y}{w_x + w_y}) \tag{3.15}$$

$\wedge^M(X,Y)$: L'affinité de mouvement calculée à l'aide du filtre de kalman par l'équation 3.16 :

$$\wedge^M(X,Y) = N(P_x^{tail} + v_F^x \Theta; P_y^{head}, O^F) * N(P_y^{head} + v_B^y \Theta; P_x^{tail}, O^B) \tag{3.16}$$

Avec $xtail$: la dernière position, $yhead$, la première position et Θ la différence de frames.

3.4.3 Le stage global

Dans le stage global, nous avons l'association des détections notée "tracklet" comme entrée. Le but de cette étape est d'associer ces tracklets afin d'avoir des trajectoires. En fait, lors de la construction des tracklets, ces dernières peuvent subir une fragmentation et, par conséquent, le même objet peut avoir plusieurs tracklets; ce qui nécessite un stage global pour associer ces fragments. L'association est basée sur une comparaison des signatures. Nos contributions à ce niveau se manifestent dans la méthode de suivi qui prend en considération quelques défis à relever. En d'autres termes, lors de l'association, nous essayons de résoudre les problèmes suivants :

— Les phénomènes naturels : le contrôle d'entrée et de sortie des objets dans la scène.

— Le croisement entre les objets de même classe

— L'occultation entre l'objet et l'arrière-plan ou entre les objets de classes différentes.

Ces problèmes provoquent le changement des IDs entre les objets et la fragmentation des trajectoires. Premièrement, nous traitons le problème naturel d'entrée et de sortie des objets. Afin de limiter les défauts qu'un objet peut provoquer par sa disparition ou son entrée pour la première fois, nous vérifions le nombre de nœuds dans chaque tracklet. Plus précisément, si le nombre de nœuds manquant dans une tracklet dépasse le nombre minimal exigé pour construire une tracklet, nous concluons que l'objet est sorti de la scène et son ID n'est plus attribuée. Par conséquent, par cette stratégie, nous managerons la sortie des objets et leur entrée.

Deuxièmement, afin de remédier aux erreurs de croisement entre les objets, nous comparons les apparences des tracklets lors des premières détections après le croisement. Nous comparons également leur chevauchement par l'équation 3.17 :

$$Overlap_{(i,j)} = max\left(\frac{O_t^i \cap O_t^j}{Area(O_t^i)}, \frac{O_t^i \cap O_t^j}{Area(O_t^j)}\right) \tag{3.17}$$

Troisièmement, nous prenons en considération les problèmes d'occultation qui sont assez nombreux dans la plupart des scènes où l'objet se cache pendant une durée déterminée derrière un autre objet de l'arrière-plan. Dans ces cas, les trajectoires subissent généralement une coupure lors de cet événement. Pour y remédier, nous vérifions la similarité entre le dernier nœud de la tracklet numéro 1 et celui de la tracklet numéro 2. Nous cherchons la similarité par la méthode Hungarian [Kuh55].

3.4.4 Étape de mise à jour

Dans l'étape de mise à jour, nous corrigeons les trajectoires issues de l'étape globale. D'ailleurs, si le détecteur produit de faux négatifs, c'est à dire que l'objet n'est pas détecté à un certain moment t. Par conséquent, avec des nœuds manquant,la trajectoire ne peut être correctement construite; c'est pourquoi, à cette étape de l'architecture, nous prédisons la position des non détections. Un exemple de correction de trajectoire par interpolation dans la figure 3.7.

La correction dans cette figure se fait par une interpolation entre l'instant $t-1$ et $t+1$. Nous utilisons les deux nœuds avant et après l'absence de détection. L'équation 3.18 détermine la taille de l'objet manquant.

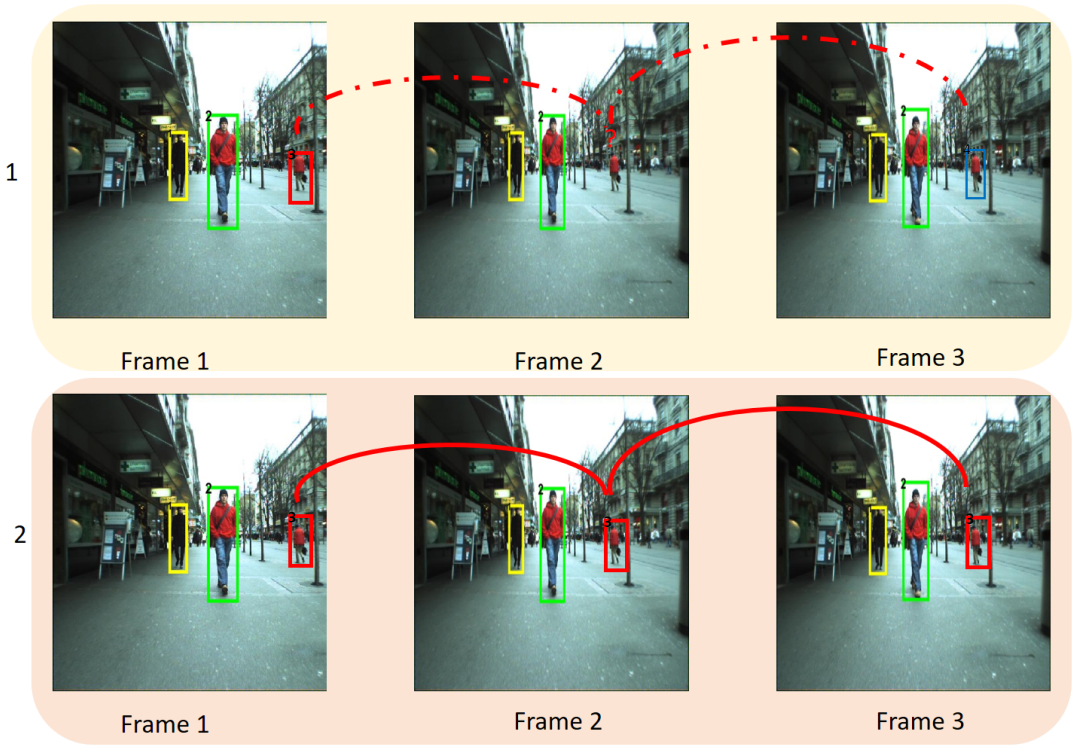


FIGURE 3.7 – Correction d'un objet non détecté par une interpolation

$$Size_t = \frac{S^i_{t-1} \cap S^i_{t+1}}{2} \tag{3.18}$$

Et nous utilisons l'équation 3.19 pour déterminer la position de l'objet avec X_t et son état à l'instant t qui inclut l'abscisse x et la coordonnée y.

$$Position_{X_t} = \frac{X^i_{t-1} \cap X^i_{t+1}}{2} \tag{3.19}$$

A cette étape, nous vérifions aussi s'il y a des détections qui ne sont pas associées et nous les comparons aux signatures des tracklets les plus proches afin de les attribuer à des trajectoires.

Après cette étape de mise à jour, nous aurons des trajectoires avec des signatures uniques pour chaque objet.

3.4.5 Minimisation de l'architecture par RNN

Selon ce que nous avons démontré dans la section précédente, les tracklets permettent d'améliorer la qualité du suivi en limitant quelques problématiques. Ceci est dû au pouvoir des tracklets qui permettent de prendre en considération les états précédents afin de prédire les états futurs tout en éliminant les faux positifs et de prédire les objets non détectés. De là, notre architecture, qui contient des blocs séquentiels, peut faire face à des occultations et des changements de pose. Dans cette section, nous essayons de minimiser cette architecture tout en gardant les mêmes défis. Pour cela, nous proposons une nouvelle architecture à base d'un réseau récurrent (RNN) qui nous permet de minimiser les erreurs par la rétro-propagation et apprendre l'association des détections par les états sauvegardés dans les couches cachées.

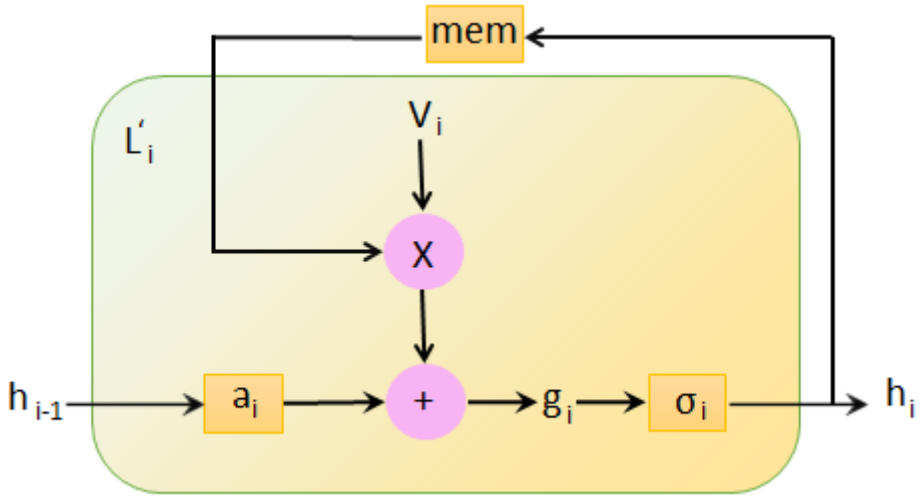


FIGURE 3.8 – La propagation de l'information par un passe-avant d'une couche i du RNN

A cet égard, nous rappelons le principe des RNNs et nous mettons en valeur spécialement le LSTM, cas particulier du réseau récurrent.

Les réseaux de neurones récurrents

Un réseau de neurones récurrent, contrairement à d'autres réseaux de neurones, contient au moins un cycle dans son graphe de connexion. Ce réseau a été développé pendant les trente dernières années et il a présenté des performances assez élevées dans plusieurs domaines. Il présente des performances particulières lorsque les vecteurs d'entrée ont une indépendance spatiale ou temporelle. Effectivement, il introduit un mécanisme qui permet de mémoriser des états précédents afin de les introduire à nouveau dans le réseau. Il influence de façon indirecte les états futurs. En fait, il faut ajouter une matrice d'interconnexion $V_i \in \mathbb{R}^{n_i * n_i}$ par rapport au réseau MLP ordinaire afin d'obtenir une couche $L_i^{'}$. Pour un vecteur $x(t)$ avec $t \in [1, t_f]$, il y aura un vecteur de sortie z_t avec $t \in [1, t_f]$ et des initialisations $h_o(t) = x(t)$ et $h_i(0) = 0$ avec $i \in [1, N]$. Puis, ces étapes sont appliquées récursivement à chaque instant. L'équation 3.20 illustre l'application de la transformation affine a_i du vecteur de sortie de la couche précédente et l'ajoute à V_i appliqué au vecteur de l'état précédent. L'équation 3.21 réfère à l'application de la fonction de transfert au résultat.

$$g_i(t) = W_i * h_{i-1}(t) + V_i * h_i(t-1) + b_i \tag{3.20}$$

$$h_i(t) = \sigma_i(g_i(t)) \tag{3.21}$$

La figure 3.8 illustre la liaison entre l'entrée et la sortie ainsi que son impact sur la propagation de l'information. Les couches de classification peuvent être des couches inspirées de MLP. Par contre, celles de calcul d'erreurs sont exécutées, soit à chaque pas de temps afin de prendre en considération toutes les décisions, soit à la fin pour prendre une décision finale.

La rétro-propagation pour les RNNs (cf. figure 3.9) est légèrement différente par rapport à la version de [Wer90] du MLP. En fait, traditionnellement pour tous les pas de temps, $t \in [1, t_f]$ les dérivées des fonctions composées sont appliquées. En fait, le gradient, lors de la passe-arrière, se rétro-propage de la même façon qu'en MLP. Cependant,

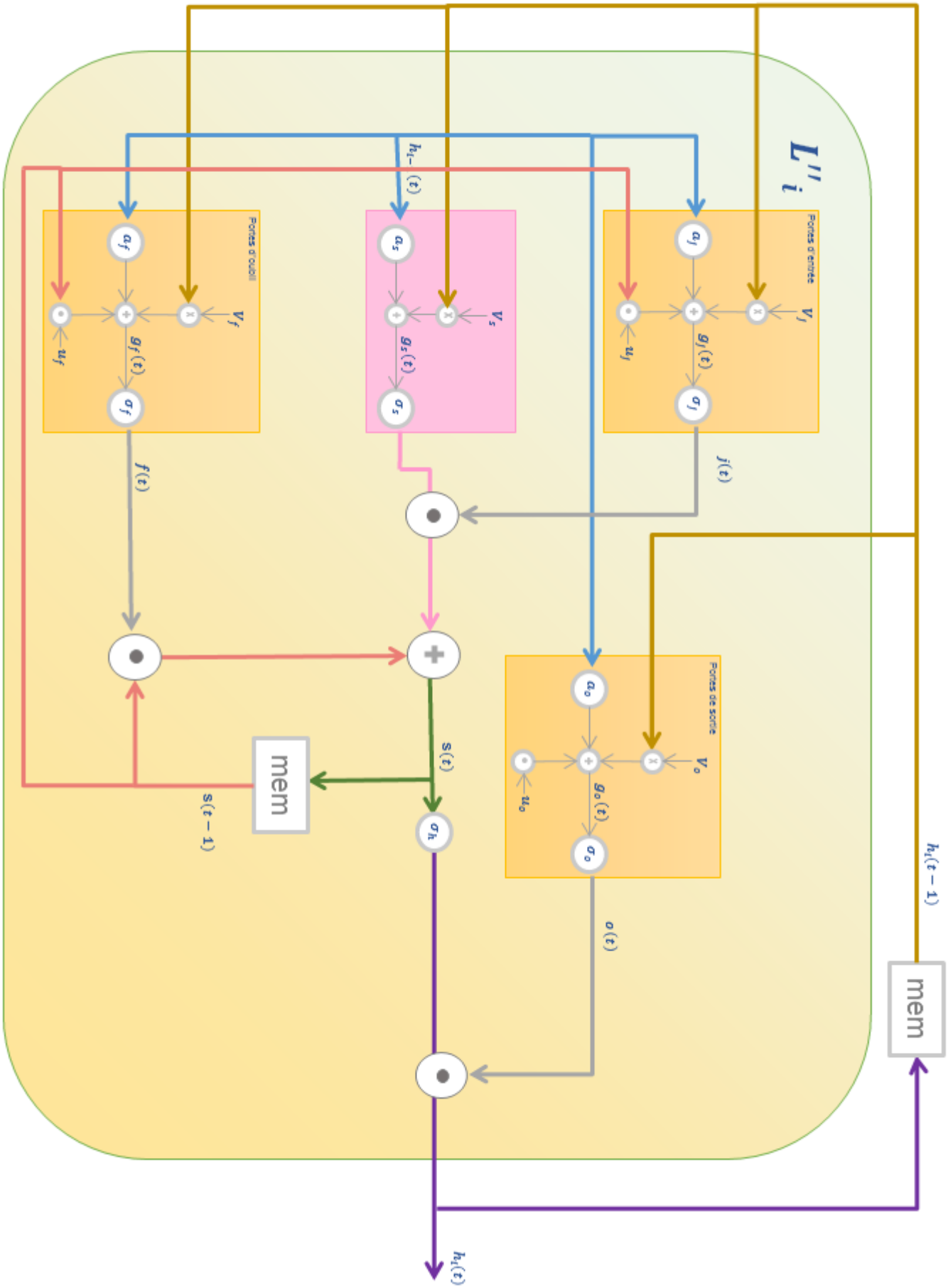


FIGURE 3.9 – La rétro-propagation du gradient par un passe-arrière d'une couche i du RNN

dans le cas du RNN, le lien récurrent $\varepsilon_i(t+1)$ est également rétro-propagé par l'équation 3.22 de la rétro-propagation. Les dépendances au temps apparaissent :

$$\frac{\partial C}{\partial g_i}(t) = J_{\sigma j}(g_i(t)) * \left(\frac{\partial C}{\partial h_i}(t) + V_i^T * \frac{\partial C}{\partial g_i}(t+1)\right) \tag{3.22}$$

Avec $\partial g_i(t_f+1) = 0$ et l'équation 3.23 illustre la dépendance au temps.

$$\frac{\partial C}{\partial h_{i-1}}(t) = W_i^T * \frac{\partial C}{\partial g_i}(t) \tag{3.23}$$

Lorsque l'ensemble des pas du temps sont achevés, la dérivée partielle est déterminée pour chaque couche i avec $i \in [1, N]$ en ajoutant les pas du temps par les équations 3.24, 3.25 et 3.26.

$$\frac{\partial C}{\partial b_i} = \sum_{t=1}^{t_f} \frac{\partial C}{\partial g_i}(t) \tag{3.24}$$

$$\frac{\partial C}{\partial W_i} = \sum_{t=1}^{t_f} \frac{\partial C}{\partial g_i}(t) * \frac{\partial g_i}{\partial W i}(t) = \sum_{t=1}^{t_f} \frac{\partial C}{\partial g_i}(t) * (h_{i-1}(t))^T \tag{3.25}$$

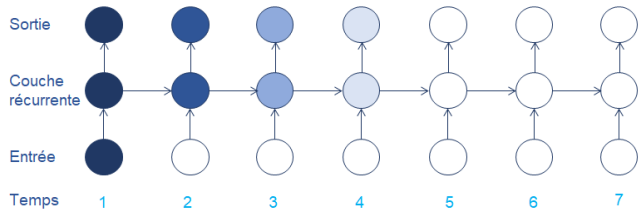
$$\frac{\partial C}{\partial V_i} = \sum_{t=1}^{t_f} \frac{\partial C}{\partial g_i}(t) * \frac{\partial g_i}{\partial V i}(t) = \sum_{t=1}^{t_f} \frac{\partial C}{\partial g_i}(t) * (h_i(t-1))^T \tag{3.26}$$

Le point fort des RNNs est qu'ils sont capables d'exploiter l'information contextuelle afin de passer de l'entrée à la sortie. Cependant, l'apprentissage se révèle difficile et le contexte temporel exploité est local. L'origine des problèmes de ce réseau est que la sortie dépend des états récurrents, concrètement via la multiplication par la matrice V_i. Le cumul des multiplications et des fonctions d'activation engendre une augmentation exponentielle de l'influence sur la sortie et la rétro-propagation du gradient, ce qui est appelé dans la littérature "vanishing gradient problem".

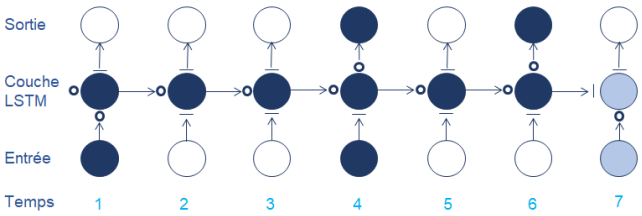
Depuis l'année 1990, plusieurs méthodes ont essayé de contrer ces problèmes. Cependant, l'approche qui a prouvé son efficacité et qui est devenue la plus connue pour le traitement des séquences est le LSTM proposé par [HS97]. En fait, LSTM permet de garder l'information à un long intervalle, de faciliter l'accès à ces informations pertinentes et de réduire le problème de gradient évanescent grâce à ses portes logiques multiplicatives. La figure 3.10 illustre la différence entre le RNN et LSTM lors de la conservation de l'information.

En outre, nous avons présenté dans la figure 3.10a une couche RNN où la sensibilité à l'information d'entrée ne dépasse pas un pas de temps. Ce phénomène est représenté de sorte que plus le neurone est sensible à l'information d'entrée plus il est sombre. La sensibilité décroît exponentiellement et l'entrée de nouvelles informations efface la mémoire de la première entrée. Cependant, la figure 3.10b présente une couche de LSTM capable de préserver le gradient sur plusieurs pas de temps. La couche LSTM est représentée avec les trois portes (d'entrée, d'oubli et de sortie) ce qui est plus nombreux que pour la couche de RNN. Les portes sont représentées par "*o*" si elles sont ouvertes et par "-" si elles sont fermées. L'état interne dans LSTM se montre capable de garder l'information le plus longtemps possible par rapport à une couche RNN tant que la porte d'oubli reste ouverte et celle d'entrée reste fermée. De plus, pour la couche de sortie, la sensibilité peut varier, sans influencer l'état interne, grâce à la porte d'oubli.

C'est à ce niveau-là et après cette comparaison entre une couche de RNN et de LSTM, que nous présentons notre contribution qui ne se manifeste pas dans l'utilisation du



(a) Le gradient diminue dans un RNN et la sensibilité à l'information d'entrée décroît avec le temps



(b) Préservation du gradient dans un LSTM et conservation de l'information sur un nombre important d'impacts de pas de temps

FIGURE 3.10 – La différence de conservation d'information entre une couche de RNN et LSTM

LSTM mais plutôt dans l'exploitation de l'information contextuelle entre les détections qui nous permettent de minimiser et faciliter l'association des détections pour construire les tracklets. En revanche, cette nouvelle méthode d'association nous a permis d'améliorer les performances, par ailleurs les coûts du passage d'une architecture séquentielle à une architecture à base de LSTM n'augmentent pas le temps de calcul qui reste négligeable. Par la suite, nous détaillons le modèle d'un LSTM.

Description d'un modèle LSTM

Le modèle "Long short-term memory" à mémoire long/court-terme persistante est nommé LSTM. En fait, une couche L"$_i$ de LSTM contient quatre couches de RNN dont trois servent à contrôler et agissent comme des portes logiques qui délivrent 1 ou 0. La première sert à contrôler la quantité d'information qui entre à la couche i ; la deuxième contrôle les informations cachées par les états internes d'un pas à un autre ; et la troisième gère les informations sortantes de la couche i. Cependant, la dernière couche du RNN alimente l'état interne via la porte d'entrée de la cellule LSTM. Un flux d'informations dans une couche L"$_i$ est décrit dans la figure 3.11.

Pour plus de précision, la figure 3.12 illustre un fonctionnement détaillé d'une couche i du LSTM. En effet, pour un vecteur d'entrée $x(t)$ et $t \in [1, t_f]$ qui passe par l'ensemble de la couche L"$_i$ avec $i \in [1, N]$ à chaque pas du temps, $z(t)$ sera le vecteur de sortie avec $t \in [1, t_f]$. La sortie de la couche i est calculée comme suit :

— L'état des portes d'entrée w_j et b_j est déterminé par l'application de la transformation affine a_j avec j, la couche précédente. Le vecteur V_j est ajouté par une transformation linéaire à $t-1$ pas du temps au vecteur qui va sortir de la couche i. De plus, un vecteur d'état interne est ajouté au pas du temps précédent à la couche i via u_j. Les équations 3.27 et 3.28 représentent la liaison entre ces paramètres et l'influence de la couche précédente j sur la couche courante i.

$$g_j(t) = W_j * h_{i-1}(t) + V_j * h_i(t-1) + u_j \odot s(t-1) + b_j \tag{3.27}$$

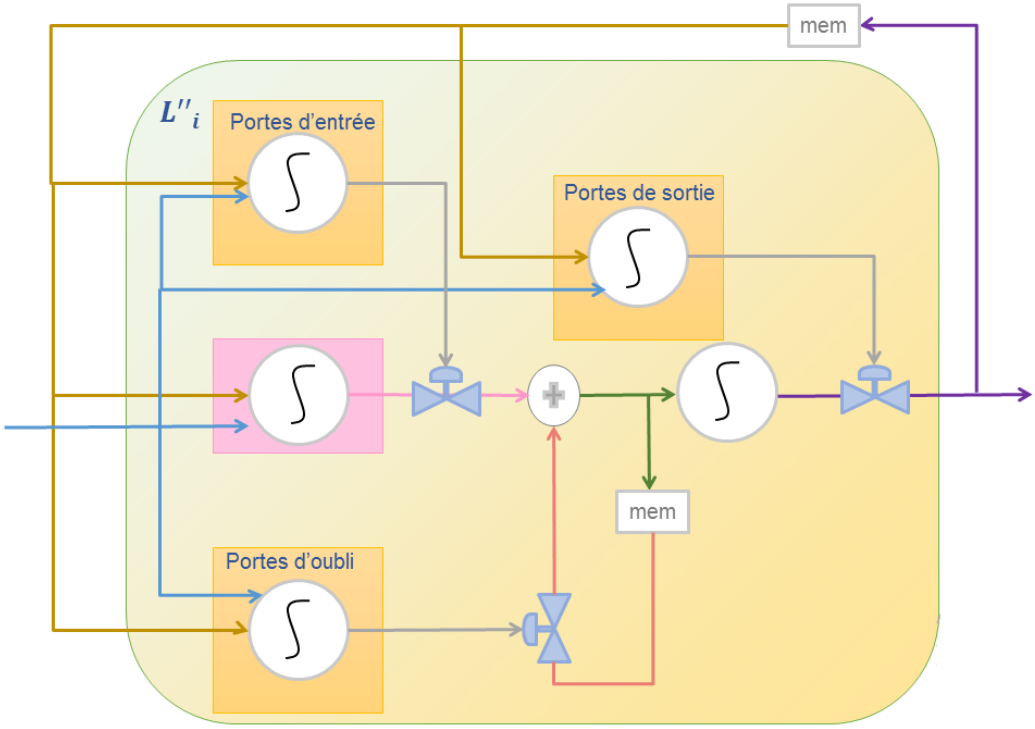


FIGURE 3.11 – L'interaction entre les portes et les vannes qui contrôlent la propagation en passe-avant de l'information provenant de la mémoire interne, de l'entrée et de la sortie par le LSTM

$$j(t) = \sigma_j(g_j(t)) \tag{3.28}$$

Avec $\odot$: multiplication terme à terme afin que les portes d'entrée ne peuvent avoir accès qu'à l'état interne qu'elles peuvent influencer.

— La même procédure est suivie pour l'état des portes d'oubli :

$$g_f(t) = W_f * h_{i-1}(t) + V_f * h_i(t-1) + u_f \odot s(t-1) + b_f \tag{3.29}$$

$$f(t) = \sigma_f(g_f(t)) \tag{3.30}$$

— La pondération du nouveau vecteur par la valeur de la porte d'entrée ainsi que celle du vecteur interne par la valeur de la porte d'oubli au pas précédent, détermine la nouvelle valeur de l'état interne de la couche i.

$$g_s(t) = W_s * h_{i-1}(t) + V_s * h_i(t-1) + b_s \tag{3.31}$$

$$s(t) = f(t) \odot s(t-1) + j(t) \odot \sigma_s(g_s(t)) \tag{3.32}$$

— L'état de la porte de sortie est déterminé à partir de l'état d'entrée de la même façon que celle de la porte d'entrée.

$$g_o(t) = W_o * h_{i-1}(t) + V_o * h_i(t-1) + u_o \odot s(t) + b_o \tag{3.33}$$

$$o(t) = \sigma_o(g_o(t)) \tag{3.34}$$

— La sortie de cette couche i est déterminée par la pondération de l'état interne par les valeurs de la porte de sortie au pas de temps.

$$h_i(t) = o(t) \odot \sigma_h(s(t)) \tag{3.35}$$

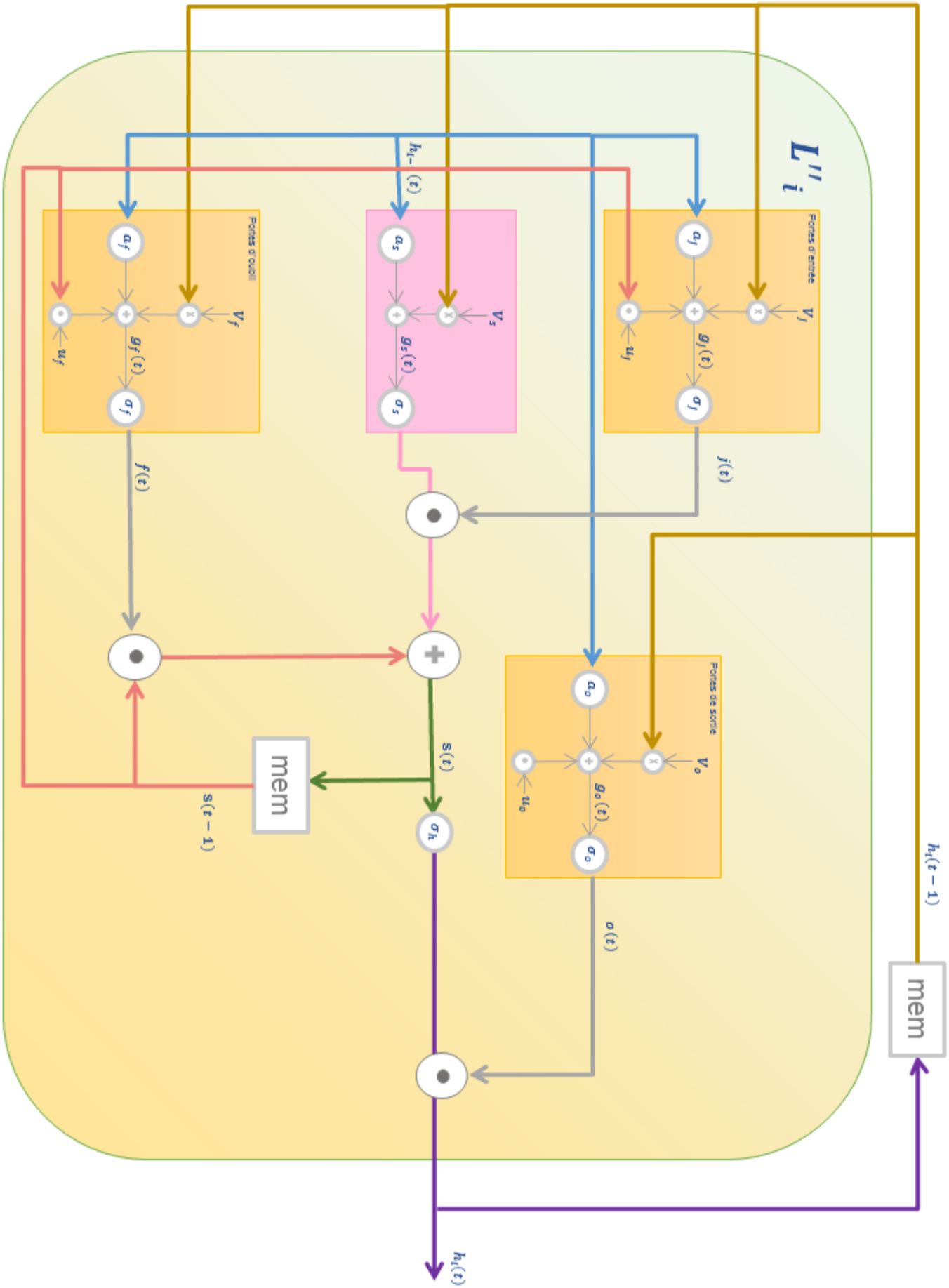


FIGURE 3.12 – Visualisation de la propagation du gradient lors d'un passe-avant dans une couche LSTM

A ce stade-là, les vecteurs d'activation sont définis comme suit :
$i(t)$ de la porte d'entrée de la couche i gère la qualité de l'information entrante.
$f(t)$ de la porte d'oubli gère la quantité de l'information qui a été mémorisée.
$s(t)$ gère l'état interne.
$o(t)$ de la porte de sortie gère l'information sortante.

La rétro-propagation d'un modèle LSTM

La rétro-propagation du LSTM ne diffère pas trop de celle d'un RNN standard. A chaque pas de temps, un calcul de gradient est exécuté et la somme, également à chaque pas de temps, donne un gradient global. La figure 3.13 illustre la rétro-propagation du gradient dans une couche i d'une cellule LSTM dans un pas de temps t.

Afin de calculer à chaque pas de temps la dérivée partielle de C par rapport aux différents paramètres de la couche de LSTM, contrairement au passe-avant dans la rétro-propagation,les calculs commencent à partir de la porte de sortie (équation 3.37).

$$\frac{\partial C}{\partial o}(t) = (\frac{\partial C}{\partial h_i}(t) + \epsilon(t+1)) \odot \sigma_h(s(t)) \tag{3.36}$$

$$\epsilon(t) = \epsilon_o(t) + \epsilon_s(t) + \epsilon_f(t) + \epsilon_j(t) \tag{3.37}$$

Avec $\epsilon(t+1)$ issu du lien récurrent de la couche i et $\epsilon(t_f+1) = 0$.

La rétro-propagation par la matrice jacobienne $J_{\sigma_o}(g_o(t))$ liée à la fonction σ_o d'activation au point $g_o(t)$ est illustrée dans l'équation 3.38.

$$\frac{\partial C}{\partial g_o}(t) = J_{\sigma_o}(g_o(t)) * \frac{\partial C}{\partial o}(t) \tag{3.38}$$

Ensuite, les contributions de la porte de sortie sont définies comme suit :

$$\delta_o(t) = W_o^T * \frac{\partial C}{\partial g_o}(t) \tag{3.39}$$

$$\epsilon_o(t) = V_o^T * \frac{\partial C}{\partial g_o}(t) \tag{3.40}$$

$$\gamma_o(t) = u_o \odot \frac{\partial C}{\partial g_o}(t) \tag{3.41}$$

Et le gradient par rapport à l'état interne est :

$$\frac{\partial C}{\partial s}(t) = J_{\sigma_h}(s(t)) * ((\frac{\partial C}{\partial h_i}(t) + \epsilon(t+1)) \odot o(t)) + \gamma(t) \tag{3.42}$$

$$\gamma(t) = \gamma_o(t) + \gamma_s(t+1) + \gamma_f(t+1) + \gamma_j(t+1) \tag{3.43}$$

Puis, le gradient pondéré du vecteur interne est calculé par l'équation suivante :

$$\frac{\partial C}{\partial g_s}(t) = J_{\sigma_s}(g_s(t)) * (\frac{\partial C}{\partial s}(t) \odot j(t)) \tag{3.44}$$

$$\gamma_s(t) = \frac{\partial C}{\partial s}(t) \odot f(t) \tag{3.45}$$

De même, le gradient de la porte d'entrée et d'oubli est :

$$\frac{\partial C}{\partial j}(t) = \frac{\partial C}{\partial s}(t) \odot \sigma_s(g_s(t)) \tag{3.46}$$

$$\frac{\partial \mathrm{C}}{\partial f}(t) = \frac{\partial \mathrm{C}}{\partial s}(t) \odot s(t-1) \tag{3.47}$$

Et le gradient d'entrée est calculé par les équations 3.48 et 3.49 :

$$\frac{\partial \mathrm{C}}{\partial g_f}(t) = \mathrm{J}_{\sigma_f}(g_f(t)) * \frac{\partial \mathrm{C}}{\partial f}(t) \tag{3.48}$$

$$\frac{\partial \mathrm{C}}{\partial g_j}(t) = \mathrm{J}_{\sigma_j}(g_j(t)) * \frac{\partial \mathrm{C}}{\partial j}(t) \tag{3.49}$$

Les contributions de la porte d'oubli sont définies par :

$$\delta_f(t) = \mathrm{W}_f^{\mathrm{T}} * \frac{\partial \mathrm{C}}{\partial g_f}(t) \tag{3.50}$$

$$\epsilon_f(t) = \mathrm{V}_f^{\mathrm{T}} * \frac{\partial \mathrm{C}}{\partial g_f}(t) \tag{3.51}$$

$$\gamma_f(t) = u_f \odot \frac{\partial \mathrm{C}}{\partial g_f}(t) \tag{3.52}$$

Egalement, les contributions de la porte d'entrée sont calculées comme suit :

$$\delta_j(t) = \mathrm{W}_j^{\mathrm{T}} * \frac{\partial \mathrm{C}}{\partial g_j}(t) \tag{3.53}$$

$$\epsilon_j(t) = \mathrm{V}_j^{\mathrm{T}} * \frac{\partial \mathrm{C}}{\partial g_j}(t) \tag{3.54}$$

$$\gamma_j(t) = u_j \odot \frac{\partial \mathrm{C}}{\partial g_j}(t) \tag{3.55}$$

Et les contributions de l'état interne sont :

$$\delta_s(t) = \mathrm{W}_s^{\mathrm{T}} * \frac{\partial \mathrm{C}}{\partial g_s}(t) \tag{3.56}$$

$$\epsilon_s(t) = \mathrm{V}_s^{\mathrm{T}} * \frac{\partial \mathrm{C}}{\partial g_s}(t) \tag{3.57}$$

A la fin de la rétro-propagation, le coût C est calculé à partir de la couche i lorsque la séquence est parcourue et il est calculé par rapport au paramètre x avec $x = i, f, s, ouo$:

$$\frac{\partial \mathrm{C}}{\partial \mathrm{W}_x}(t) = \sum_{t=1}^{t_f} \frac{\partial \mathrm{C}}{\partial g_x}(t) * h_{i-1}(t)^{\mathrm{T}} \tag{3.58}$$

$$\frac{\partial \mathrm{C}}{\partial \mathrm{V}_x}(t) = \sum_{t=1}^{t_f} \frac{\partial \mathrm{C}}{\partial g_x}(t) * h_{i-1}(t)^{\mathrm{T}} \tag{3.59}$$

$$\frac{\partial \mathrm{C}}{\partial b_x}(t) = \sum_{t=1}^{t_f} \frac{\partial \mathrm{C}}{\partial g_x}(t) \tag{3.60}$$

Il est aussi calculé pour la porte d'entrée, de sortie et d'oubli :

$$\frac{\partial \mathrm{C}}{\partial u_j} = \sum_{t=1}^{t_f} \frac{\partial \mathrm{C}}{\partial g_j}(t) \odot s(t-1)^{\mathrm{T}} \tag{3.61}$$

$$\frac{\partial \mathrm{C}}{\partial u_o} = \sum_{t=1}^{t_f} \frac{\partial \mathrm{C}}{\partial g_o}(t) \odot s(t)^{\mathrm{T}} \tag{3.62}$$

$$\frac{\partial \mathrm{C}}{\partial u_f} = \sum_{t=1}^{t_f} \frac{\partial \mathrm{C}}{\partial g_f}(t) \odot s(t-1)^{\mathrm{T}} \tag{3.63}$$

Architecture de suivi

Nous transformons notre ancienne architecture à des blocs de RNN et de LSTM.

Comme nous l'avons prouvé, les tracklets prennent en considération les états précédents. Ce phénomène est cohérent avec les caractéristiques des RNNs. En fait, le partage de l'information entre les couches cachées dans le module d'association permet la sélection des informations les plus pertinentes dans le but de faciliter l'association entre les différentes détections. Puis, la prédiction à l'instant $t-1$ alimente l'association à l'instant t et permet la mise à jour au même moment t. La figure 3.14 montre la liaison entre les différents blocs (prédiction, association et mise à jour). L'association nécessite un LSTM ; alors que la prédiction et l'étape de mise à jour peuvent avoir lieu avec de simples RNN puisque ces deux étapes ne nécessitent pas de longue mémoire pour leurs tâches.

Le module de l'association est émis par un LSTM caractérisé par la longue mémoire qui permet d'apprendre les associations dans de longues séquences contrairement aux architectures classiques qui considèrent que toutes les hypothèses sont valides, ce qui augmente la complexité. En fait, à ce niveau-là, nous définissons l'appartenance des détections à des trajectoires. Pour cela, nous définissons l'architecture LSTM utilisée et qui est capable d'apprendre les liaisons entre les objets détectés. Le modèle illustré dans la figure 3.15 montre l'architecture de notre LSTM. L'idée est d'exploiter l'information temporelle extraite à chaque pas du LSTM pour prédire l'affectation de chaque détection. L'entrée de cette unité à l'étape i avec l'état caché h_i et l'état de la cellule c_i se présente comme un vecteur de caractéristiques. Dans notre cas, nous utilisons la matrice de distance $\mathrm{C}_{ij} = \| x^i - z^j \|_2$ qui calcule la distance euclidienne entre l'état prédit de i et celui mesuré de j. Le vecteur utilisé dans notre cas est le même vecteur de caractéristiques d'apparence. La sortie de cette unité est le vecteur de probabilité A_i pour une détection ; et toutes les mesures sont valables. Cette probabilité est calculée grâce à une couche de Softmax avec une normalisation de la valeur prédite. Cette couche n'apparaît pas dans le LSTM classique montré et détaillé dans la sous-section précédente.

Pour le calcul d'erreurs de cette unité, nous utilisons l'équation 3.64 :

$$\mathscr{L}(\mathrm{A}^i, \tilde{\mathrm{a}}) = -log(\mathrm{A}_{i\tilde{\mathrm{a}}}) \tag{3.64}$$

Avec A_{ij} la probabilité d'affectation de i à j et ã une affectation correcte.

Nous avons détaillé l'unité LSTM qui présente le noyau de notre architecture et ainsi nous détaillons les deux autres modules de prédiction et de mise à jour. A l'instant t, le RNN produit quatre valeurs ainsi que l'état caché h_t pour l'étape suivante : la première valeur $x^*_{t+1} \in \mathbb{R}^{\mathrm{N.D}}$ est issue de la prédiction pour toutes les cibles. La deuxième valeur $x_{t+1} \in \mathbb{R}^{\mathrm{N.D}}$ est issue de l'étape de mise à jour. La troisième valeur est celle de la probabilité $\epsilon \in (0,1)^{\mathrm{N}}$ qui indique s'il s'agit d'une vraie trajectoire ou pas. Et la dernière valeur ϵ^*_{t+1} est calculée à partir de l'état actuel x_t, sa probabilité d'existence ϵ_t, la mesure z_{t+1} ainsi que la probabilité d'association A_{t+1} ; ce qui permet la prédiction à partir d'un apprentissage dynamique du modèle afin de prédire une cible dans l'absence de mesure. De même, la mise à jour permet d'apprendre la correction de la distribution des états afin d'affecter les cibles aux mesures correspondantes.

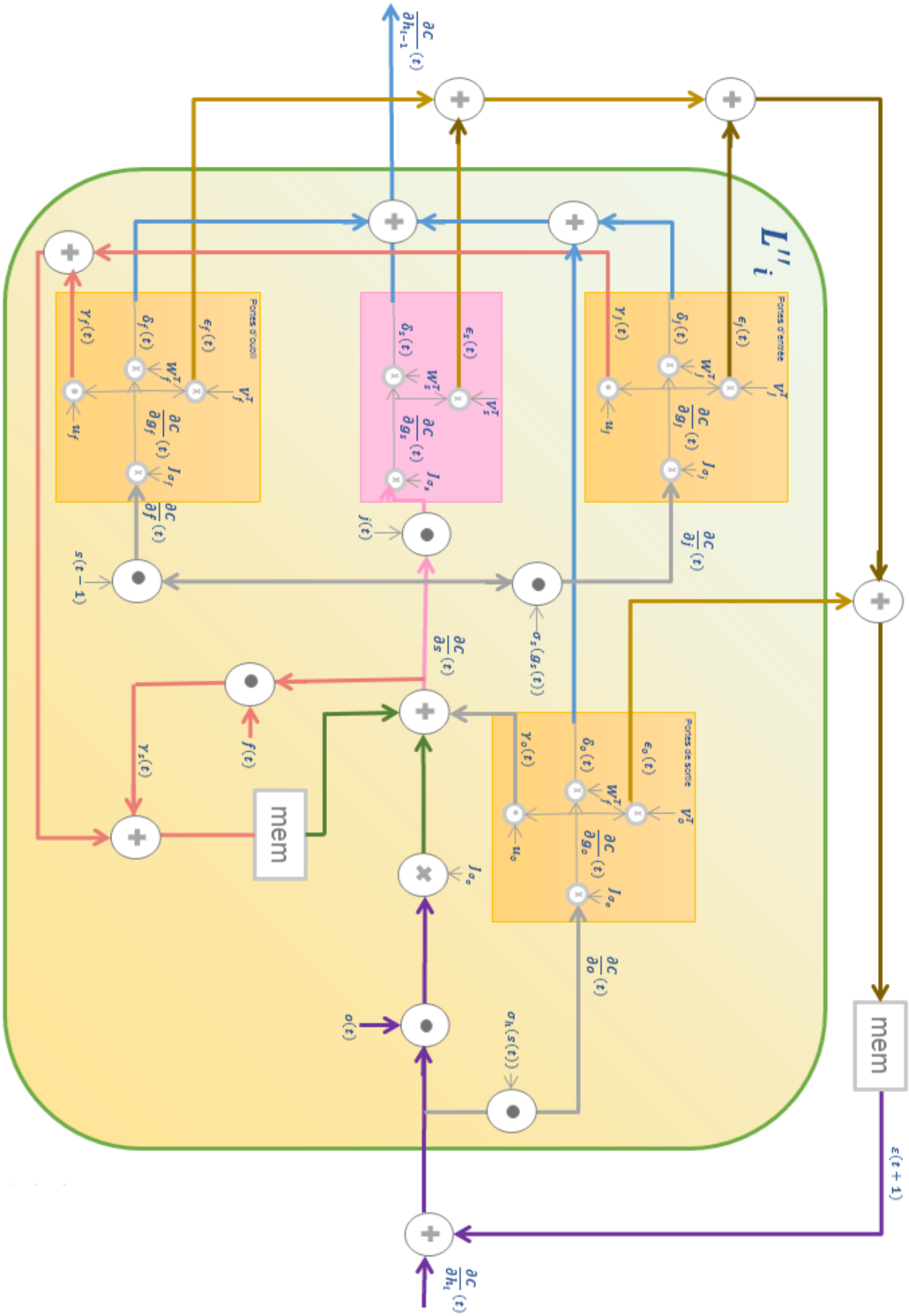


FIGURE 3.13 – Visualisation de la rétro-propagation du gradient lors du passe-arrière dans une couche LSTM

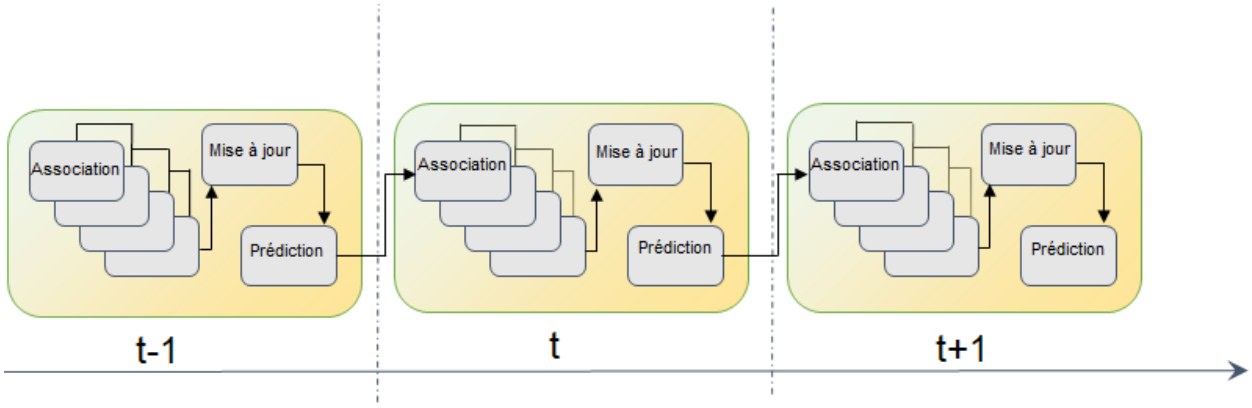


FIGURE 3.14 – Architecture de suivi mono-caméra

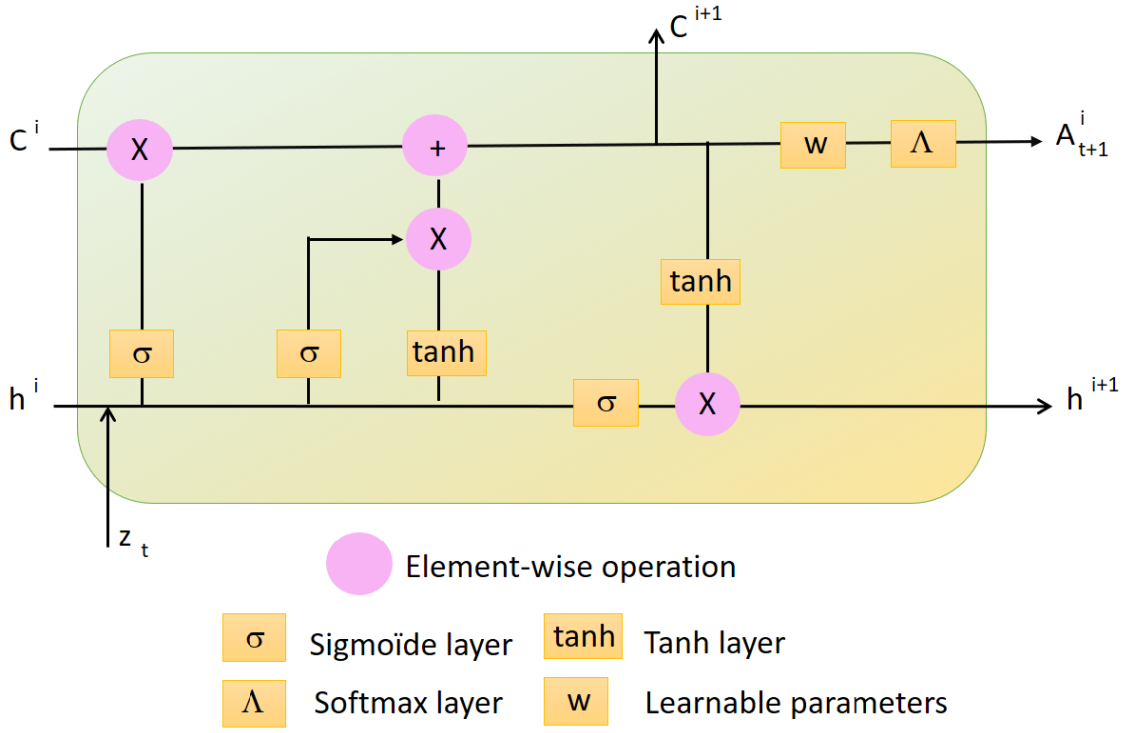


FIGURE 3.15 – Une unité de LSTM

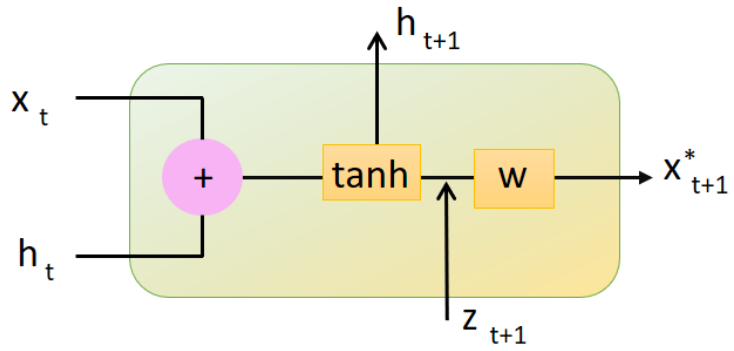


FIGURE 3.16 – Le module de prédiction

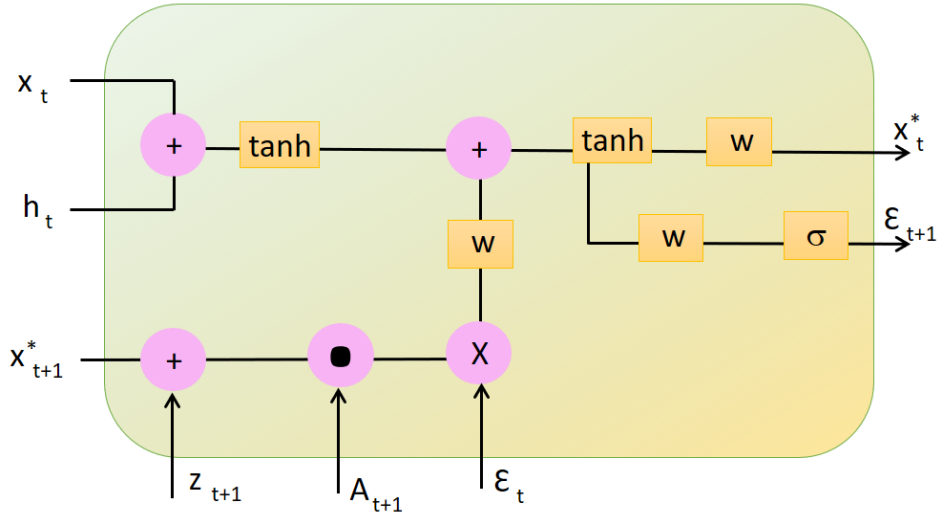


FIGURE 3.17 – Le module de mise à jour

Les deux blocs de prédiction et de mise à jour sont illustrés dans les figures 3.16 et 3.17. La prédiction x^*_{t+1} pour la frame suivante dépend de l'état actuel x_t et l'état caché h_t. Une fois que l'association A_{t+1} est faite pour cette frame, l'état est mis à jour selon la probabilité. A la fin, toutes les mesures et les états prédits sont concaténés pour former $\widehat{x} = [z_{t+1}; x^*_{t+1}]$ pondéré par la probabilité d'affiliation A_{t+1}.

Pour le calcul d'erreurs, comme pour tout objectif d'apprentissage automatique, le but est de voir la qualité du modèle. Nous nous intéressons à une perte en corrélation avec les performances de suivi, laquelle est illustrée par l'équation 3.65 suivante :

$$\mathscr{L}(x^*, x) = \frac{\lambda}{ND} \sum \| x^* - x \|^2 + \frac{K}{ND} \| x - x \|^2 \tag{3.65}$$

Avec x^* la valeur prédite et avec x la vraie valeur, nous voulons que l'architecture apprenne à prédire les trajectoires les plus proches de la vérité terrain. Cela est valable pour la prédiction et la mise à jour même s'il y a un manque de mesure. A la fin, nous minimisons l'erreur quadratique moyenne entre les prédictions, la mise à jour et la vérité de terrain.

TABLEAU 3.2 – Les bases de données utilisées dans le suivi et leurs caractéristiques

Bases de données	Caractéristiques
PETS 2009 S2L1	Occultation, croisement et changement d'ID
PETS 2009 S2L2	nombre important de piétons
ETHMS (Sunny and Bahnhof)	Caméra mobile
"LATIS-IP"	Conditions différentes (jour et nuit)

3.5 Expérimentation

Dans cette section, nous présentons l'évaluation de nos deux architectures de suivi dans une seule caméra. Nous validons nos résultats sur des bases publiques en utilisant des métriques de suivi. Les contributions au niveau du suivi dans une seule caméra se manifestent par l'élimination de faux positifs et par la supposition des mesures des objets qui devaient être détectés mais qui ne l'ont pas été. Pour se faire, la partie expérimentale et les résultats seront présentés de façon à mettre en valeur ces contributions.

D'abord, nous détaillons les conditions et les différents paramètres d'implémentation : les deux architectures sont implémentées dans un PC avec 3.5 GHz CPU et 2 Go GPU. Le temps d'exécution dépend, en effet, de la scène traitée. Il est égal à 0.45 (sec/frame) pour la séquence PETS 2009 S2L2, égal à 0.35 pour la base ETHMS et égal à 0.2 pour la base PETS 2009 S2L1. Le calcul de l'apparence est la partie qui prend le plus de temps et qui est équivalent à peu près à 40% du temps de calcul total. La phase de prédiction et de mise à jour sont faites par une seule couche de RNN et de 300 couches cachées. Un LSTM, qui permet l'association, nécessite plus d'énergie et de pouvoir pour l'apprentissage. Il est fait de 2 couches et de 500 couches cachées. Nous utilisons 200 000 itérations qui sont suffisantes pour la convergence. L'apprentissage prend approximativement une journée et se fait sous la bibliothèque Torch 7 en utilisant lua.

3.5.1 Base de données

Dans le suivi, les bases de données sont assez nombreuses (publiques ou privées). Nous utilisons une sélection de ces bases afin de montrer la robustesse de notre architecture. Chaque base de données utilisée ainsi que ses caractéristiques sont mentionnées dans le tableau 3.2.

PETS 2009 S2L1 et PETS 2009 S2L2 [FS09]

La base PETS 2009 contient 7 séquences filmées à l'extérieur avec une base faisant partie de plusieurs challenges de suivi. Nous nous intéressons aux deux séquences S2L1 et S2L2. La première séquence de 795 frames présente plusieurs cas d'occultation. La figure 3.18 montre quelques frames de cette séquence.

La seconde séquence est encombrée et contient 436 frames. Elle contient 74 piétons qui se déplacent de façon aléatoire et non-linéaire. La figure 3.19 présente quelques frames de cette séquence.

ETHMS (Sunny and Bahnhof) [ELVG07]

Cette base est obtenue à partir d'une caméra mobile, ce qui rend le suivi plus difficile. Elle est filmée à l'extérieur avec une caméra stéréo et contient 1000 frames. Elle a toujours

FIGURE 3.18 – Exemple de frame de la séquence S2L1

FIGURE 3.19 – Exemple de frame de la séquence S2L2

été le sujet de plusieurs défis car elle présente une variété de problématiques tels que la similarité entre les piétons, le croisement et l'occultation.

LATIS-IP

L'acquisition de cette base est faite dans la plateforme française PAVIN. En effet, cette base est conçue afin de valider des résultats de ré-identification comme il sera détaillé dans le chapitre suivant. Cependant, nous pouvons exploiter cette base ainsi que chaque séquence séparément pour faire le suivi. L'acquisition est faite par des caméras statiques installées dans la plateforme. Les caméras utilisées sont de type "FUJINON FISH EYE 1 :1.4/1.8mm" de résolution 640 X 480. Cette base contient 12000 images et est compo-

FIGURE 3.20 – Exemple de frame de la base ETHMS

FIGURE 3.21 – Exemple de frame de la base LATIS-IP

sée de deux sous-bases : la première renferme des séquences filmées pendant le jour ; la deuxième des séquences dans des conditions météorologiques différentes.

3.5.2 Métriques

Les bases de données ne sont pas les seules utilisées pour valider nos architectures. Avec la même importance, nous choisissons les métriques qui déterminent la robustesse d'une architecture de suivi. Les métriques utilisées sont de "CLEAR MOT" : MOTP, MOTA, Precision, recall et FP.

- **MOTP metric :** La métrique, qui correspond à la précision du suivi multi-objets, est comparée aux résultats de la vérité terrain.
- **MOTA metric :** La précision du suivi multi-objets prend en considération le changement d'ID, le nombre de détections manquantes et le nombre de faux positifs. C'est une des métriques les plus utilisées pour évaluer la qualité du suivi.
- **Recall :** Les objets correctement appariés sont comparés à des objets de la vérité terrain.
- **Precision :** Les objets correctement appariés sont comparés à la totalité des objets.
- **FP :** Le nombre de faux positifs.
- **FN :** Le nombre de faux négatifs.

3.5.3 Résultats et discussions

Dans cette partie, nous validons notre approche de suivi et nous mettons en valeur nos contributions. Les résultats seront présentés de la manière suivante : une étude sera faite afin d'analyser l'impact d'utilisation des tracklets sur le nombre de faux positifs. Puis, nous exposons nos résultats de suivi sur des plateformes fixes et mobiles à travers des bases de données publiques tout en les comparant à d'autres travaux de l'état de l'art. L'approche est aussi validée avec d'autres exemples d'objets tels que les véhicules.

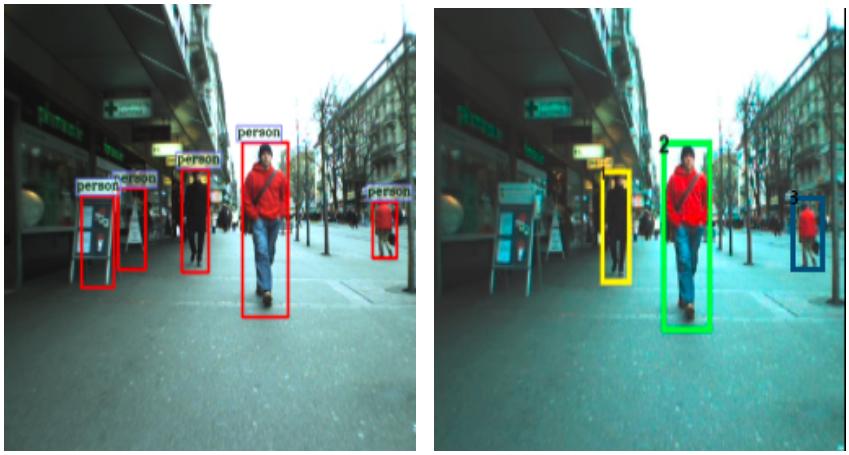
Impact des tracklets sur le nombre de faux positifs

Théoriquement, nous avons démontré que les tracklets éliminent les faux positifs lors de l'association, à supposer que la taille d'une tracklet est inférieure au nombre de N que nous avons choisi pour définir le nombre minimum pour construire une tracklet. Expérimentalement, nous prouvons l'effet des tracklets sur cette problématique par le calcul

TABLEAU 3.3 – Les taux de faux positifs avant et après le suivi

	ETHMS	PETS2009 S2L1	PETS2009 S2L2
FP du détecteur	0.053	0.081	0.004
FP après le suivi	0.01	0.03	0.002

des faux positifs après la phase de détection faite par Faster R-CNN après le suivi. Un exemple de résultat de détection sur la première frame de la base ETHMS est illustré dans la figure 3.22 avec une comparaison du résultat de suivi. Le détecteur arrive à détecter les trois piétons dans la scène mais indique aussi deux objets de l'arrière-plan comme étant également des personnes; ce qui n'est plus le cas après la méthode de suivi.



(a) Résultat de détection (b) Résultat du suivi

FIGURE 3.22 – Elimination des faux positifs issus de la détection par la méthode de suivi

Pour prouver cette hypothèse, nous appliquons notre approche de suivi sur les trois bases publiques (ETHMS, PETS2009 S2L1 et PETS2009 S2L2) et nous calculons le taux de faux positifs comme il est illustré dans le tableau 3.3. En effet, notre approche a pu limiter quelques défauts issus du détecteur pour les trois bases.

Validation de notre approche de suivi sur une plateforme statique

Les deux séquences PETS2009 S2L1 et PETS2009 S2L2 sont filmées par des caméras statiques. Le choix de ces séquences est basé sur le fait que les deux ont fait l'objet de plusieurs challenges de suivi et que chacune d'elle présente un cas particulier à traiter.

PETS2009 S2L1 : La séquence PETS2009 S2L1 filmée à l'extérieur ne présente pas seulement des piétons mais aussi plusieurs cas d'occultation et de croisement entre les objets. Le tableau 3.4 prouve l'efficacité de notre approche de suivi face à d'autres travaux de l'état de l'art.

D'après ce tableau, nos approches ont les meilleurs résultats. Nos meilleures valeurs de MOTA, MOTP sont respectivement 0.86 et 0.72, par rapport à (0.83, 0.69) de [BY14] et (0.79, 056) de [BRL+11].

Dans [BY14], le meilleur des deux travaux est une méthode basée sur des tracklets. Cependant, notre méthode présente plus de performances grâce au module de vérification et de mise à jour vérifiant les trajectoires.

Ensuite, nous atteignons 0.93 de précision avec notre approche basée sur le RNN, ce qui signifie que le réseau récurrent apprend l'association des détections avec beaucoup

TABLEAU 3.4 – Comparaison de nos approches appliquées à la base PETS2009 S2L1 par les métriques MOTA, MOTP, Precision, Recall et FP

Base de données	Méthodes	MOTA	MOTP	Precision	Recall	FP
PETS S2L1	Breitenstein et al. 2011[BRL+11]	0.79	0.56	-	-	-
	Vo et al. 2013[VV13]	-	-	0.81	0.82	0.16
	Bae et al. 2014[BY14]	0.83	0.69	-	-	0.19
	Yoon et al. 2015 [YYLY15]	-	-	0.85	0.8	0.25
	Kim et al. 2017 [Kim17]	-	-	0.89	0.90	-
	Notre approche sans RNN	**0.86**	0.69	**0.93**	0.82	0.05
	Notre approche à base de RNN	0.84	**0.72**	0.9	**0.87**	**0.03**

TABLEAU 3.5 – Comparaison de nos approches appliquées à la base PETS2009 S2L2 par les métriques MOTA, MOTP, Precision, Recall et FP

Base de données	Méthodes	MOTA	MOTP	Precision	Recall	FP
PETS S2L2	Poiesi et al .2013 [PMC13]	0.59	-	-	-	-
	Bae et al. 2014 [BY14]	0.7	0.53	-	-	0.14
	Milan et al. 2015 [MLTSR15]	0.46	0.67	-	-	-
	Zhang et al. 2015 [ZWW+15]	-	-	0.92	0.62	-
	Hong et al.2016 [HYLYY16]	0.44	0.69	-	-	-
	Yu et al. 2017 [YCZ+17]	-	-	0.87	0.69	-
	Notre approche sans RNN	**0.68**	0.54	**0.99**	0.6	**0.002**
	Notre approche à base de RNN	0.66	**0.69**	0.89	**0.65**	0.01

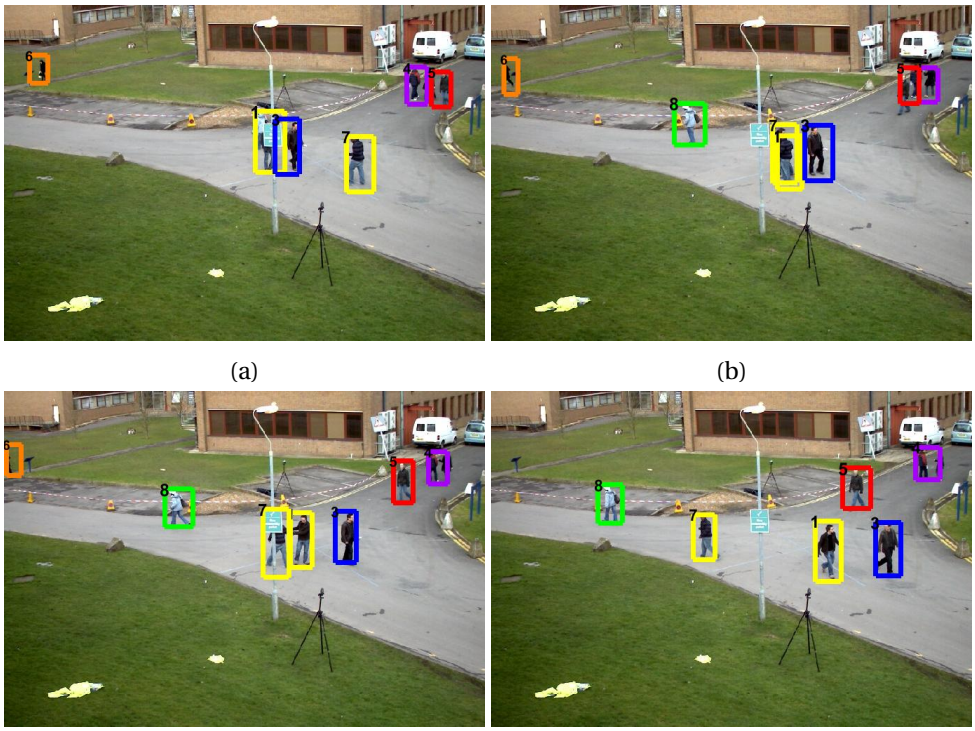
de précision et nous enregistrons aussi une valeur de "Recall" égale à 0.87. Ces deux valeurs dépassent les valeurs de l'état de l'art alors que les travaux de Vo et al. [VV13] ont 0.81 de précision et 0.82 de Recall, [YYLY15] Yoon et al. présentent 0.85 et 0.8 ; et Kim et al. [Kim17] ont 0.89 et 0.9.

Nous exposons aussi quelques frames de la séquence afin de visualiser les résultats de suivi dans la figure 3.23. Nous pouvons visualiser un suivi simple sans circonstance particulière dans la frame 156. Puis, dans la frame 168, on trouve un croisement entre 3 piétons dont 2 marchant dans la même direction et un dans la direction opposée. Spécialement, le piéton 7, qui croise le piéton 3, subira ensuite dans la frame 173 une occultation avec un objet de l'arrière-plan. Et nous pouvons vérifier la robustesse de notre approche dans la frame 185 où tous les piétons ont pu garder leur IDs.

PETS2009 S2L2 : Cette séquence, filmée aussi à l'extérieur, présente un nombre important de piétons marchant ensemble avec beaucoup de similarité dans leur apparence ainsi que des croisements entre eux.

Le tableau 3.5 présente les résultats des métriques appliquées sur cette séquence en comparaison avec d'autres travaux de l'état de l'art. Nous enregistrons 0.68, 0.69, 0.99, 0.65, 0.01 de MOTA, MOTP, Precision, Recall et FP respectivement par rapport à 0.7 de valeur de MOTA de [BY14] qui est une valeur largement supérieure à la nôtre. Ceci est dû, selon nos analyses, au fait que dans les travaux de [BY14], ils classifient les tracklets selon leur confiance et ce particulièrement dans une scène comme celle-ci qui peut donner un meilleur résultat ; et qui, en contrepartie, ralentit un peu plus le temps de calcul par une étape de plus. Puis, nous présentons la meilleure précision par rapport à 0.92 de la part de [ZWW+15]. Enfin, dans une base contenant une foule, nous présentons 0.002 de faux positifs grâce à l'apprentissage avec le RNN.

Un exemple de frame indiquant le suivi par notre méthode est illustré dans la figure



(a) (b)

(c) (d)

FIGURE 3.23 – Les résultats de suivi sur la base PETS2009 S2L1 : (a)Frame 156 : le suivi est fait sans circonstances particulières; (b)Frame 168 : Croisement entre les piétons 7,3 et 1; (c)Frame 173 : Occultation entre le piéton 7 et un objet de l'arrière-plan; (d)Frame 185 : les piétons ont pu garder leur ids malgré l'occultation et le croisement

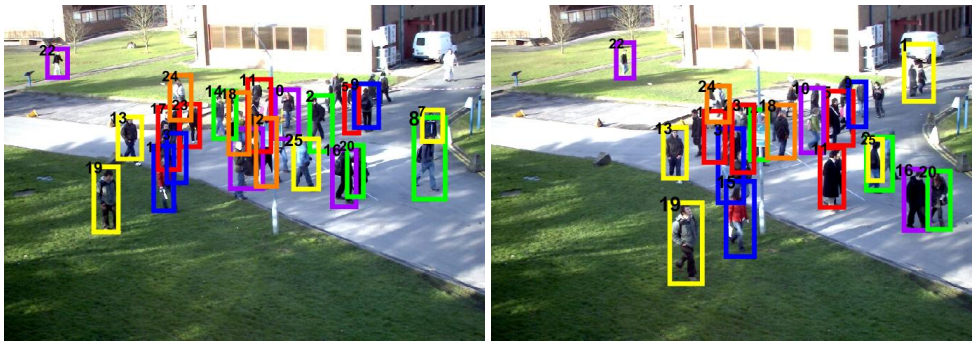
3.24. Même si nous n'avons pas les trajectoires de tous les piétons de la scène, les piétons suivis gardent leur ID même après le croisement avec d'autres et ils sont suivis le plus longtemps possible.

Validation de notre approche de suivi sur une plateforme mobile

La base de données ETHMS (Sunny and Bahnhof) a été choisie afin de valider notre approche sur une plateforme mobile. Étant une base publique utilisée par plusieurs chercheurs pour valider leurs approches, elle est issue d'une caméra mobile dans des rues bondées.

D'après le tableau 3.6, nous pouvons confirmer la robustesse de notre architecture face à des problématiques plus complexes que celles rencontrées dans les plateformes statiques. Nous avons pu enregistrer jusqu'à 0.86 et 0.72 de MOTA et MOTP respectivement. La précision est de valeur 0.93 et le nombre de faux positifs ne dépasse pas 0.03 par rapport à 0.25 de [YYLY15] qui utilise un réseau bayésien. Notre précision est meilleure que celle de [Kim17] qui est égale à 0.89.

Afin de visualiser l'effet de notre approche avec une caméra mobile, nous présentons quelques frames dans la figure 3.25. En fait, cette figure illustre la robustesse de l'approche de suivi face au changement de pose ainsi que la taille des piétons notamment dans la figure (h) où l'on voit des piétons proches de la caméra, raison pour laquelle leur taille est importante et des piétons se situant loin de la caméra et donc de taille réduite ainsi que des piétons se trouvant en face ou de côté par rapport à celle-ci de même que d'autres tournant le dos à la caméra. Notre approche a pu les suivre tous quelle que soit leur pose. La figure montre aussi que l'approche fait face à des occultations et révèle la disparition



(a) (b)

FIGURE 3.24 – Les résultats de suivi sur la base PETS2009 S2L2 : (a)Frame 67 ; (b)Frame 95

TABLEAU 3.6 – Comparaison de nos approches appliquées à la base ETHMS par les métriques MOTA, MOTP, Precision, Recall et FP

Base de données	Méthodes	MOTA	MOTP	Precision	Recall	FP
ETHMS	Vo et al. 2013[VV13]	-	-	0.76	0.71	0.88
	Poiesi et al. 2013[PMC13]	-	-	0.85	0.78	-
	Bae and al. 2014[BY14]	0.72	0.64	-	-	0.04
	Yoon et al. 2015[YYLY15]	-	-	0.86	0.81	-
	Yu and al. 2017[YCZ+17]	-	-	0.9	0.79	-
	Kim et al. 2017 [Kim17]	-	-	0.82	0.73	0.78
	Notre approche sans RNN	**0.74**	0.66	**0.99**	0.6	**0.002**
	Notre approche à base de RNN	**0.69**	0.68	0.89	**0.65**	0.01

d'objets : la figure (c) montre un piéton qui se cache derrière un arbre et dans la figure (b), on peut apercevoir un piéton qui disparaît complètement derrière un autre objet. Enfin, les objets sont suivis le plus longtemps possible : par exemple le piéton 3 qui apparaît dès la première frame, disparaît après 216 frames (cf. figure f).

Le suivi dans des cas particuliers

Notre approche n'est pas influencée par les changements météorologiques, ce qui est prouvé par notre base (cf. figure 3.26) où le suivi est effectué pendant la nuit et le jour.

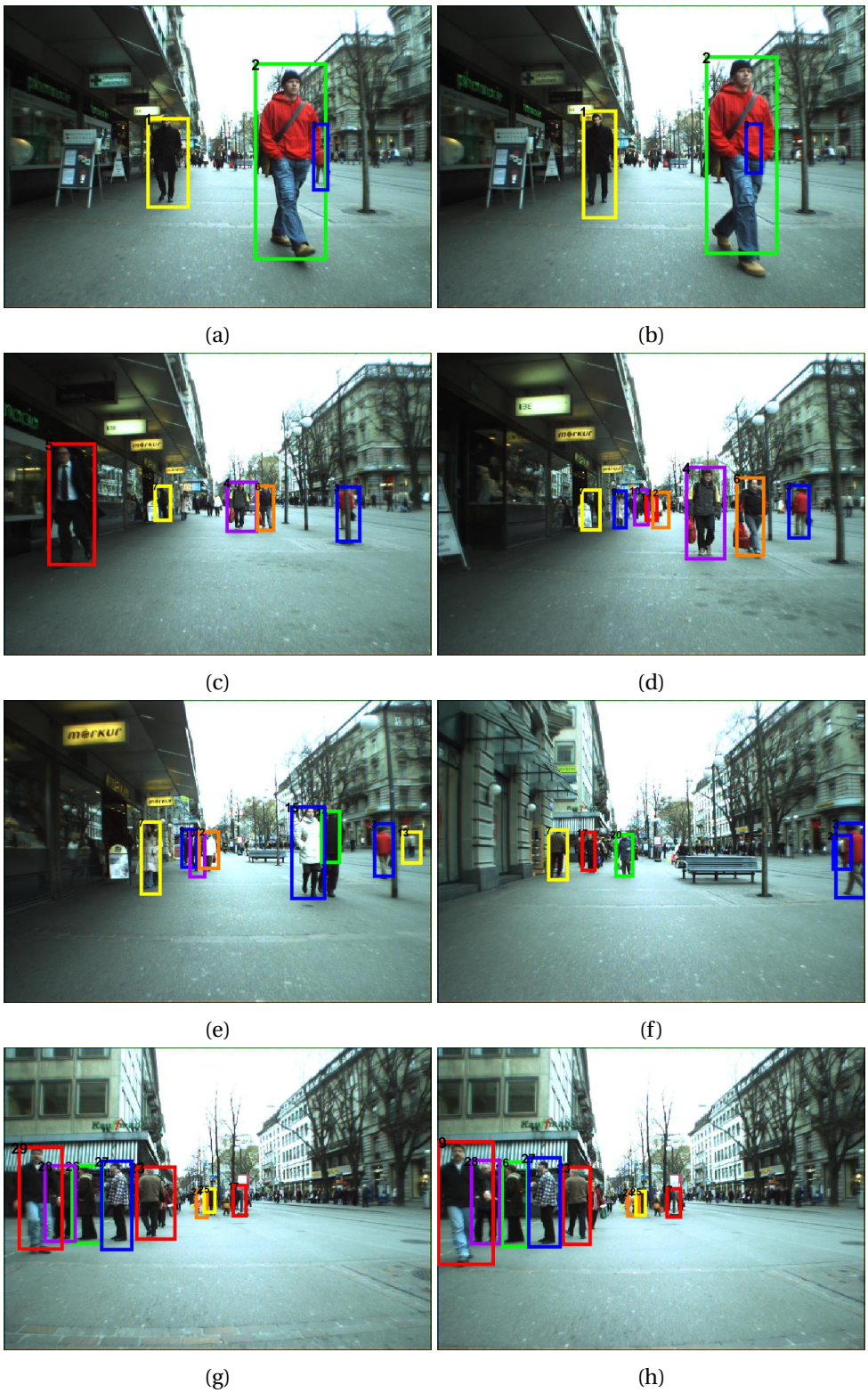
Des objets autres que les piétons ont été utilisés lors de la phase de test afin de valider la notion de multi-objets. L'évaluation est faite sur une base privée "CIF" et les résultats sont présentés dans la figure 3.27 où tous les véhicules sont suivis dans les deux voies dès leur apparition et jusqu'à leur disparition.

Notre approche apprend les caractéristiques de l'objet indépendamment de la nature de l'objet lui-même, ce qui est mis en valeur par la figure 3.28 qui présente plusieurs objets dans la même scène (motos, voitures, piétons et tuks tuks).

3.5.4 Analyse des erreurs

Dans cette partie du chapitre, nous critiquons notre approche de suivi dans une seule caméra afin de la développer pour effectuer un suivi dans un réseau de caméras.

Avec le développement des détecteurs, qui présentent encore des défauts au niveau de la détection, et grâce à la notion des tracklets, nous arrivons à limiter le taux de faux positifs. Or, si un faux positif est associé à une tracklet, il sera de même associé dans la



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 3.25 – Notre approche de suivi est validée par la base ETHMS (Sunny and Bahnhof) : (a)Frame 17 : La détection et le suivi des 3 premiers piétons apparaissant dans la vidéo. (b)Frame 19 : Suivi réussi malgré l'occultation entre l'objet 3 en bleu et 2 en vert. (c)Frame 70 : La poursuite du piéton 3 en présence de l'arbre. (d)Frame 90 : Ré-identification du piéton 3 après la disparition derrière l'arbre. (e)Frame 121 : Différentes tailles de piétons poursuivis avec succès. (f)Frame 216 : La dernière apparition du piéton 3 qui s'est montré pour la première fois dans la frame 1. (g)Frame 312 : Réussir à suivre des piétons apparaissant pour la première fois. (h)Frame 315 : Suivi de différents piétons avec différents angles de vue

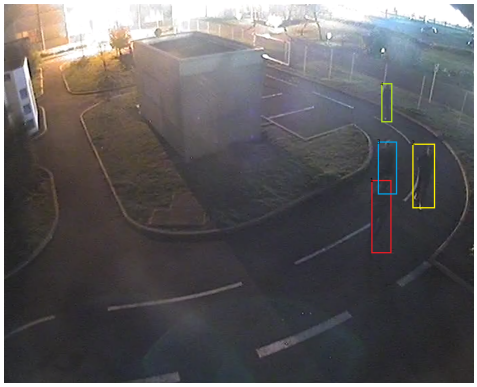


FIGURE 3.26 – Le suivi dans des conditions météorologiques dégradées

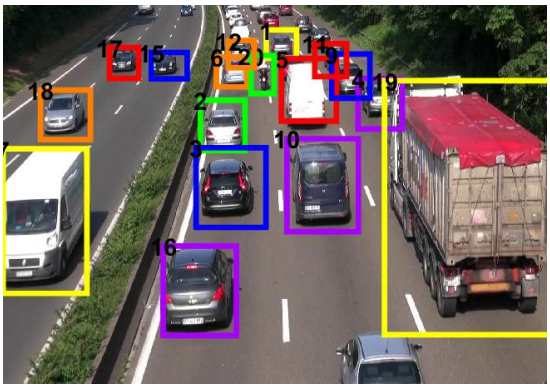


FIGURE 3.27 – Une frame qui illustre le suivi des véhicules dans la base CIF

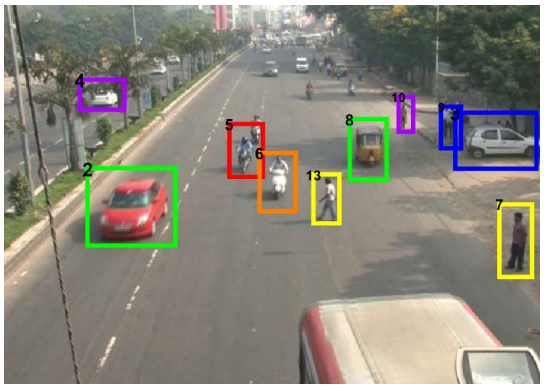


FIGURE 3.28 – Le suivi des différents objets de la base "INDIA"

trajectoire et propagé dans le réseau de caméras à une plus grande échelle. Cependant, ce qui nous encourage à négliger ce défaut c'est le taux faible et négligeable de faux positifs calculé sur plusieurs bases.

L'extraction des caractéristiques par l'histogramme présente plusieurs lacunes, ce qui provoquera plus de problèmes notamment si nous passons d'une caméra à une autre ou d'un environnement à un autre où la luminosité est différente.

3.6 Conclusion

Dans ce chapitre, nous avons présenté notre méthode de suivi qui passe d'une chaîne séquentielle à une architecture basée sur un réseau récurrent. Notre méthode de suivi a été conçue pour des scénarios présentant des piétons ou des véhicules grâce à son module de signature qui apprend les caractéristiques indépendamment de l'objet. Elle est robuste à des scénarios complexes. Même si des foules ou un nombre important de piétons ou de véhicules sont présents, et même dans le cas particulier où la séance est filmée la nuit, cette méthode montre de très bonnes performances.

Afin de mettre en valeur les atouts de notre méthode, nous l'avons présentée de la façon suivante : une vue globale de la méthode de suivi est présentée en premier ; puis nous avons détaillé la méthode de détection en rappelant l'architecture du détecteur utilisé : ensuite, nous avons développé la définition de la signature à travers l'extraction des caractéristiques. Après la définition des signatures, nous avons présenté notre architecture de suivi où nous avons associé les détections ayant des signatures proches afin de définir les tracklets. L'association des tracklets a été présentée dans une phase que nous avons appelée "globale" et que nous avons présentée, dans une sous-section : la partie de mise à jour où nous avons exposé la correction des trajectoires.

De même, nous avons présenté la minimisation de l'architecture en utilisant les réseaux récurrents tout en mettant en valeur leurs atouts quant à notre architecture.

Dans une dernière partie, nous avons évalué nos approches avec des bases de données publiques afin de montrer que nos performances sont robustes et meilleures comparées à d'autres travaux de l'état de l'art.

Chapitre 4

La ré-identification dans un réseau de caméras

« *Les meilleures choses ont besoin de patience* »

Jean Anglade

Sommaire

4.1 Introduction 72
4.2 La ré-identification des piétons dans un réseau de caméras 72
4.3 Transfert de tracklets entre n caméras 73
4.3.1 Principe du CycleGAN 73
4.3.2 Transfert des tracklets 75
4.4 Vue globale de l'architecture de ré-identification dans un réseau de caméras 77
4.4.1 Extraction des caractéristiques 78
4.4.2 Extraction de l'information temporelle par LSTM 79
4.4.3 Comparaison et décision 80
4.5 Approche de ré-identification dans des conditions particulières 81
4.6 Expérimentation 84
4.6.1 Base de données 85
4.6.2 Métrique 88
4.6.3 Résultats et discussions 88
4.7 Conclusion 95

4.1 Introduction

Dans ce chapitre, l'approche du suivi dans une seule caméra, développée dans le chapitre précédent, a été étendue afin de répondre aux challenges de la ré-identification. En fait, le suivi dans un réseau de caméra fait appel à plusieurs défis par rapport au suivi limité à une seule caméra : plus précisément, la différence d'environnement, d'arrière-plan, de luminosité ou le passage d'une caméra à une autre, appelée discontinuité spatio-temporelle et visuelle. Un autre facteur peut aussi influencer les performances de la ré-identification par la problématique d'extraction des caractéristiques pour décrire l'objet en général et les piétons de façon particulière. Une description précise des piétons et une multiplication de données avec une architecture profonde sont une stratégie efficace permettant de limiter toutes ces difficultés.

Dans ce chapitre, une solution de ré-identification est présentée en étant basée sur le transfert des tracklets d'une caméra à une autre afin d'augmenter les données nécessaires à un apprentissage plus efficace et d'éviter les problèmes de surdimensionnement "over-fitting". De plus, l'extraction des caractéristiques est effectuée selon une méthode de description par parties afin d'avoir des informations plus précises sur le piéton. Ces caractéristiques offriront par la suite une entrée à un réseau récurrent afin d'extraire de l'information temporelle, la classification se faisant par un réseau de calcul de similarité.

Dans la deuxième section de ce chapitre, soit après l'introduction, la ré-identification dans un réseau de caméras est mise en valeur par une vision globale de l'architecture. La troisième section est consacrée au détail du pré-traitement et à la mise en valeur de notre première contribution du transfert de tracklets. Nous rappelons les différentes étapes de ré-identification après la phase de transfert dans une section de vue globale qui détaillera ensuite l'extraction des caractéristiques et la phase de décision. Dans la section cinq, nous mettons en valeur notre contribution au niveau de l'application et ce en adaptant notre approche sur des conditions météorologiques dégradées. Enfin, la section six présente la partie expérimentale permettant de valider l'approche présentée. La dernière section sera la conclusion.

4.2 La ré-identification des piétons dans un réseau de caméras

La ré-identification des piétons dans le réseau de caméras se fait par notre algorithme basé sur une architecture de LSTM. Afin d'optimiser notre phase d'apprentissage et de faciliter l'attribution des IDs, une phase de transfert de tracklets d'une caméra à une autre est utilisée. La figure 4.1 illustre l'approche de façon générale et dont l'apprentissage de la méthode dépend des tracklets déjà définies dans le chapitre précédent. Tout d'abord, l'acquisition se fait au niveau des n caméras présentes dans le réseau. Une phase de post-traitement aura lieu au moyen d'un auto-encodeur pour reconstruire des tracklets d'une caméra à une autre. Puis, dans l'architecture de ré-identification, une phase d'extraction des caractéristiques aura lieu par un réseau CNN en vue d'extraire les informations liées à l'apparence. En se basant sur un réseau récurrent, une extraction des informations temporelles est établie. Une fois toutes les informations spatio-temporelles extraites, la dernière couche de classification du réseau est remplacée par un réseau de calcul de similarité afin de calculer la similarité entre les tracklets issues de chaque caméra.

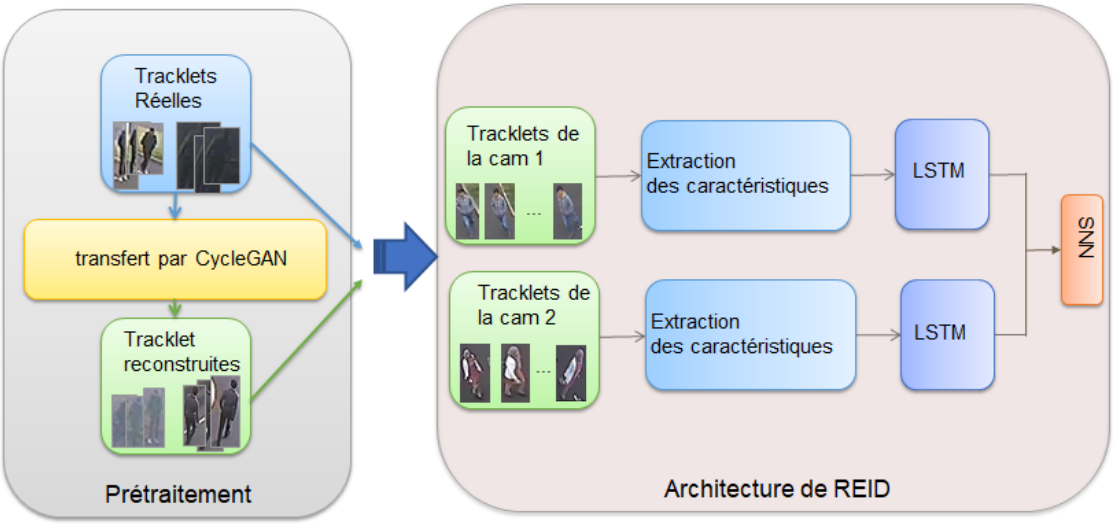


FIGURE 4.1 – Notre architecture de ré-identification

4.3 Transfert de tracklets entre n caméras

Dans une étape de prétraitement, où parfois les données ne sont pas assez suffisantes pour avoir un apprentissage profond parfait, un transfert de données peut être une solution. Dans cette section, une approche de transfert de données est détaillée.

4.3.1 Principe du CycleGAN

Le "Generative Adversarial Network" (GAN) est un réseau capable d'apprendre la correspondance "the mapping" d'une image du domaine X à une autre image du domaine Y. Il apprend à translater automatiquement l'image d'un domaine à un autre.

Le CycleGAN est un cas particulier du réseau GAN qui permet d'apprendre à translater simultanément deux représentations de deux images différentes X et Y tels que : $X \rightarrow Y$ et $Y \rightarrow X$.

Par ailleurs, ce réseau permet d'extraire les caractéristiques spéciales des images et de comprendre comment ces caractéristiques pourraient être traduites en d'autres. Pendant des années, les chercheurs ont réussi à avoir des modèles robustes pour la translation d'un domaine à un autre en présence de paires pour l'entraînement. Cependant, la tâche de la translation d'un domaine à un autre, en absence de toutes données exemples pour l'entraînement, reste une tâche complexe. Bien que l'entraînement par des auto-encodeurs soit non supervisé, la supervision au niveau de l'ensemble des images dans le domaine X et dans le domaine Y peut être exploitable en donnant des images du domaine X et des images différentes du domaine Y et par la suite un apprentissage de correspondance "mapping" aura lieu entre les deux.

En effet, l'entraînement se fait à partir de la fonction de correspondance entre les deux domaines $G : X \rightarrow Y$ avec une sortie $\hat{y} = G(x)$, $x \in X$ et cette sortie est indifférenciable de l'image $y \in Y$. La fonction G translate la représentation du domaine X en domaine $\hat{Y}$ similaire à Y avec une fonction D_y pour vérifier s'il s'agit d'une bonne reconstruction ou non selon le domaine Y. Il y a une infinité de fonctions de correspondances G qui amènera à la même sortie $\hat{y}$. La figure 4.2 illustre un modèle simple de CycleGan avec un exemple d'images. L'objectif est de translater d'un domaine à un autre. Quant à la translation inverse, il s'agira de reconstruire l'objet initial. Par exemple, lors de la translation

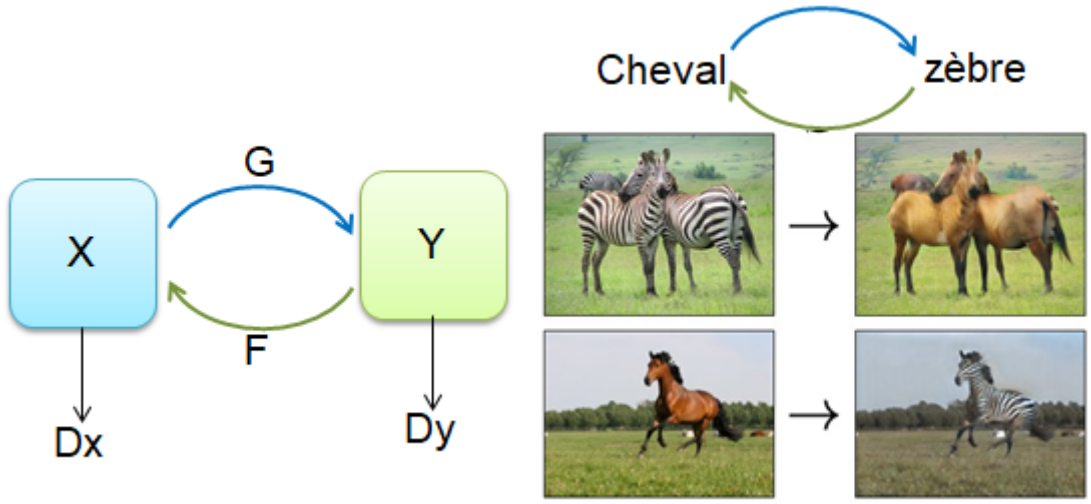


FIGURE 4.2 – La translation d'une image de domaine X à une autre image de domaine Y et l'inverse

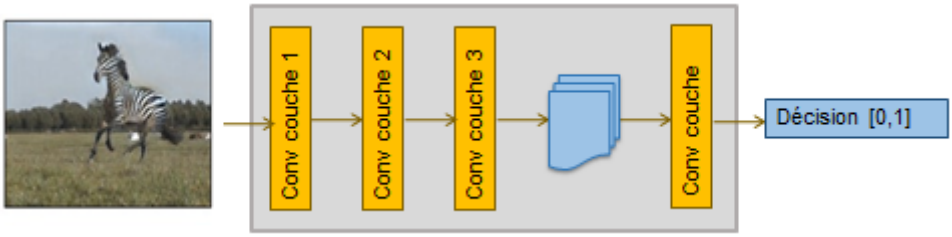


FIGURE 4.3 – L'architecture du discriminateur qui prend la décision après la reconstruction

d'une phrase du français à l'anglais, le but est de retrouver la même phrase d'origine en français dans le cas où la traduction est faite à partir de la phrase en anglais.

Le CycleGAN est composé de deux modèles :

Modèle discriminateur : Ce modèle est responsable de la vérification des données générées qui peuvent être vraies ou fausses. Le discriminateur prend la décision pour classer les données générées.C'est un réseau de convolution avec une couche de décision (cf. figure 4.3).

Modèle générateur : Ce modèle est responsable de la génération de nouvelles données similaires aux données d'entrée. Le générateur, comme tout autre système, peut produire des défaillances.

En mathématique, soit les deux fonctions $G : X \rightarrow Y$ et $F : Y \rightarrow X$ l'une est l'inverse de l'autre. La fonction F translate Y avec une fonction D_x pour vérifier la reconstruction selon le domaine X.

En outre, il y aura un apprentissage simultané des deux fonctions en ajoutant une fonction de minimisation "cycle consistency loss" pour le calcul d'erreurs et qui permet d'avoir $F(G(x)) \approx x$ et $G(F(y)) \approx y$. Deux fonctions "adversarial discriminators" D_A et D_B sont utilisées pour vérifier si les images sont bien translatées ; d'où la fonction "loss function" dans l'équation 4.1 et qui additionne les fonctions d'erreurs $V_{GAN}(D_B, G, A, B)$ et $V_{GAN}(D_A, F, B, A)$ pour les fonctions de correspondance (G et F) et les fonctions discriminateur D_A et D_B. Le $V_{cyc}(G, F)$ est la fonction de "cycle consistency loss" qui fait que $F(G(x)) \approx x$ et $G(F(y)) \approx y$ dans chaque image et qui peut être reconstruite après chaque cycle de correspondance :

$$V(G, F, D_A, D_B) = V_{GAN}(D_B, G, A, B) + V_{GAN}(D_A, F, B, A) + \lambda V_{cyc}(G, F) \tag{4.1}$$

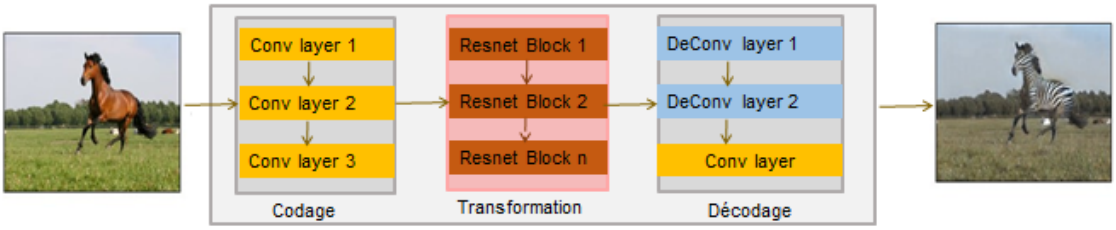


FIGURE 4.4 – L'architecture d'un générateur

La figure 4.4 illustre l'architecture du générateur construite de trois étapes (codage, transformation et décodage).

Étape de codage : L'entrée du codage est une image de taille [256,256,3] ; et la première étape est l'extraction des caractéristiques par les couches de convolution. La sortie de la couche de convolution est un tenseur de dimension [256,256,64] qui entre par la suite dans d'autres couches de convolution. Chaque couche de convolution conduit à une extraction progressive des caractéristiques les plus élevées ; ce qui conduit à la compression de l'image à 256 vecteurs de caractéristiques de taille $64 * 64$. A ce stade, la transformation aura lieu pour transformer un vecteur de caractéristiques d'un domaine X à un autre Y.

Étape de transformation : Dans la phase de transformation, une combinaison entre les caractéristiques des deux domaines aura lieu. La sortie de la phase de transformation est un vecteur de [64,64,256]. Le réseau est composé de six blocs de resnet. En effet, le resnet est un réseau de neurone composé de deux couches de convolution dont l'entrée est connectée à la sortie. Ceci permet de s'assurer que les propriétés de la couche d'entrée sont également disponibles dans la couche de sortie, de sorte que la sortie ne diverge pas de l'entrée et que les caractéristiques de l'image d'entrée seront conservées.

Étape de décodage : L'entrée du décodage est le vecteur de sortie du transformateur de taille [64,64,256]. La phase de décodage est l'inverse de la première phase de codage ; et ce dans le but de reconstruire les caractéristiques à partir du vecteur issu de l'étape de transformation. Ce réseau est construit par des couches de déconvolution (la transposée de convolution). Enfin, le vecteur final est converti en image de sortie dans le domaine Y avec une taille [256,256,3].

4.3.2 Transfert des tracklets

Le CycleGan permet de reconstruire des échantillons provenant d'un domaine à un autre dans notre travail. Ce réseau a été exploité afin de reconstruire des tracklets d'une caméra à une autre. En fait, le choix de reconstruction des tracklets, et non pas des échantillons, est fondé sur les points forts de tracklets. Grâce aux tracklets (l'ensemble des détections successives sur un nombre de frames défini), les faux positifs détectés dans une caméra ne sont plus envoyés ou reconstruits dans les autres caméras, ce qui garantit moins de bruit et facilite la ré-identification.

D'ailleurs, la fonction G devient une fonction de mapping entre une tracklet T_x de la caméra x et une autre T_y de la caméra y tel que $G : T_x \rightarrow T_y$. Cette fonction aura une tracklet de sortie $\widehat{T_y}$ dans la caméra y qui est similaire à T_y (la tracklet issue de la caméra y). Puisque l'idée est de construire des tracklets d'une caméra à une autre, la fonction inverse ($F : T_y \rightarrow T_x$) reconstruira les tracklets originales. L'apprentissage se fait avec la fonction de perte "cycle consistency loss" qui minimise l'erreur afin d'avoir $F(G(T_x) \approx T_x$ et $G(F(T_y) \approx T_y$. La figure 4.5 illustre le transfert d'une tracklet de la caméra numéro 1 qui a filmé une séquence vidéo pendant le jour vers une caméra qui a filmé une séquence

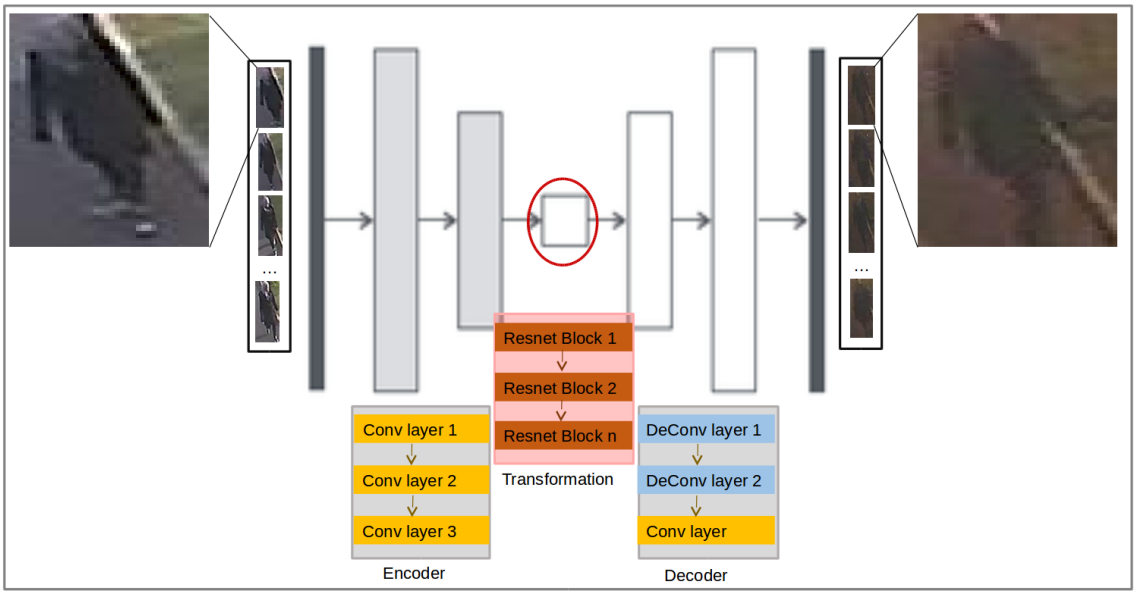


FIGURE 4.5 – La reconstruction des tracklets d'une caméra à une autre

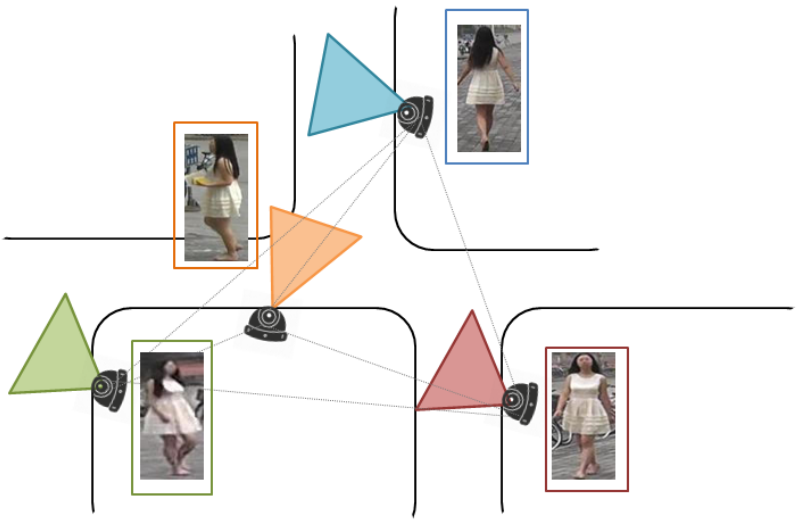


FIGURE 4.6 – La ré-identification dans un réseau de caméras

vidéo pendant la nuit.

Ainsi, le transfert des tracklets est utilisé dans cette thèse dans deux scénarios différents : lorsque la ré-identification a lieu dans un nombre de caméras important ou lorsque les piétons sont situés dans deux moments différents (jour/nuit) :

La ré-identification dans plusieurs caméras :

Dans un réseau de caméras, où le nombre de caméras est supérieur à deux, la ré-identification devient de plus en plus complexe surtout avec le changement de pose et d'angle de vue de la caméra. Un exemple de répartition des caméras et des images est illustré dans la figure 4.6.

La ré-identification dans deux environnements différents :

La ré-identification peut avoir lieu dans un même endroit à différents moments de la journée, par exemple : le jour, la nuit, par temps ensoleillé, ombragé ou sombre, pluvieux ou brumeux, ... La figure 4.7 illustre un exemple d'une même caméra qui a filmé deux séquences d'images dans deux conditions météorologiques différentes (jour/nuit).

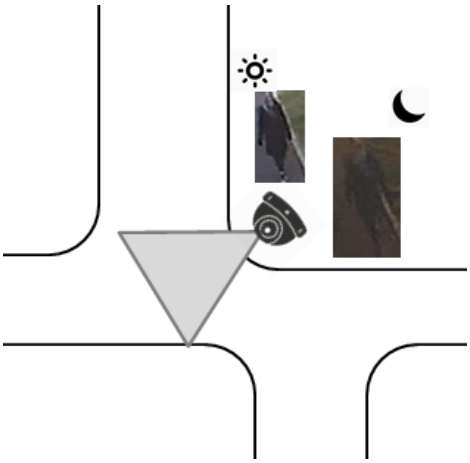


FIGURE 4.7 – La ré-identification dans deux moments différents

4.4 Vue globale de l'architecture de ré-identification dans un réseau de caméras

Après la phase de pré-traitement, où les tracklets sont reconstruites d'une caméra à une autre afin d'avoir un volume suffisant d'échantillons, une architecture à base d'un LSTM est faite afin d'accomplir la phase de ré-identification. L'entrée de l'architecture est démontrée dans la figure 4.8.

Dans la ré-identification, la première phase consiste à extraire des caractéristiques par un réseau de convolution. Dans cette phase, un tenseur de caractéristiques de chaque échantillon est réalisé. Puis, chaque échantillon passe par un LSTM pour extraire l'information temporelle. A ce stade, une comparaison d'échantillon-trajectoire aura lieu après l'extraction de l'information temporelle par LSTM. Chaque tracklet de chaque caméra sera comparée aux tracklets issues d'autres caméras du réseau. Un des points forts de cette thèse est que la ré-identification ne se fait pas échantillon par échantillon mais plutôt tracklet par tracklet (cf. figure 4.9).

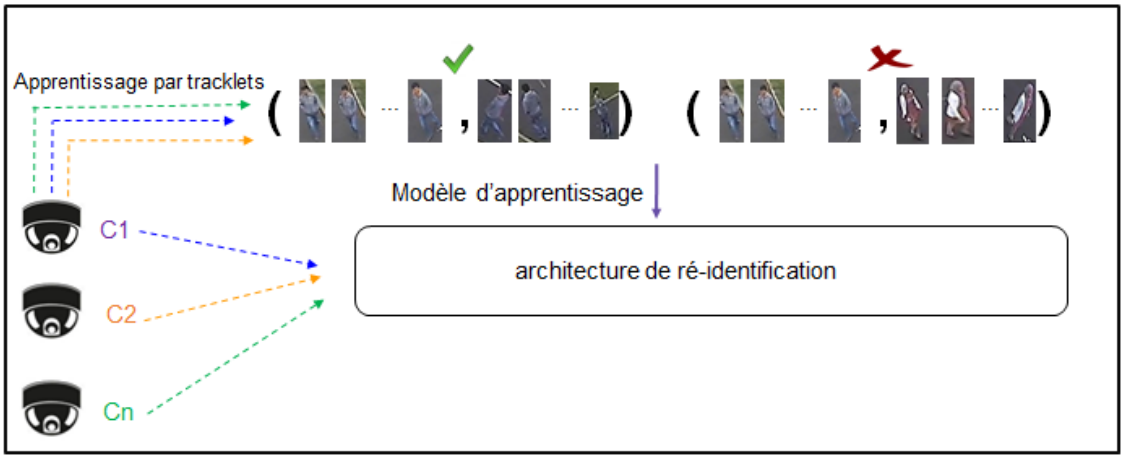


FIGURE 4.8 – Les tracklets sont l'entrée principale de l'architecture de la ré-identification

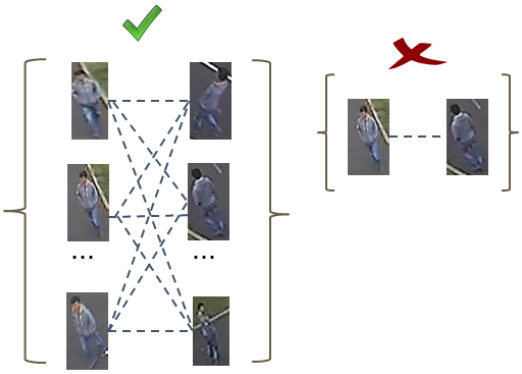


FIGURE 4.9 – La comparaison se fait entre les tracklets issues de chaque caméra

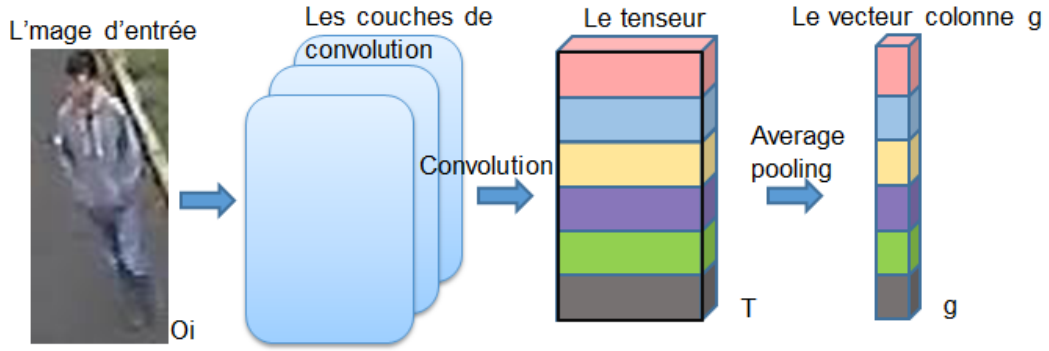


FIGURE 4.10 – La reconstruction du vecteur de caractéristiques

4.4.1 Extraction des caractéristiques

Dans la phase d'extraction des caractéristiques, un réseau de convolution est utilisé, inspiré par les travaux de [SZY+17]. L'entrée du réseau est une image et la sortie est un tenseur 3D. Le réseau de convolution utilisé dans cette partie est un ResNet50, connu par ses performances [HZRS16]. Les couches de pooling de ce réseau ne sont pas utiles; elles en sont donc supprimées, tout comme la couche de décision. Ce type de réseau est utilisé pour définir un vecteur descriptif; et les couches de convolution ainsi que celles totalement connectées sont suffisantes pour atteindre ce but. Quand une image passe par ce réseau, elle devient un tenseur d'activation 3D. Ce vecteur de caractéristiques est un vecteur colonne et il est divisé en répartitions horizontales p avec une couche de "conventional average pooling" afin d'avoir un seul vecteur colonne g avec p répartition $g_i (i = 1, 2, ..., p)$. A son tour, ce vecteur sera utilisé pour comparer chaque nœud d'une tracklet T_n issue d'une caméra x avec la tracklet d'une caméra y. Durant la phase de test, chaque pièce p de g est concaténée afin de construire le vecteur final de caractéristiques G avec $G = [g_1, g_2, ..., g_p]$ qui est utilisé par la suite pour la classification et l'extraction de l'information temporelle sous forme $M = [m_1, m_2, ..., m_p]$. La figure 4.10 met en valeur le passage d'une image d'entrée à un vecteur de caractéristiques.

Les paramètres importants de ce module d'extraction des caractéristiques sont de la taille de l'image d'entrée [H,W] ainsi que celle du tenseur d'activation et son nombre de répartitions p. D'ailleurs, l'image d'entrée représente la taille des nœuds qui forment les tracklets et le nombre de répartitions est défini dans la partie expérimentation.

A cet égard et à la fin de cette section, une explication et un rappel du choix du réseau ResNet50 seront détaillés. En effet, ce réseau a pu enregistrer 28% du gain de performances par rapport à VGG-16 de Faster R-CNN.

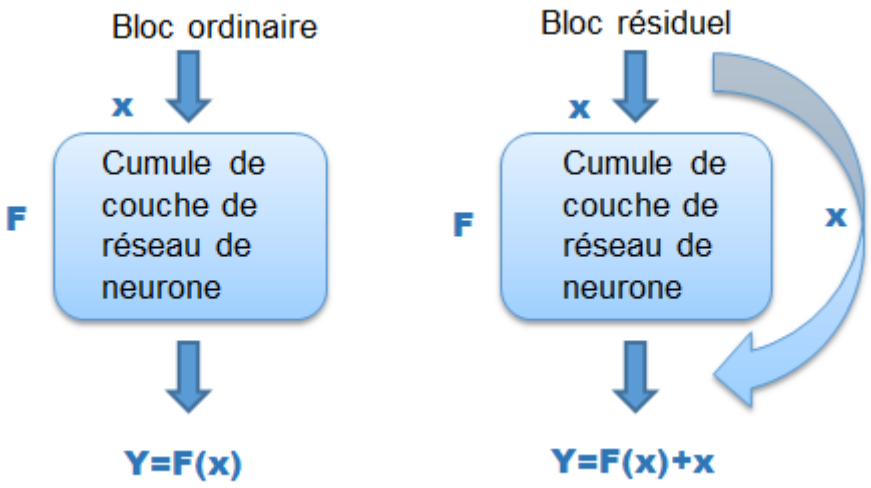


FIGURE 4.11 – Une comparaison entre un bloc de réseau de neurones et un bloc de réseau de neurones résiduel

Les réseaux de neurones convolutifs profonds ont subi une hausse remarquable d'utilisation durant ces dernières années. Ils sont utilisés dans plusieurs tâches (détection, classification, reconnaissance ...). Ainsi, au fil des années, il y a eu une tendance à aller plus loin, à résoudre des tâches plus complexes et à augmenter la précision et la profondeur de ces réseaux. D'ailleurs, quand un réseau profond commence à converger, un problème de dégradation est mis en évidence car la précision devient saturée et elle se dégrade rapidement. Par conséquent, l'utilisation de ces réseaux dégrade les performances du modèle. L'utilisation des réseaux profonds résiduels peut être une solution. Au lieu d'apprendre directement la fonction de correspondance, qui mène de x à y avec $(y = \mathrm{F}(x))$, une fonction de résidu est définie par $\mathrm{H}(x) = \mathrm{F}(x) + x$. Il est plus facile de faire tendre le résidu à zéro $(\mathrm{F}(x) = 0)$ que d'avoir $\mathrm{F}(x) = x$ par le cumul de couches CNN non-linéaires si l'identifiant des correspondances est optimal. Une comparaison entre un bloc de CNN ordinaire et un résiduel est illustré dans la figure 4.11.

4.4.2 Extraction de l'information temporelle par LSTM

Dans cette thèse, des séquences d'images successives, issues de vidéos, sont traitées car les vidéos sont les plus proches des scènes de ré-identification réelles. Les tracklets issues de ces séquences sont l'entrée de notre architecture de ré-identification. L'information temporelle est extraite de ces tracklets.

Les réseaux récurrents sont les meilleurs pour extraire les informations temporelles. Les LSTMs sont faites spécialement pour analyser les dépendances des données séquentielles (détaillées dans le chapitre précédent). Dans les travaux précédents, ce type de problème est résolu par "average ou max pooling" sur des trajectoires pour obtenir une similarité globale ou pour sélectionner la similarité par le dernier nœud de la trajectoire. Ces méthodes présentent des limites, notamment au niveau du traitement des vidéos, dues à la grande variabilité et complexité des données à traiter. En fait, la valeur moyenne ou maximale ("average ou max pooling") ne peut présenter totalement l'information des trajectoires. Bien que ces méthodes puissent apprendre l'information la plus pertinente, il est plus simple et plus sûr d'utiliser les couches cachées de LSTM.

Une unité de LSTM est basée sur ses trois portes : celle d'entrée i_t, de sortie o_t et celle

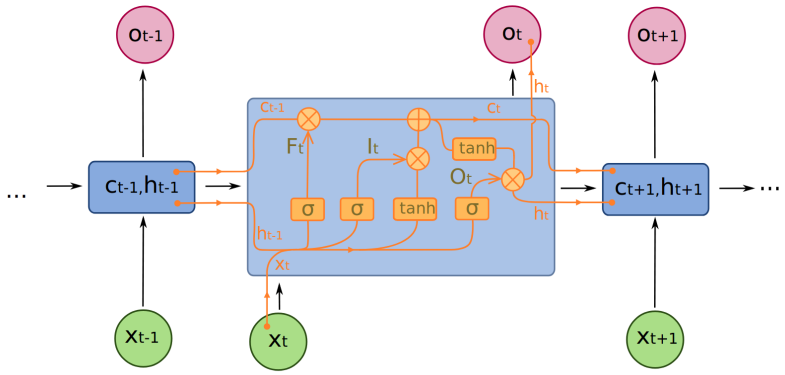


FIGURE 4.12 – Une unité de LSTM

de l'oubli f_t, ainsi que le vecteur d'état g_t. La figure 4.12 montre une unité de LSTM ainsi que ses portes. L'interaction entre les états et les portes est définie par l'équation 4.2 :

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} \mathrm{M} \begin{pmatrix} h_{t-1} \\ m_t \end{pmatrix}, c_t = f_t \odot c_{t-1} + i_t \odot g_t, h_t = o_t \odot tanh(c_t). \tag{4.2}$$

Avec : c_t encode l'état actuel.

h_t encode l'état caché.

m_t : le vecteur qui provient du vecteur de caractéristiques de l'étape précédente.

L'opération $\odot$ représente la multiplication par éléments, utilisant le vecteur $\mathrm{M} = [m_1, m_2, ..., m_p]$, issu du réseau de convolution. Une seule unité de LSTM, qui comporte la séquence des états cachés $(h_1, ..., h_n)$, s'avère insuffisante. Empiler des unités de LSTM augmente la précision et la capacité discriminante du réseau par la liaison des couches cachées, la sortie de l'une sera l'entrée de la couche suivante. Après la validation des expérimentations, deux unités de LSTM seront suffisantes.

4.4.3 Comparaison et décision

Une des contributions de cette thèse est la méthode de comparaison entre les échantillons issus de chaque caméra. Il y en a trois : une comparaison entre les échantillons (un par un), entre les trajectoires (multiple par multiple) et entre échantillon et trajectoire (un par multiple).

Étant donné que le changement qu'un objet peut subir lors du passage d'une caméra à une autre est le changement d'environnement, la comparaison échantillon par échantillon sera très difficile à résoudre pour la tâche de ré-identification. De plus, la comparaison trajectoire-trajectoire peut ne pas prendre en charge de petits détails et des informations qui seront utiles pour distinguer des objets et faire face à la similarité inter-classe.

Dans ce qui suit, la stratégie de comparaison un (une détection) par multiple (une tracklet) est adoptée afin de comparer chaque nœud d'une tracklet d'une caméra x aux tracklets d'une caméra y. L'entrée de cette phase de décision est les couches cachées du LSTM $(h_1, h_2, ..., h_n)$ et le vecteur descriptif m_t du nœud de la tracklet. La sortie est un score de similarité. Cette phase de décision est basée sur un réseau de calcul de distance qui sert à fusionner les nœuds de similarité situés dans différents points de vue.

Pour faciliter la compréhension de la stratégie de calcul de similarité, une comparaison unitaire est détaillée en premier entre une couche cachée de LSTM h_i et un vecteur

de caractéristiques m.

En effet, la similarité entre la couche cachée h_i et le vecteur m est modelée par un réseau composé de deux couches totalement connectées. La sortie de la dernière couche du réseau est une valeur z_i et la similarité est exprimée par y_i (4.3) :

$$y_i = \frac{1}{1+e^{-z_i}} \tag{4.3}$$

L'apprentissage du réseau de similarité est reformulé afin de minimiser l'erreur. Les N caractéristiques issues de la première couche totalement connectée $fc1$ sont $\mathrm{X} = \{x_1, x_2, ..., x_n\}$; et chacun porte un label $\widehat{y_k} \in \{0,1\}$ avec $k \in [0, \mathrm{N}]$. Le "zéro" signifie qu'il n'y a pas de similarité et le "un" dénote la similarité. La fonction d'erreur est définie comme suit (4.4) :

$$\mathrm{L} = \frac{1}{\mathrm{N}} \sum_{k=1}^{\mathrm{N}} (\widehat{y_k} log(y_k) + (1-\widehat{y_k}) log(1-y_k)) + \lambda \parallel \mathrm{W}_i \parallel^2 \tag{4.4}$$

Ainsi, ces résultats passeront par les nœuds de fusion afin de générer une sortie scalaire qui contrôle les poids de la fusion de similarité. Ces nœuds de fusion seront transmis couche par couche avec une structure arborescente pour fusionner les résultats précédents en vue d'avoir à la fin une similitude globale. La fusion de bas niveau $\mathrm{F}n_{ij}$, dans une première étape, fusionne les nœuds de similarité unitaire du $i$ème nœud du niveau inférieur avec le $j$ème nœud du niveau voisin supérieur et ε_{ij} est la valeur intermédiaire entre les deux niveaux (4.5) :

$$\varepsilon_{ij} = v_{ij}^{\mathrm{T}}(x_{ij}) \tag{4.5}$$

v_{ij} est le paramètre du nœud de fusion et x_{ij} est le vecteur de caractéristiques issu de la première couche $fc1$ du calcul de similarité unitaire. Chaque niveau inférieur de la structure arborescente est lié à des nœuds de similarité unitaire et la sortie de ce niveau g_{ij} est un score pondéré normalisé par la somme de tous les nœuds de fusion : (4.6).

$$g_{ij} = \frac{e^{\varepsilon_{ij}}}{\sum_i e^{\varepsilon_{ij}}} \tag{4.6}$$

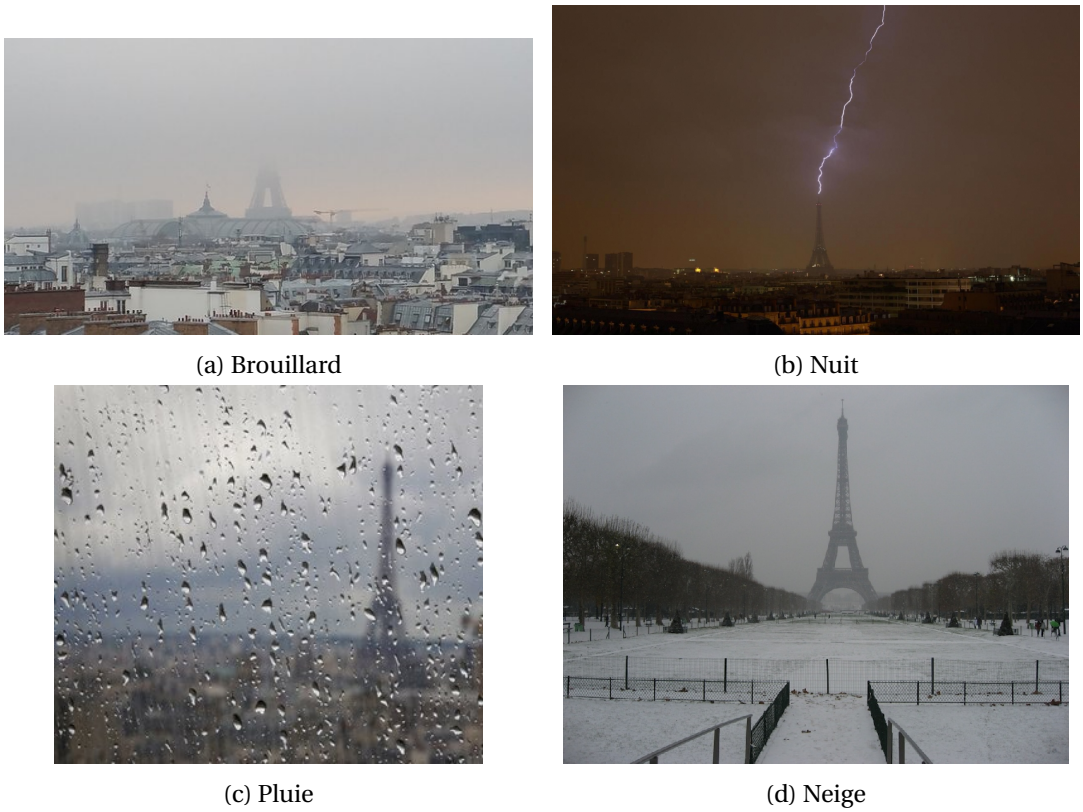
De même, pour un niveau supérieur de la structure arborescente, la valeur intermédiaire v_j et le score pondéré g_j sont re-calculés. Avec ce type de structure, les paramètres de fusion des nœuds sont mis à jour dans la propagation du réseau et une fois la similarité converge, la stratégie de fusion est obtenue, d'où le calcul de la similarité globale s_g : (4.7).

$$p(s_g) = \sum_j g_j \sum_i g_{ij} p_i(y) \tag{4.7}$$

Cette équation montre que la similarité de tous les niveaux de la structure arborescente sont multipliés par les nœuds de fusion afin d'avoir un score de similarité globale.

4.5 Approche de ré-identification dans des conditions particulières

Notre approche de ré-identification est capable de remédier à des problématiques liées à des conditions particulièrement dégradées. Ces conditions ont un effet sur les performances de la ré-identification comme les conditions météorologiques (pluie, brouillard, nuit...). La figure 4.13 montre l'effet des conditions météorologiques sur un objet (La Tour Eiffel), rendant la détection et l'identification difficiles.



(a) Brouillard (b) Nuit

(c) Pluie (d) Neige

FIGURE 4.13 – L'effet des conditions météorologiques sur l'apparence d'un objet

Dans cette thèse, une base de données, qui présente une scène filmée pendant le jour et la nuit, est utilisée afin de mettre en valeur la robustesse de notre architecture de ré-identification qui fait face à la dégradation des performances durant la nuit. Cette base de données et les résultats expérimentaux seront détaillés dans la section 4.6. En fait, la particularité de cette base se résume dans le changement que subit l'arrière-plan et l'apparence des piétons.La figure 4.14)représente un exemple d'images de la base.

Dans ce cas, un pré-traitement de reconstruction des tracklets est obligatoire afin de se libérer des dépendances liées au milieu et au temps. Ainsi, après la reconstruction des tracklets de la scène de jour dans la scène de nuit, la similarité entre ces dernières devient plus pertinente, ce qui facilite, par conséquent, la ré-identification. Dans ce cas, un transfert des tracklets est fait dans les deux sens, c'est à dire la reconstruction des tracklets se fait de la scène de jour vers celle de nuit et vice versa. Toutes les tracklets de la caméra de jour seront reconstruites dans la scène de la caméra de nuit. Si l'on suppose qu'il y a n tracklets issues de la caméra de jour et m tracklets issues de celle de nuit, après le transfert, le nombre des tracklets dans la caméra de nuit devient $n+m$. L'algorithme 1 et 2 détaillent les différentes étapes de ré-identification sur les deux scènes de jour et de nuit, issues de la même caméra avec les mêmes piétons. Le but de ces algorithmes est d'associer les tracklets issues des deux caméras. L'algorithme 2 détaille les étapes de décision qui feront partie de l'algorithme 3, et ce afin de concevoir la ré-identification dans les deux scènes de jour et de nuit.

La première étape consiste à extraire un vecteur de caractéristiques pour chaque nœud des tracklets. L'extraction des caractéristiques ne se limite pas aux vraies tracklets mais traite aussi celles reconstruites et qui seront associées aux vraies tracklets des scènes. En sortie de cette phase, un vecteur de caractéristiques G = $[g_1, g_{2,\ldots,g_p}]$ est construit pour chaque nœud des tracklets. Puis, dans une étape d'extraction de l'information temporelle,

chaque tracklet des différentes caméras passe par les deux LSTM pour obtenir l'information temporelle issue des couches cachées. Le vecteur de caractéristiques ainsi que les sorties des couches cachées passent par un réseau de similarité afin d'avoir un score de similarité par unités. Ensuite une structure arborescente fusionne les sorties pour avoir un score de similarité globale.

Grâce à cette architecture, la ré-identification dans des conditions météorologiques dégradées, spécialement le passage jour/nuit, est devenue plus simple, tout en rappelant que l'utilisation de notre architecture est motivée par le fait qu'elle limite les problématiques liées à la détection et au suivi.

(a) Acquisition pendant le jour (b) Acquisition pendant la nuit

FIGURE 4.14 – Exemple d'image de la base LATIS-IP

Fonction décision(h,m : **couche cachée,vecteur caractéristique**) : **double**
[Calcul similarité sur une unité]
Répéter
$y_i = \frac{1}{1+e^{-z_i}}$
jusqu'à ce que ($L = \frac{1}{N}\sum_{k=1}^{N}(\widehat{y_k} log(y_k) + (1-\widehat{y_k}) log(1-y_k)) + \lambda \parallel W_i \parallel^2$;)
[fusion bas niveau]
h$[h_1, h_2, ..., h_n]$% n couches cachées
Pour (i=1 ; i ≤ n ; i++) **faire**
[Calcul de valeur intermédiaire]
$\varepsilon_{ij} = v_{ij}^{T}(x_{ij})$
[calcul du score pondéré]
$g_{ij} = \frac{e^{\varepsilon_{ij}}}{\sum_i e^{\varepsilon_{ij}}}$
Fin Pour
[Calcul de la similarité globale]
$p(s_g) = \sum_j g_j \sum_i g_{ij} p_i(y)$
Retourner score de similarité ;
Fin

Algorithme 2 – Algorithme de décision

Fonction REID jour/nuit(Tx,Ty : **Tracklet**) : **Trajectoire**
résultat : **trajectoire**
[Transfert des tracklets]
Pour (x=0 ; x ≤ length Tx ; x++) **faire**
[Fonction de correspondance]
[G :Tx → Ty]
G(Tx)=$\widehat{Ty}$
[F :Ty → Tx]
F(Ty)=$\widehat{Tx}$
Fin Pour
[Extraction des caractéristiques]
T=[Tx+$\widehat{Ty}$,Ty+$\widehat{Tx}$]
Tx=[$O_{1x}, O_{2x}, ..., O_{nx}$]
$\widehat{Ty}$=[$O_{1\hat{y}}, O_{2\hat{y}}, ..., O_{n\hat{y}}$]
Ty=[$O_{1y}, O_{2y}, ..., O_{ny}$]
$\widehat{Tx}$=[$O_{1\hat{x}}, O_{2\hat{x}}, ..., O_{n\hat{x}}$]
Pour (m=0 ; m ≤ length T ; m++) **faire**
[Apprentissage]
Tx+$\widehat{Ty}$ ← Gx
Ty +$\widehat{Tx}$ ← Gy
Fin Pour
Pour (i=1 ; i ≤ p ; i++) **faire**
[Répartition des tenseurs]
Gx=[$g_{1x}, g_{2x}, ..., g_{px}$]
Gy=[$g_{1y}, g_{2y}, ..., g_{py}$]
Fin Pour
[Extraction de l'information temporelle]
M[$m_1, m_2, ..., m_p$]=G[$g_1, g_2, ..., g_p$] ;
Répéter
$c_t = f_t \odot c_{t-1} + i_t \odot g_t$,
$h_t = o_t \odot tanh(c_t)$
jusqu'à ce que ($c_t \leftarrow c_n$ & $h_t \leftarrow h_n$;)
[Décision]
Algorithme 1
Retourner résultat ;
Fin

Algorithme 3 – Algorithme de ré-identification appliqué à une même scène filmée pendant le jour et la nuit

4.6 Expérimentation

Dans cette section , les bases de données, les métriques et les résultats sont présentés afin de mettre en relief la robustesse de notre architecture de ré-identification. Cette section présente les expérimentations effectuées et les résultats obtenus. Une application de l'approche proposée dans les sections précédentes, utilisant des bases publiques et privées, est également présentée. Les contributions au niveau de la ré-identification les-

TABLEAU 4.1 – Les bases de données utilisées et leurs caractéristiques

Bases de données	Caractéristiques
PRID 2011	2 caméras avec des champs disjoints
Market-1501	6 caméras avec des champs disjoints
VIPeR	2 caméras avec deux angles de vue
i-LIDS	2 caméras avec des champs disjoints
Notre base "LATIS-IP"	Séquence vidéo avec des champs joints

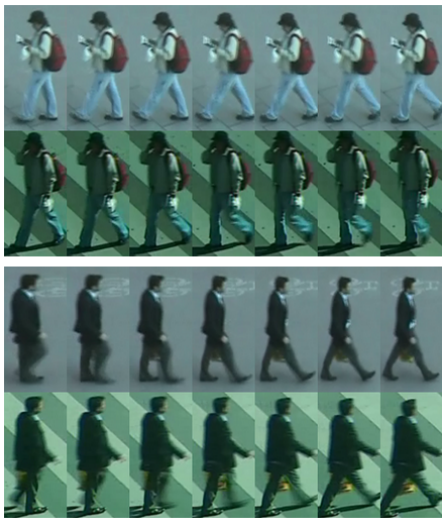


FIGURE 4.15 – Exemple d'image de la base PRID 2011

quelles liées aux reconstructions des tracklets et à la réparation des vecteurs de caractéristiques sont mises en valeur par des expérimentations en vue de prouver leur atout dans l'architecture de ré-identification. Par conséquent, l'évaluation de la chaîne s'effectue en deux étapes. Dans la première, une comparaison se fait avec et sans les contributions. Puis, dans la deuxième, nos résultats sont comparés à des travaux de l'état de l'art.

4.6.1 Base de données

Dans la ré-identification, plusieurs bases de données publiques ont été mises à disposition afin de valider les approches de ré-identification. Chaque base de données a des particularités qui la distinguent des autres. Une sélection des bases est utilisée dans cette phase, le tableau 4.1 illustre les bases utilisées pour valider notre approche ainsi que leurs caractéristiques.

PRID 2011 [HBRB11]

Cette base de données contient 385 trajectoires issues de la caméra A et 749 de la caméra B. 200 personnes apparaissent dans les deux caméras. 100 personnes seront utilisées pour l'apprentissage et 100 pour le test. Un exemple de la base est présenté dans la figure 4.15.

Market-1501 [ZST+15]

La base de données Market-1501 est collectée devant un supermarché de l'université Tsinghua. Elle contient 6 caméras au total dont 5 de haute résolution et une de faible

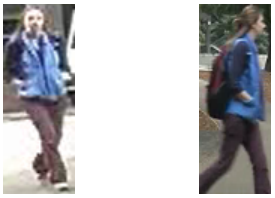
FIGURE 4.16 – Exemple d'image de la base Market-1501

résolution. Ces caméras ont un champ joint qui contient 32668 échantillons dont 1501 identifiants. 751 identifiants de 32668 images ont été utilisées pour l'apprentissage et 750 issues de 19732 pour le test. La figure 4.16 montre un exemple de la base.

VIPeR [GBT07]

Une des bases les plus utilisées dans les travaux de l'état de l'art présente des défis importants pour la ré-identification et est acquise dans des conditions météorologiques normales. Cependant, la résolution des images est faible, les personnes sont prises avec plusieurs angles de vue et les conditions d'illumination sont variées. Elle contient 632 piétons présentés par deux images issues de chaque caméra, ce qui ne permet pas de traiter cette base comme une séquence vidéo servant à extraire des tracklets, raison pour laquelle l'information temporelle est perdue. Un exemple des paires de piétons de cette base est montré dans la figure 4.17. 316 identifiants ont été utilisés pour l'apprentissage et 316 pour le test.

(a) P1 cam1 (b) P1 cam2



(c) P2 cam1 (d) P2 cam2

(e) P3 cam1 (f) P3 cam2

FIGURE 4.17 – Exemple d'image des personnes de la base de VIPeR

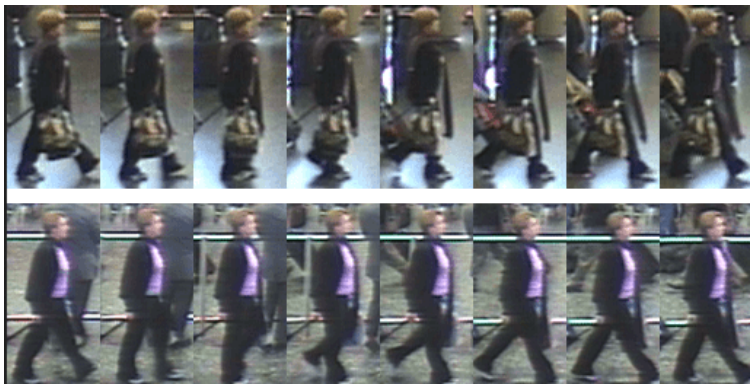


FIGURE 4.18 – Exemple d'image de la base i-LIDS

i-LIDS [ZGX11]

L'acquisition de cette base est faite dans un hall d'aéroport. Elle est composée de 2 caméras qui ont un champ disjoint. Cette base est composée de 600 séquences d'images avec 300 personnes avec deux vues issues de chaque caméra. Chaque séquence d'images varie entre 23 à 192 images avec une moyenne de 73. Cette base présente beaucoup de challenges dus à la similarité des tenues vestimentaires, la luminosité, les différents angles de vue, l'arrière-plan encombré et les occultations. 150 identifiants ont été utilisées pendant l'entraînement et 150 pour le test. La figure 4.18 présente deux trajectoires pour deux personnes différentes.

Our dataset LATIS-IP

Notre base de données est composée de deux sous-bases, une pour la ré-identification dans des conditions météorologiques normales : deux caméras de 12000 séquences d'images avec champs joint qui contiennent 11 personnes dont 6 apparaissent dans les deux caméras. La figure 4.19 illustre deux images de la première sous-base. La deuxième sous-base, filmée par la même caméra mais à deux moments différents (le jour et la nuit), est composée de 12000 images. Elle contient 4 personnes. La figure 4.20 illustre les deux images des deux caméras. 6000 images ont été utilisées pour l'apprentissage et 6000 pour le test.

(a) cam1 (b) cam2

FIGURE 4.19 – Exemple d'image de notre sous-base 1

(a) cam1 : la nuit (b) cam1 : le jour

FIGURE 4.20 – Exemple d'image de notre sous-base 2

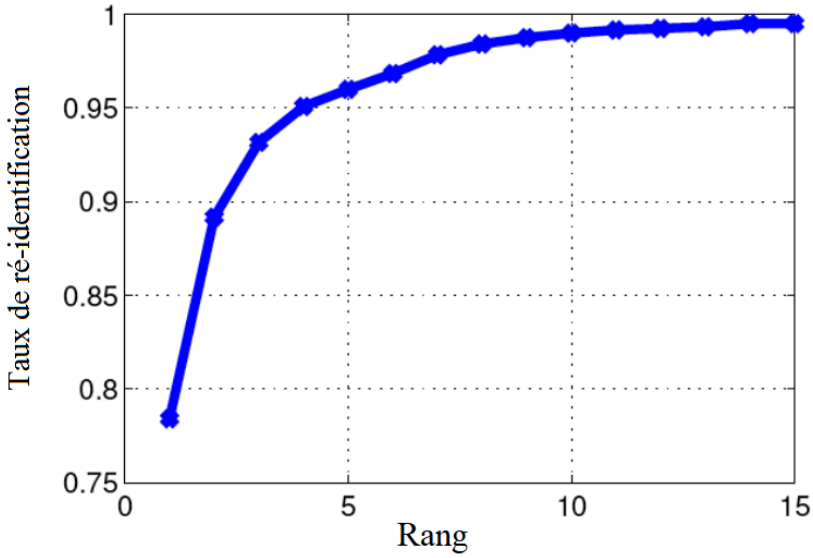


FIGURE 4.21 – Exemple de courbe de correspondance cumulée CMC

4.6.2 Métrique

Pour l'évaluation des performances de la ré-identification, une des métriques, la plus utilisée, est la courbe de correspondance cumulée "Cumulative Match Characteristic" (CMC). La ré-identification s'effectue pour le multi-objets pour en identifier un parmi n références. Par conséquent, grâce à cette courbe, un calcul de probabilité est réalisé pour identifier la bonne personne recherchée entre les r meilleurs appariements. r intitule le rang de la ré-identification et sa première valeur rang 1 est appelée "Correct Classification Rate"(CCR). Ainsi, cette courbe retourne le pourcentage de l'objet ré-identifié par rapport au rang de ré-identification. La CMC est une courbe exponentielle qui converge à 100% de ré-identification (cf. figure 4.21).

4.6.3 Résultats et discussions

Dans cette partie de la thèse, les résultats de notre approche appliquée aux différentes bases de données sont présentés avec une analyse des résultats. Plus précisément, les résultats sont exposés en trois niveaux afin de mettre en valeur nos contributions. Dans un premier temps, l'approche est présentée avec et sans la répartition du vecteur de caractéristiques (OPD et PPD). Puis, les résultats sont présentés avec et sans transfert des

TABLEAU 4.2 – Comparaison de notre approche avec une description globale et avec répartition des personnes par la courbe CMC (%) appliquée aux bases de données PRID-2011 et VIPeR

	PRID-2011				VIPeR			
Nos approches	R-1	R-5	R-10	R-20	R-1	R-5	R-10	R-20
OPD	71.8	89.1	95.4	97.8	45.1	75.6	88.1	93.5
PPD	79.3	96.1	97.3	98.6	49.6	78.2	89.7	95.8

tracklets; (T-PPD, S-PPD) sont exposés afin de montrer l'impact du transfert sur l'architecture de la ré-identification. A la fin, une comparaison avec d'autres travaux de l'état de l'art est décrite. Nos résultats sont démontrés de la façon suivante :

OPD : "Overall pedestrian description" le piéton est décrit d'une façon globale sans répartition.

PPD : "Part pedestrian description" le piéton est décrit par parties et le nombre de répartitions varie de 1 à 6.

T-PPD : Un pré-traitement est fait dans ce cas-là avec un transfert des tracklets d'une caméra à une autre.

S-PPD : Dans ce cas, le transfert est basé sur des échantillons et non pas sur des tracklets et ce pour augmenter le volume des données lors de l'apprentissage.

Évaluation de la description des piétons par répartitions

A ce niveau là, une comparaison entre nos deux architectures OPD et PPD est décrite afin de permettre le jugement de la qualité des résultats de description par parties des piétons. En effet, une étude comparative entre les résultats appliqués sur deux bases différentes est démontrée. Le tableau 4.2 montre que la description par répartition produit de meilleurs résultats.

La base PRID-2011 : D'après notre description dans la section des bases de données 4.6.1, la base PRID-2011 est une base qui présente de multi-échantillons. Il s'agit d'une scène complexe à traiter qui filme des personnes dans des conditions non contrôlées et réelles. La variation d'éclairage présente aussi une contrainte. Cette base est filmée à l'aide de deux caméras de surveillance à deux endroits différents de la rue. Le tableau 4.2 montre une comparaison des valeurs de CMC de rang 1 à 20 et prouve que l'approche fondée sur la description par répartition est meilleure que celle fondée sur une description globale de la personne. D'ailleurs, l'approche PPD réalise une amélioration de 7.5% pour sa valeur de CCR et de 7.2% pour sa valeur de rang 5.

La base VIPeR : Le même test est appliqué à la base VIPeR. Cependant, cette base ne contient pas de séquences d'images mais plutôt des images de 632 personnes filmées sous différents angles de vue par deux caméras de champs de vue disjoints. Il s'agit d'une base mono-échantillon. Dans cette thèse, elle est utilisée pour prouver la performance de l'utilisation de la description par parties et pour montrer l'effet de cette contribution sur la valeur de CCR qui a pu être améliorée de 4.5%.

Notamment, le nombre de répartitions influe sur les performances, comme le montre le tableau 4.3. Le nombre de répartitions p varie entre 1 à 6. Lorsque $p = 1$, la description est considérée comme globale. Le tableau indique que plus le nombre de répartitions augmente, plus la valeur CCR est meilleure. Elle augmente de 5.6% entre $p = 1$ et $p = 4$ et de 1.9% entre $p = 4$ et $p = 6$ pour la base PRID-2011.

En réalité, même si la description par répartitions augmente les performances, elle reste, elle aussi, influençable par le nombre de répartitions. Selon la valeur de CCR de la fi-

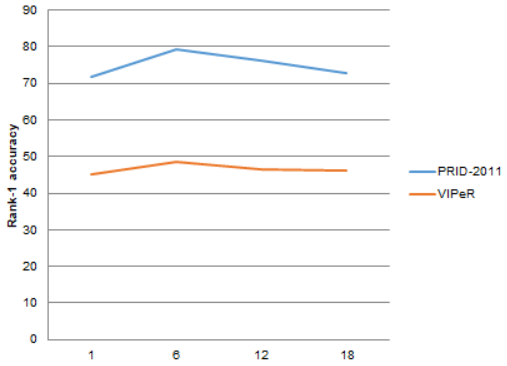


FIGURE 4.22 – La dégradation des performance au-delà de 6 répartitions

TABLEAU 4.3 – L'influence du nombre de répartition p sur la valeur CCR (%) appliquée à la base PRID-2011 et VIPeR.

p	1	2	3	4	5	6
PRID-2011	71.8	73.9	74.2	77.4	78.1	79.3
VIPeR	45.1	45.9	46.3	47.1	47.5	49.6

gure 4.22, au delà de 6 répartitions, les performances diminuent. Ainsi, lorsque le nombre dépasse 6, le pourcentage des zones similaires ou vides augmente.

Évaluation et effet de transfert des tracklets

L'étape de transfert des tracklets est une étape de pré-traitement. Mais pour l'évaluation, elle est située à ce niveau afin d'avoir son influence sur toute l'architecture basée sur une description par parties pour les piétons. Grâce à ce pré-traitement, qui permet d'augmenter le volume des échantillons lors de l'apprentissage et de diminuer la différence de similarité entre les différentes personnes dans les différentes caméras, la ré-identification est facilitée.

Dans cette partie, le transfert des tracklets se fait dans deux scénarios différents. Afin d'évaluer l'effet du transfert, deux bases seront utilisées (LATIS-IP et Market-1501).

La base LATIS-IP, comme définie dans la sous-section précédente, est une base qui contient deux sous-bases : la base de ré-identification entre deux caméras de champs joints et la sous-base filmée par la même caméra avec deux conditions météorologiques différentes (jour/nuit). La sous-base jour et nuit est utilisée dans cette partie pour mettre en valeur le transfert des tracklets par rapport à celui des échantillons. Cette sous-base filmée la nuit présente plusieurs problématiques, à savoir les défauts de détection, la similarité entre les personnes et le croisement entre les personnes. Le transfert d'échantillons présente des performances moins bonnes que celles du transfert des tracklets car le risque d'envoyer de faux positifs augmente avec le transfert par échantillons.

En ce qui concerne l'apprentissage, un CycleGan $C_2^2 = 1$ est utilisé pour cette sous-base qui contient deux caméras. Il y aura deux transferts : le premier sera celui des tracklets de jour vers la scène de nuit et le deuxième l'inverse. Pour l'entraînement, la même stratégie de [ZYH16] est utilisée. Le tableau 4.4 montre l'effet de transfert des tracklets par rapport au transfert d'échantillons. En effet, le transfert des tracklets présente des performances plus élevées de +1.5% par rapport à celui des échantillons et ce selon le pourcentage de la valeur CCR.

La base Market-1501, décrite précédemment, contient 6 caméras. Le transfert se fait

TABLEAU 4.4 – Comparaison entre le transfert des tracklets et celui des échantillons sur la base LATIS-IP et Market-1501 selon la métrique CMC (%)

	LATIS-IP				Market-1501			
Nos approches	R-1	R-5	R-10	R-20	R-1	R-5	R-10	R-20
S-PPD	82.4	96.1	97.9	99.0	85.7	87.5	89.9	99.2
T-PPD	84.6	97.3	98.0	99.1	93.9	98.1	99.2	99.3

TABLEAU 4.5 – Comparaison de notre méthode avec d'autres travaux de l'état de l'art appliqués à la base de données VIPeR et utilisant la métrique CMC (%)

Méthodes	R-1	R-5	R-10	R-20
DML [YLLL14]	28.23	59.27	73.45	86.39
RDC [KHW+12]	15.66	38.42	53.86	70.09
Saliencel [ZOW13]	30.16	52.3	-	-
PPCA [MJ12]	19.27	48.89	64.91	80.28
CRAFT [CZZL18]	50.3	79.1	77.9	95.0
OPD	**45.1**	**75.6**	**88.1**	**93.5**
PPD	**49.6**	**78.2**	**89.7**	**95.8**

entre les 6, donc CycleGAN $C_6^2 = 15$ sont utilisés pour l'apprentissage. Les performances selon le tableau 4.4 prouve aussi que le transfert par tracklets appliqué à cette base a augmenté de +8.2% par rapport au transfert par échantillons. En effet, la métrique CMC caractérise la capacité du réseau de caméras à trouver la liaison entre les similarités de la manière la plus simple. Par conséquent, les résultats sur les deux bases éprouvent que le transfert des tracklets est bénéfique pour des ré-identifications complexes.

Comparaison avec les travaux précédents

Une sélection des travaux de l'état de l'art a été choisie pour être située par rapport à la littérature. Cette sélection présente des travaux qui ont des points communs avec notre approche (cf. tableau 4.6). Les quatre bases de données publiques : PRID-2011, i-LIDS, VIPeR et Market-1501 sont utilisées dans notre comparaison.

Dans une première comparaison, (cf. tableau 4.5), nous utilisons la base VIPeR avec la métrique CMC. Dans ce tableau, nous comparons nos méthodes OPD et PPD aux différentes méthodes de l'état de l'art. En fait, la base VIPeR ne contient que des échantillons où l'information temporelle n'est pas accessible. La comparaison entre les deux caméras ne se fait qu'entre les échantillons. Avec cette base, nous mettons en valeur la robustesse de notre approche pour la recherche de similarité et nous arrivons à atteindre 49.6% en CCR. Cette valeur est proche et inférieure à la méthode CRAFT [CZZL18] pour la valeur CCR et le rang 5. Puis, elle augmente un peu à partir du rang 10. Notre approche dépasse tous les rangs PPCA [MJ12], Saliencel [ZOW13], RDC [KHW+12] et DML [YLLL14] respectivement de 30.33%, 19.44%, 33.94% et 21.37%, (cf. figure 4.23).

Dans l'ensemble, nos résultats pour cette base sont meilleurs en comparaison avec la plupart des résultats. Cependant, ils n'ont pas pu dépasser certains autres; ceci est dû principalement à notre approche basée sur deux contributions majeures où la collaboration entre les deux permet d'avoir un système assez robuste puisque une seule contribution n'est pas suffisante. Plus précisément, dans cette base, la description par répartitions des piétons a été utilisée; or, ni le transfert de tracklets ni l'extraction de l'information temporelle n'avait lieu, ce qui limite les performances d'après nos analyses.

TABLEAU 4.6 – Une sélection de quelques travaux de l'état de l'art et leurs caractéristiques

Méthodes	Architecture	Base de données	Numéro de caméra
ASTPN [XCG+17]	Réseau récurrent avec le spatial-temporal pooling	MARS, iLIDS-VID, PRID-2011	Min=2, Max= 6
RNN-CNN [MMdRM16]	Réseau de neurones convolutif associé à des couches récurrentes et temporal pooling	PRID-2011, iLIDS-VID	2
RFA-Net [YNS+16]	Graph matching, réseau de LSTM et description par segmentation	iLIDS-VID, PRID-2011	2
STA [LMZH15]	Apprentissage par la probabilité visuelle au moyen du modèle Gaussian Mixture (GMMs)	iLIDS-VID, PRID-2011	2
VR [WGZW14]	Optic flow energy et HOG3D	iLIDS-VID, PRID-2011	2
AFDA [LWKR15]	Couleur d'histogramme : RGB, HSV et YCbCr	iLIDS-VID, PRID-2011, SAIVT-SoftBio	Min=2, Max= 8
DML [YLLL14]	Réseaux de neurones siamois	VIPeR, PRID-2011	2
RDC [KHW+12]	Métrique KISS metric learning	VIPeR, ToyCars	2
Saliencel [ZOW13]	Salience probability map avec K-nearest neighbors	VIPeR, CUHK	2
PPCA [MJ12]	Pairwise constrained component analysis	VIPeR	2
CRAFT [CZZL18]	Feature augmentation	VIPeR, CUHK01, CUHK03, Market-1501, QMUL GRID	Min=2, Max= 8
SVDNet [SZDW17]	Backbone networks avec eigenlayer	Market-1501, CUHK03, DukeMTMC-reID	Min=2, Max= 10
Triplet loss [HBL17]	Réseau de neurones convolutif avec triplet loss	Market-1501, MARS	Min=2, Max= 6
Multiregion [UGL17]	Deep metric learning et fine-grained recognition method	CUHK03, Market-1501, CUHK01	Min=2, Max= 10
HydraPlus-Net [LZT+17]	Multi-region bilinear	CUHK03, Market-1501	Min=2, Max= 10
PAR [ZLZW17]	description par parties et body part-aligned representation par fully convolutional neural network (FCN)	Market-1501, CUHK03, CUHK01, VIPeR	Min=2, Max= 10
FEN+FWN [SLZ+17]	Feature embedding sub-Net et feature weighting sub-Net	CUHK03, Market-1501, VIPeR	Min=2, Max= 10
PCB+RPP [SZY+17]	Convolutional baseline	Market-1501, CUHK03, DukeMTMC-reID	Min=2, Max= 10

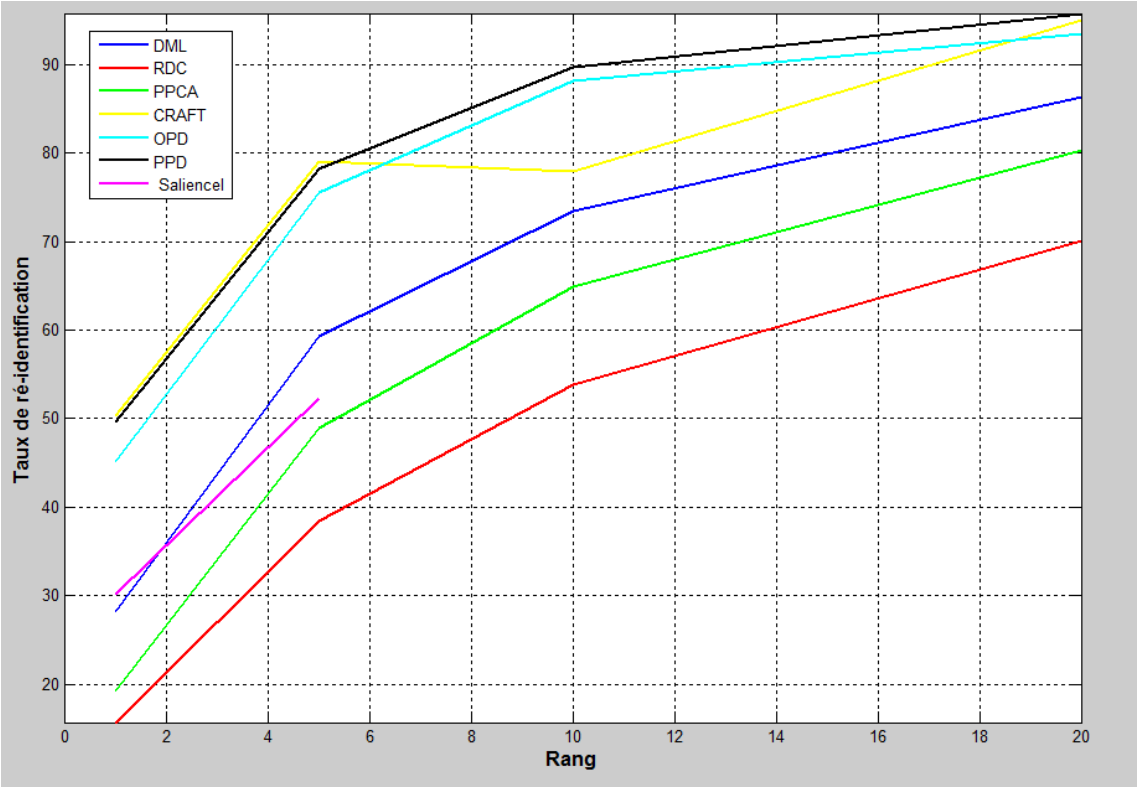


FIGURE 4.23 – Comparaison de nos approches par la courbe CMC appliquée à la base VIPeR

Une deuxième comparaison est illustrée dans le tableau 4.7 dans lequel nous comparons nos résultats appliqués aux deux bases de données PRID-2011 et i-LIDS. Dans ce cas, l'information temporelle est accessible et est extraite par LSTM. Notre résultat PPD sur la base PRID-2011 atteint 79.3%. Notre approche est plus performante pour le premier rang CCR que ASTPN [XCG+17] et RNN-CNN [MMdRM16] de +2.3% et +9.3% respectivement (cf. figure 4.24).

Appliquée sur la base i-LIDS, nous atteignons 63.8% ce qui est plus performant qu'avec [XCG+17] et RNN-CNN [MMdRM16] de +1.8% et +5.8% respectivement (cf. figure 4.25).

Notre objectif est d'augmenter le volume des données par le transfert des tracklets. Pour prouver l'effet du transfert des tracklets, nous appliquons notre méthode sur une

TABLEAU 4.7 – Comparaison de notre approche avec une sélection de travaux de l'état de l'art appliquée sur les bases PRID 2011 et i-LIDS selon la métrique CMC (%)

	PRID 2011				i-LIDS			
Méthodes	R-1	R-5	R-10	R-20	R-1	R-5	R-10	R-20
ASTPN [XCG+17]	77	95	99	99	62	86	94	98
RNN-CNN [MMdRM16]	70	90	95	97	58	84	91	96
RFA [YNS+16]	64	86	93	98	49	77	85	92
STA [LMZH15]	64	87	90	92	44	72	84	92
VR [WGZW14]	42	65	78	89	35	57	68	78
AFDA [LWKR15]	43	73	85	92	38	63	73	82
KISS+ [HZS+20]	-	-	-	-	57.98	83.33	93.78	98
OPD	**71.8**	**89.1**	**95.4**	**97.8**	**60.4**	**82.1**	**90.7**	**95.0**
PPD	**79.3**	**96.1**	**97.3**	**98.6**	**63.8**	**86.2**	**95.2**	**99.1**

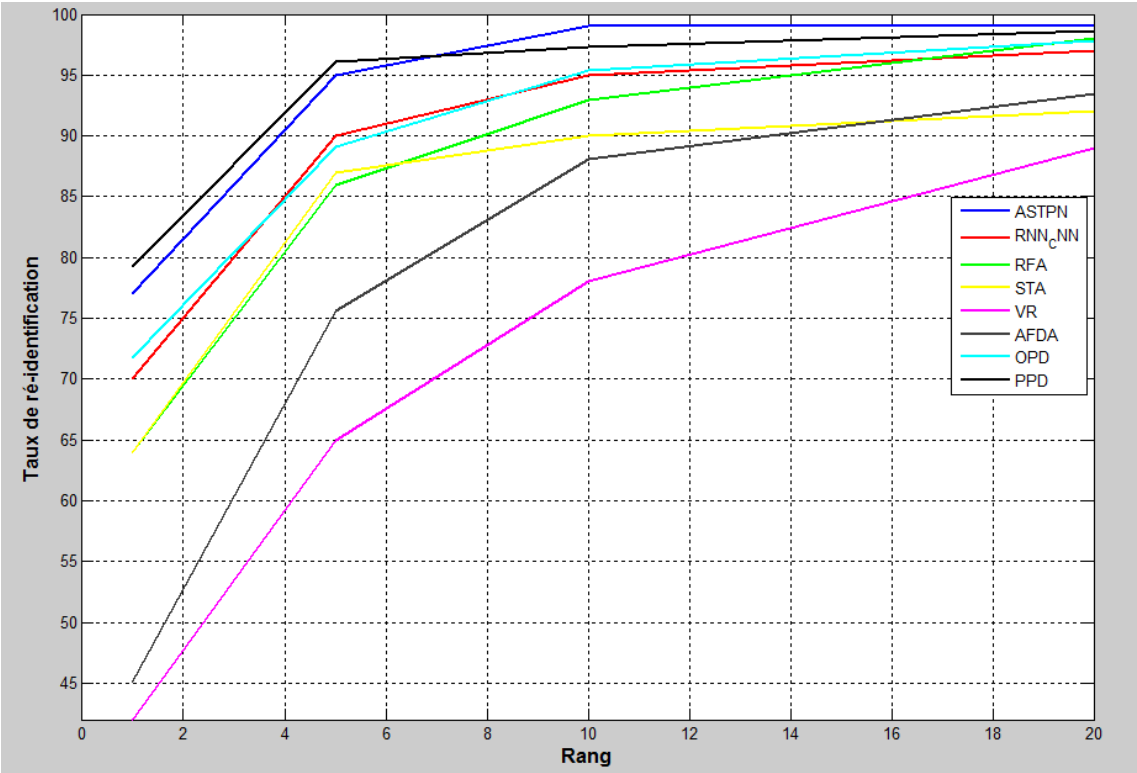


FIGURE 4.24 – Comparaison de nos approches par la courbe CMC appliquée à la base PRID-2011

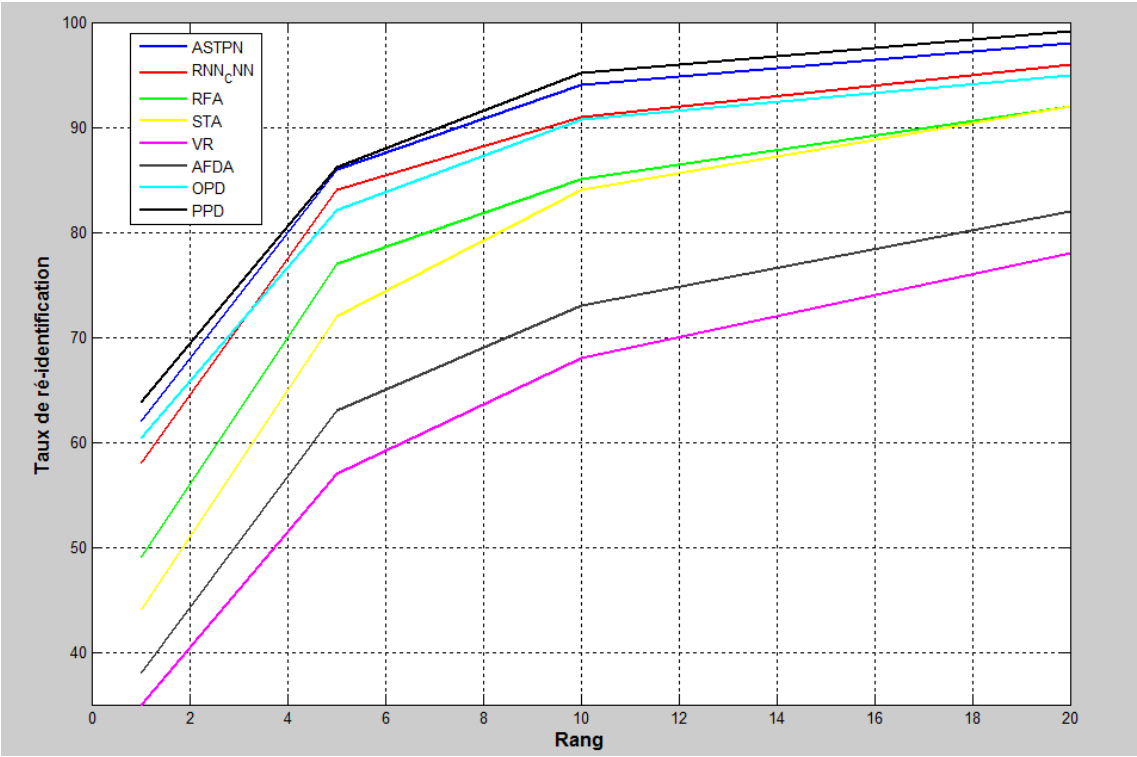


FIGURE 4.25 – Comparaison de nos approches par la courbe CMC appliquée à la base i-LIDS

TABLEAU 4.8 – Comparaison de notre approche avec l'état de l'art selon la métrique CMC (%) sur la base Market-1501

Méthodes	R-1	R-5	R-10	R-20
SVDNet [SZDW17]	82.3	92.3	95.2	-
Triplet Loss [HBL17]	84.9	94.2	-	-
MultiRegion [UGL17]	66.4	85.0	90.2	-
HydraPlus [LZT+17]	76.9	91.3	94.5	-
PAR [ZLZW17]	81.0	92.0	94.7	-
PDC [SLZ+17]	84.4	92.7	94.9	-
PCB+RPP [SZY+17]	93.8	97.5	98.5	-
S-PPD	**85.7**	**87.5**	**89.9**	**99.2**
T-PPD	**93.9**	**98.1**	**99.2**	**99.3**

base qui contient plus de 2 caméras comme l'étude sur la base Market-1501 qui contient 6 caméras. La comparaison entre notre approche et celle de l'état de l'art est illustrée dans le tableau 4.8. Avec le transfert des tracklets, notre approche est robuste et atteint 93.9% de performance, ce qui est meilleur que dans [SZY+17], [SLZ+17]...

Analyse des erreurs

Dans la dernière partie de ce chapitre, nous examinons les erreurs et les défauts des résultats et nous essayons d'en identifier les causes. En fait, la tâche de ré-identification, influençable par les tâches qui la précèdent, devient de plus en plus difficile à résoudre. Pour ce qui suit, nous allons critiquer les résultats et la différence de performance entre nos approches et les autres travaux.

Le meilleur taux de ré-identification pour la base VIPeR est 50.3%, considéré comme un taux assez faible. D'une part, il n'y avait pas assez d'informations surtout que nous ne pouvions pas extraire de l'information temporelle. De plus, la méthode CRAFT [CZZL18] présente le taux le plus élevé basé sur l'augmentation des caractéristiques ; ceci confirme que cette base présente un manque d'informations. La différence de taux de performance entre nos deux méthodes est due à la différence de méthodes d'augmentation des caractéristiques car dans ce cas-là, nous n'avons pas la possibilité d'utiliser notre contribution de transfert de tracklets. D'autre part, la comparaison entre échantillons est paralysée par le changement de luminosité et la similarité entre les piétons.

4.7 Conclusion

Dans ce chapitre, notre méthode de ré-identification, telle que détaillée, est adéquate pour des scénarios simples et complexes. Il s'agit d'un système basé sur l'augmentation du volume des données et l'extraction par parties pour un vecteur de caractéristiques robuste. Puis, une extraction de l'information temporelle est présentée ainsi qu'un module de recherche de similarité qui dépend d'une comparaison détaillée entre chaque trajectoire issue de chaque caméra et chaque échantillon provenant de différents réseaux.

Pour l'augmentation du volume des données, nous avons utilisé un transfert de tracklets qui nous garantit une augmentation d'échantillons par une méthode non supervisée, automatique et reconstruite uniquement par de vrais positifs et ce grâce à la notion des tracklets qui permet d'éliminer les faux positifs.

Ensuite, nous avons opté pour une extraction profonde des caractéristiques par répartition afin d'avoir des détails plus pertinents sur les piétons.

L'information temporelle est très utile pour le suivi et la ré-identification, en vue de laquelle nous avons utilisé un réseau récurrent à base de LSTM.

La comparaison dans le réseau est l'une de nos contributions basées sur une recherche de similarité entre une trajectoire issue d'une caméra et chaque nœud des différentes trajectoires du réseau afin de limiter l'impact du changement de pose, de luminosité et d'apparence.

Notre méthode est robuste à la ré-identification dans des scénarios simples et complexes qui contiennent plusieurs caméras ou qui sont filmés dans des conditions météorologiques particulières.

Afin d'évaluer notre méthode, quatre bases publiques et une base privée ont été utilisées, ce qui a montré des performances élevées qui dépassent la plupart des travaux de l'état de l'art. Puis, l'analyse des erreurs explique les lacunes que peut avoir notre méthode.

Chapitre 5

Conclusion et perspectives

« *Les hommes passent leur vie à la recherche d'eux-mêmes. On n'arrive jamais à une conclusion définitive en ce domaine.* »

Rosa Candida

Sommaire

5.1 Résumé des contributions **98**

5.2 Limites et perspectives **99**

5.1 Résumé des contributions

Cette thèse traite de la ré-identification multi-objets dans un réseau de caméras avec le cadre applicatif de la vidéo-surveillance. Plusieurs problèmes surgissent pour qui souhaite développer un système de suivi dans un réseau de caméras. En effet, chaque objet peut apparaître dans plusieurs caméras. Par conséquent, son apparence subit plusieurs variations dues au changement du milieu, de l'angle de vue, de la pose et de la luminosité. Les défauts du détecteur influencent aussi les performances du suivi et celles de la ré-identification. Les conditions d'acquisition et le nombre d'échantillons introduits compliquent la ré-identification.

Ainsi, l'objectif de ce travail a été de développer un système robuste de suivi multi-objets dans une seule caméra et dans un réseau de caméras. Le but était d'assurer la bonne qualité des trajectoires avec une complexité de calcul adéquate. Ceci nous a amenés à concevoir une chaîne algorithmique composée de deux parties : la première permet le suivi dans une seule caméra et la seconde permet le suivi dans un réseau de caméras, c'est-à-dire la ré-identification. Cette chaîne a permis de résoudre et de limiter plusieurs problèmes sur différents niveaux tels que les défauts du détecteur et le changement d'identifiant à cause d'une occultation ou d'un croisement. Cette chaîne s'est montrée robuste grâce à la définition d'une signature unique pour chaque objet. La signature est basée sur des caractéristiques qui restent intactes par rapport aux changements de luminosité et de pose. Ceci a permis d'avoir une très bonne qualité de suivi par rapport à des algorithmes de l'état de l'art.

Cette chaîne algorithmique est composée de deux parties : une première liée au suivi dans une caméra et une seconde liée à la ré-identification dans le réseau de caméras. Nous nous sommes d'abord intéressés à une méthode de suivi par détection, ce qui a conduit à choisir un détecteur avant de concevoir notre chaîne de suivi. Après une comparaison de différents détecteurs de l'état de l'art, nous avons choisi un des réseaux de neurones convolutifs parmi les plus performants pour effectuer la détection : le Faster R-CNN.

Cependant, comme tout système, ce détecteur présente des défauts comme les faux positifs et les non-détections. Or, le taux de ces défauts de détection a forcément un impact sur le suivi. Pour pallier ce problème, une signature a été définie pour chaque objet détecté par Faster RCNN. Elle est constituée de caractéristiques d'apparences définies à partir de l'histogramme de couleurs HSV et de caractéristiques géométriques 2D estimées par un filtre de Kalman. Ces signatures ont été utilisées pour construire les tracklets. Si les caractéristiques géométriques de deux signatures sont proches (taux de recouvrement des fenêtres d'intérêt estimé par le filtre de Kalman supérieur à un seuil), les détections sont alors associées. Si plus de deux fenêtres d'intérêt se superposent (cas de trajectoires qui se croisent), c'est l'histogramme de couleurs qui les discrimine. Le même identifiant est alors attribué à chaque détection. Au final, l'ensemble des détections, qui portent le même identifiant, constitue une tracklet. Une fois les tracklets construites, nous avons associé ces dernières afin d'avoir des trajectoires en prenant en considération l'apparition ou la disparition et la sortie de chaque objet. Ceci évite la fragmentation de la trajectoire après un croisement ou une occultation par exemple. Nous avons ajouté une phase de mise à jour à la chaîne afin de vérifier la qualité des trajectoires et l'ajout, s'il est nécessaire, des nœuds manquants par une interpolation.

Nous avons modifié notre chaîne en utilisant deux RNNs et un LSTM pour prédire et mettre à jour l'état de l'objet. Nous avons testé nos chaînes sans et avec le réseau récurrent sur des bases de données. Nous avons pu montrer grâce à des métriques que les

performances de notre algorithme dépasse celles des travaux de l'état de l'art. Nous avons également démontré sur les différentes bases de test la robustesse de notre méthode face aux faux positifs. Leur nombre est considérablement réduit par rapport aux sorties d'un seul détecteur.

Puis, des tests ont eu lieu avec la base PETS 2009 S2L1 qui présente plusieurs cas d'occultation et de croisement. Il s'agit d'une base issue d'une caméra fixe et nous avons pu enregistrer des performances jusqu'à 0.86 de valeur de MOTA, ce qui est supérieur à tous les travaux de l'état de l'art que nous avons testés à l'époque. Dans le même contexte, avec des bases issues d'une caméra fixe, nous avons testé nos architectures avec la séquence PETS 2009 S2L2 qui présente un nombre important de personnes et nous avons réussi à atteindre 0.69 de MOTP. Nos algorithmes ont été également testés sur une base de caméra mobile qui présentait différentes tailles de personnes et de nombreuses occultations. Les performances avec la métrique MOTA ont pu atteindre 0.69. Nous avons aussi prouvé que le suivi est robuste par un test avec une base filmée la nuit et une autre qui présente des multi-objets.

A plus grande échelle, nous avons voulu évaluer notre chaîne dans le cas d'un suivi dans un réseau de caméras. En fait, la première problématique que nous avons rencontrée était liée au manque d'échantillons positifs dans certaines caméras. Ceci nous a encouragés à ajouter une phase de pré-traitement où nous avons pu augmenter le volume des données de façon automatique et non supervisée. Grâce à une méthode de base d'auto-encodeur, nous avons pu transférer des tracklets d'une caméra x à une caméra y par une méthode de reconstruction. Cette reconstruction, nous a permis de transférer un lot d'échantillons positifs entre deux caméras. Ce transfert ne se produit que pour deux scénarios différents : lorsque le nombre de caméras est important ou en présence d'une grande différence d'environnement. Après ce pré-traitement, nous avons pu extraire les caractéristiques de chaque objet du réseau par une méthode d'extraction par parties basée sur un réseau de neurones convolutif et nous avons aussi extrait l'information temporelle grâce à LSTM afin de définir une signature temporelle pour les tracklets de chaque caméra. Lors de la comparaison, nous avons effectué une prise de décision par un score de similarité comparant chaque nœud d'une caméra x avec toutes les tracklets issues des autres caméras. Nous avons appliqué notre stratégie de ré-identification dans des conditions de jour et de nuit.

Ensuite, dans la phase d'expérimentation, nous avons prouvé que la description par parties est meilleure que la description globale. Nous avons montré qu'avec 6 parties les performances de la ré-identification présentent un pourcentage de CCR plus élevé. Nous avons aussi testé l'influence de la description par parties sur des bases publiques (VIPeR et PRID-2011) avec lesquelles nous avons enregistré jusqu'à 71.8% de CCR pour la base PRID-2011. En outre, l'effet de transfert de tracklets entre les caméras a été évalué sur la base Market-1501 qui présente 6 caméras et sur notre base personnelle qui présente deux scènes : la première filmée le jour et la seconde filmée la nuit. Nous avons réussi à avoir une performance plus élevée pour ces deux bases qui présentent plusieurs scénarios difficiles.

5.2 Limites et perspectives

A l'issue de nos travaux, plusieurs voies restent actives et exploitables, ce qui engendre des perspectives sur plusieurs niveaux liés à diverses problématiques. Les perspectives de nos travaux sont détaillées comme suit :

L'utilisation des tracklets limite le taux de faux positifs. Or, dans certains cas, le détecteur n'arrive pas à détecter les objets à long terme. Alors, notre approche ne peut prédire tous les nœuds manquants et par conséquent, cela influence les bonnes performances du suivi car, dans notre cas, le suivi est basé sur les détections. L'utilisation d'une méthode indépendante du détecteur devrait améliorer la qualité du suivi dans ce cas.

Au niveau de l'application, nous pouvons étendre notre architecture afin d'avoir une ré-identification en temps réel par la construction d'une trajectoire dans une caméra. Le suivi dans un réseau de caméras en temps réel s'effectue à partir de la comparaison entre le suivi de l'objet dans une seule caméra et sa trajectoire déduite dès qu'il apparaît dans une autre caméra.

Bibliographie

[AAR+17] Hayder Albehadili, Laith Alzubaidi, Jabbar Rashed, Murtadha Al-Imam, and Haider A Alwzwazy. Fast and accurate real time pedestrian detection using convolutional neural network. In *The 1 st International Conference on Information Technology (ICoIT'17)*, page 305, 2017. 17

[AJM15] Ejaz Ahmed, Michael Jones, and Tim K Marks. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3908–3916, 2015. 31

[AKV+15] Anelia Angelova, Alex Krizhevsky, Vincent Vanhoucke, Abhijit S Ogale, and Dave Ferguson. Real-time pedestrian detection with deep network cascades. In *BMVC*, volume 2, page 4, 2015. 17

[AMGC02] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2) :174–188, 2002. 19

[ARS09] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited : People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1014–1021. IEEE, 2009. 24

[BJWW15] Pablo Barros, Doreen Jirak, Cornelius Weber, and Stefan Wermter. Multimodal emotional state recognition using sequence-dependent deep hierarchical features. *Neural Networks*, 72 :140–151, 2015. 17

[Bla04] Samuel S Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1) :5–18, 2004. 19, 22, 24

[BMTVG13] Rodrigo Benenson, Markus Mathias, Tinne Tuytelaars, and Luc Van Gool. Seeking the strongest rigid detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3666–3673, 2013. 18

[BR11] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3457–3464. IEEE, 2011. 24

[BRL+11] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE transactions on pattern analysis and machine intelligence*, 33(9) :1820–1833, 2011. 64, 65

[BY14] Seung-Hwan Bae and Kuk-Jin Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1218–1225. IEEE, 2014. 64, 65, 67

[BYH+17] Xiang Bai, Mingkun Yang, Tengteng Huang, Zhiyong Dou, Rui Yu, and Yongchao Xu. Deep-person : Learning discriminative deep features for person re-identification. *arXiv preprint arXiv :1711.10658*, 2017. 33

[CGC+18] De Cheng, Yihong Gong, Xiaojun Chang, Weiwei Shi, Alexander Hauptmann, and Nanning Zheng. Deep feature learning via structured graph laplacian embedding for person re-identification. *Pattern Recognition*, 2018. 33, 34

[CGZ+16] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2016. 33, 34, 35

[CPLP16] Xiu-Yuan Chen, Xi-Yuan Peng, Jun-Bao Li, and Yu Peng. Overview of deep kernel learning based techniques and applications. *J. Netw. Intell*, 1(3) :82–97, 2016. 17

[CRH01] Yunqiang Chen, Yong Rui, and Thomas S Huang. Jpdaf based hmm or real-time contour tracking. In *null*, page 543. IEEE, 2001. 14, 16

[CZZL18] Ying-Cong Chen, Xiatian Zhu, Wei-Shi Zheng, and Jian-Huang Lai. Person re-identification by camera correlation aware feature augmentation. *IEEE transactions on pattern analysis and machine intelligence*, 40(2) :392–408, 2018. 91, 92, 95

[DABP14] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8) :1532–1545, 2014. 15, 17, 18

[DBP10] Piotr Dollár, Serge J Belongie, and Pietro Perona. The fastest pedestrian detector in the west. In *Bmvc*, volume 2, page 7. Citeseer, 2010. 18

[DCGA17] Yosra Dorai, Frédéric Chausse, Sami Gazzah, and Najoua Essoukri Ben Amara. Multi target tracking by linking tracklets with a convolutional neural network. In *VISIGRAPP (6 : VISAPP)*, pages 492–498, 2017. 15, 16

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 15, 18

[ELVG07] Andreas Ess, Bastian Leibe, and Luc Van Gool. Depth and appearance for mobile scene analysis. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 24, 61

[FGMR10] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9) :1627–1645, 2010. 17, 18

[FS09] James Ferryman and Ali Shahrokni. Pets2009 : Dataset and challenge. In *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–6. IEEE, 2009. 24, 61

[GBT07] Douglas Gray, Shane Brennan, and Hai Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *Proc. IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*, volume 3, pages 1–7. Citeseer, 2007. 12, 28, 35, 86

[GCT+14] Yanwen Guo, Ye Chen, Feng Tang, Ang Li, Weitao Luo, and Mingming Liu. Object tracking using learned feature manifolds. *Computer Vision and Image Understanding*, 118 :128–139, 2014. 12, 13, 16

[GGZC15] Quan Gan, Qipeng Guo, Zheng Zhang, and Kyunghyun Cho. First step toward model-free, anonymous object tracking with recurrent neural networks. *arXiv preprint arXiv :1511.06425*, 2015. 21

[Gir15] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. v, 21, 40, 41

[GJLY21] Jianyang Gu, Wei Jiang, Hao Luo, and Hongyan Yu. An efficient global representation constrained by angular triplet loss for vehicle re-identification. *Pattern Analysis and Applications*, 24(1) :367–379, 2021. 27

[GK96] Christoph Goller and Andreas Kuchler. Learning task-dependent distributed representations by backpropagation through structure. *Neural Networks*, 1 :347–352, 1996. 21

[GLVP18] Zhixin Guo, Wenzhi Liao, Peter Veelaert, and Wilfried Philips. Occlusion-robust detector trained with occluded pedestrians. SCITEPRESS-Science and Technology Publications, 2018. 18

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 34

[HBL17] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv :1703.07737*, 2017. 92, 95

[HBRB11] Martin Hirzer, Csaba Beleznai, Peter M Roth, and Horst Bischof. Person re-identification by descriptive and discriminative classification. In *Scandinavian conference on Image analysis*, pages 91–102. Springer, 2011. 85

[HMD15] Song Han, Huizi Mao, and William J Dally. Deep compression : Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv :1510.00149*, 2015. 40

[HOBS15] Jan Hosang, Mohamed Omran, Rodrigo Benenson, and Bernt Schiele. Taking a deeper look at pedestrians. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4073–4082, 2015. 18, 21

[HRMZ+15] Seyed Hamid Rezatofighi, Anton Milan, Zhen Zhang, Qinfeng Shi, Anthony Dick, and Ian Reid. Joint probabilistic data association revisited. In *Proceedings of the IEEE international conference on computer vision*, pages 3047–3055, 2015. 20, 22, 24

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8) :1735–1780, 1997. 51

[HYLL09] Wei He, Takayoshi Yamashita, Hongtao Lu, and Shihong Lao. Surf tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1586–1592. IEEE, 2009. 12

[HYLYY16] Ju Hong Yoon, Chang-Ryeol Lee, Ming-Hsuan Yang, and Kuk-Jin Yoon. Online multi-object tracking via structural constraint event aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1392–1400, 2016. 65

[HYW+20] Liqin Huang, Qingqing Yang, Junyi Wu, Yan Huang, Qiang Wu, and Jingsong Xu. Generated data with sparse regularized multi-pseudo label for person re-identification. *IEEE Signal Processing Letters*, 27 :391–395, 2020. 34

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 78

[HZS+20] Hua Han, MengChu Zhou, Xiwu Shang, Wei Cao, and Abdullah Abusorrah. Kiss+ for rapid and accurate pedestrian re-identification. *IEEE Transactions on Intelligent Transportation Systems*, 22(1) :394–403, 2020. 93

[JMCB06] MH Jaward, L Mihaylova, N Canagarajah, and D Bull. A data association algorithm for multiple object tracking in video sequences. In *Target Tracking : Algorithms and Applications, 2006. The IEE Seminar on (Ref. No. 2006/11359)*, pages 129–136. IET, 2006. 19, 22

[KCM04] Jinman Kang, Isaac Cohen, and Gerard Medioni. Object reacquisition using invariant appearance model. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 759–762. IEEE, 2004. v, 13, 14, 16

[KHK07] Takuya Kobayashi, Akinori Hidaka, and Takio Kurita. Selection of histograms of oriented gradients features for pedestrian detection. In *International conference on neural information processing*, pages 598–607. Springer, 2007. 15

[KHW+12] Martin Koestinger, Martin Hirzer, Paul Wohlhart, Peter M Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2288–2295. IEEE, 2012. 91, 92

[Kim17] Du Yong Kim. Visual multiple-object tracking for unknown clutter rate. *arXiv preprint arXiv :1701.02273*, 2017. 65, 66, 67

[KL09] Junseok Kwon and Kyoung Mu Lee. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1208–1215. IEEE, 2009. 20, 22

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 17, 32, 41

[Kuh55] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2) :83–97, 1955. 47

[KUS03] Pavlina Konstantinova, Alexander Udvarev, and Tzvetan Semerdjiev. A study of a target tracking algorithm using global nearest neighbor approach. In *Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech'03)*, pages 290–295, 2003. 19, 22, 24

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553) :436, 2015. 23

[LCB13] Laetitia Lamard, Roland Chapuis, and Jean Philippe Boyer. Multi target tracking with cphd filter based on asynchronous sensors. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 892–898. IEEE, 2013. 20, 22

[LCZD01] Baoxin Li, Rama Chellappa, Qinfen Zheng, and Sandor Z Der. Model-based temporal object verification using video. *IEEE Transactions on Image Processing*, 10(6) :897–908, 2001. 14

[LCZH17] Dangwei Li, Xiaotang Chen, Zhang Zhang, and Kaiqi Huang. Learning deep context-aware features over body and latent parts for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 384–393, 2017. 29, 30

[LMZH15] Kan Liu, Bingpeng Ma, Wei Zhang, and Rui Huang. A spatio-temporal appearance representation for viceo-based pedestrian re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3810–3818, 2015. 92, 93

[LRL+17] Ji Lin, Liangliang Ren, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Consistent-aware deep learning for person re-identification in a camera network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6, 2017. 33, 34

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 17

[LTWT14] Ping Luo, Yonglong Tian, Xiaogang Wang, and Xiaoou Tang. Switchable deep network for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 899–906, 2014. 17

[LWKR15] Yang Li, Ziyan Wu, Srikrishna Karanam, and Richard J Radke. Multi-shot human re-identification using adaptive fisher discriminant analysis. In *BMVC*, volume 1, page 2. Citeseer, 2015. 92, 93

[LZT+17] Xihui Liu, Haiyu Zhao, Maoqing Tian, Lu Sheng, Jing Shao, Shuai Yi, Junjie Yan, and Xiaogang Wang. Hydraplus-net : Attentive deep features for pedestrian analysis. *arXiv preprint arXiv :1709.09930*, 2017. 92, 95

[LZXW14] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid : Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 152–159, 2014. 28, 31, 35

[Man02] A-R Mansouri. Region tracking via level set pdes without motion computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7) :947–961, 2002. 14

[MDRM15] Niall McLaughlin, Jesus Martinez Del Rincon, and Paul Miller. Data-augmentation for reducing dataset bias in person re-identification. In *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*, pages 1–6. IEEE, 2015. 34

[MHB+10] Elmar Mair, Gregory D Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *European conference on Computer vision*, pages 183–196. Springer, 2010. 12

[MJ12] Alexis Mignon and Frédéric Jurie. Pcca : A new approach for distance learning from sparse pairwise constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2666–2672. IEEE, 2012. 91, 92

[MLTSR15] Anton Milan, Laura Leal-Taixé, Konrad Schindler, and Ian Reid. Joint tracking and segmentation of multiple targets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5397–5406, 2015. 65

[MMdRM16] Niall McLaughlin, Jesus Martinez del Rincon, and Paul Miller. Recurrent convolutional network for video-based person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1325–1334, 2016. 92, 93

[MSR13] Anton Milan, Konrad Schindler, and Stefan Roth. Challenges of ground truth evaluation of multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 735–742, 2013. 25, 26

[MTC08] Emilio Maggio, Murtaza Taj, and Andrea Cavallaro. Efficient multitarget visual tracking using random finite sets. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8) :1016–1027, 2008. 20, 22

[NDH14] Woonhyun Nam, Piotr Dollár, and Joon Hee Han. Local decorrelation for improved pedestrian detection. In *Advances in Neural Information Processing Systems*, pages 424–432, 2014. 17

[NZH+17] Guanghan Ning, Zhi Zhang, Chen Huang, Xiaobo Ren, Haohong Wang, Canhui Cai, and Zhihai He. Spatially supervised recurrent convolutional neural networks for visual object tracking. In *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*, pages 1–4. IEEE, 2017. 21

[OW13] Wanli Ouyang and Xiaogang Wang. Joint deep learning for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2056–2063, 2013. 18

[Pet99] Natan Peterfreund. Robust tracking of position and velocity with kalman snakes. *IEEE transactions on pattern analysis and machine intelligence*, 21(6) :564–569, 1999. 14

[PMC13] Fabio Poiesi, Riccardo Mazzon, and Andrea Cavallaro. Multi-target tracking on confidence maps : An application to people tracking. *Computer Vision and Image Understanding*, 117(10) :1257–1272, 2013. 65, 67

[PT05] Fatih Porikli and Oncel Tuzel. Multi-kernel object tracking. In *2005 IEEE International Conference on Multimedia and Expo*, pages 1234–1237. IEEE, 2005. 20, 22

[PZRD13] Dennis Park, C Lawrence Zitnick, Deva Ramanan, and Piotr Dollár. Exploring weak stabilization for motion feature extraction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2882–2889, 2013. 18

[RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once : Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 23

[RHGS17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn : towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6) :1137–1149, 2017. 17

[SAS17] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable : Learning to track multiple cues with long-term dependencies. *arXiv preprint arXiv :1701.01909*, 4(5) :6, 2017. 21, 22, 24

[SDO+12] Guang Shu, Afshin Dehghan, Omar Oreifej, Emily Hand, and Mubarak Shah. Part-based multiple-person tracking with partial occlusion handling. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1815–1821. IEEE, 2012. 24

[SKCL13] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3626–3633, 2013. 17, 18

[SLZ+17] Chi Su, Jianing Li, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. Pose-driven deep convolutional model for person re-identification. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 3980–3989. IEEE, 2017. 29, 30, 35, 92, 95

[SM07] Payam Sabzmeydani and Greg Mori. Detecting pedestrians by learning shapelet features. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 18

[SZDW17] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. *arXiv preprint*, 1(6), 2017. 92, 95

[SZY+17] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models : Person retrieval with refined part pooling. *arXiv preprint arXiv :1711.09349*, 2017. 78, 92, 95

[TLWT15] Yonglong Tian, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Pedestrian detection aided by deep learning semantic tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5079–5087, 2015. 18, 21

[TMC06] Murtaza Taj, Emilio Maggio, and Andrea Cavallaro. Multi-feature graph-based object tracking. In *International Evaluation Workshop on Classification of Events, Activities and Relationships*, pages 190–199. Springer, 2006. 19

[UGL17] Evgeniya Ustinova, Yaroslav Ganin, and Victor Lempitsky. Multi-region bilinear convolutional neural networks for person re-identification. In *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*, pages 1–6. IEEE, 2017. 92, 95

[VJS05] Paul Viola, Michael J Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2) :153–161, 2005. 15, 18, 21

[VSL+16] Rahul Rama Varior, Bing Shuai, Jiwen Lu, Dong Xu, and Gang Wang. A siamese long short-term memory architecture for human re-identification. In *European Conference on Computer Vision*, pages 135–153. Springer, 2016. 21, 22, 24, 32

[VV13] Ba-Tuong Vo and Ba-Ngu Vo. Labeled random finite sets and multi-object conjugate priors. *IEEE Transactions on Signal Processing*, 61(13) :3460–3475, 2013. 65, 67

[WCL+16] Shangxuan Wu, Ying-Cong Chen, Xiang Li, An-Cong Wu, Jin-Jie You, and Wei-Shi Zheng. An enhanced deep feature representation for person re-identification. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–8. IEEE, 2016. 28, 30, 35

[Wer90] Paul J Werbos. Backpropagation through time : what it does and how to do it. *Proceedings of the IEEE*, 78(10) :1550–1560, 1990. 49

[WGZW14] Taiqing Wang, Shaogang Gong, Xiatian Zhu, and Shengjin Wang. Person re-identification by video ranking. In *European Conference on Computer Vision*, pages 688–703. Springer, 2014. 92, 93

[WHY09] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE, 2009. 18

[WKSL11] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011. 12

[WMSS10] Stefan Walk, Nikodem Majer, Konrad Schindler, and Bernt Schiele. New features and insights for pedestrian detection. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 1030–1037. IEEE, 2010. 18

[WN07] Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2) :247–266, 2007. 46

[WS08] Christian Wojek and Bernt Schiele. A performance evaluation of single and multi-feature people detection. In *Joint Pattern Recognition Symposium*, pages 82–91. Springer, 2008. 18

[WSvdH17] Lin Wu, Chunhua Shen, and Anton van den Hengel. Deep linear discriminant analysis on fisher networks : A hybrid architecture for person re-identification. *Pattern Recognition*, 65 :238–250, 2017. 29, 30, 35

[WWG+17] Jin Wang, Zheng Wang, Changxin Gao, Nong Sang, and Rui Huang. Deeplist : Learning deep features with adaptive listwise constraint for person reidentification. *IEEE Trans. Circuits Syst. Video Techn.*, 27(3) :513–524, 2017. 31

[WZL+16] Faqiang Wang, Wangmeng Zuo, Liang Lin, David Zhang, and Lei Zhang. Joint learning of single-image and cross-image representations for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1288–1296, 2016. v, 31, 32

[XCG+17] Shuangjie Xu, Yu Cheng, Kang Gu, Yang Yang, Shiyu Chang, and Pan Zhou. Jointly attentive spatial-temporal pooling networks for video-based person re-identification. *arXiv preprint arXiv :1708.02286*, 2017. 92, 93

[XLOW16] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1249–1258, 2016. 29, 30

[YCZ+17] Ruixing Yu, Irene Cheng, Bing Zhu, Sweta Bedmutha, and Anup Basu. Adaptive resolution optimization and tracklet reliability assessment for efficient multi-object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017. 65, 67

[YJS06] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking : A survey. *Acm computing surveys (CSUR)*, 38(4) :13, 2006. 19

[YLLL14] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Deep metric learning for person re-identification. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 34–39. IEEE, 2014. 30, 35, 91, 92

[YLS04] Alper Yilmaz, Xin Li, and Mubarak Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on pattern analysis and machine intelligence*, 26(11) :1531–1536, 2004. 14, 16

[YLY13] Fan Yang, Huchuan Lu, and Ming-Hsuan Yang. Learning structured visual dictionary for object tracking. *Image and Vision Computing*, 31(12) :992–999, 2013. 12, 16

[YNS+16] Yichao Yan, Bingbing Ni, Zhichao Song, Chao Ma, Yan Yan, and Xiaokang Yang. Person re-identification via recurrent feature aggregation. In *European Conference on Computer Vision*, pages 701–716. Springer, 2016. 92, 93

[YYLY15] Ju Hong Yoon, Ming-Hsuan Yang, Jongwoo Lim, and Kuk-Jin Yoon. Bayesian multi-object tracking using motion context from multiple objects. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 33–40. IEEE, 2015. 65, 66, 67

[YZL+13] Junjie Yan, Xucong Zhang, Zhen Lei, Shengcai Liao, and Stan Z Li. Robust multi-resolution pedestrian detection in traffic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3033–3040, 2013. 18

[ZBC14] Shanshan Zhang, Christian Bauckhage, and Armin B Cremers. Informed haar-like features improve pedestrian detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 947–954, 2014. 17, 18

[ZGX11] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Person re-identification by probabilistic relative distance comparison. 2011. 87

[ZKFT17] Fuqing Zhu, Xiangwei Kong, Haiyan Fu, and Qi Tian. Pseudo-positive regularization for deep person re-identification. *Multimedia Systems*, pages 1–13, 2017. 34, 35

[ZKWM14] Guanwen Zhang, Jien Kato, Yu Wang, and Kenji Mase. People re-identification using deep convolutional neural network. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 3, pages 216–223. IEEE, 2014. 30, 35

[ZLX+14] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014. 18, 21

[ZLZ+15] Ruimao Zhang, Liang Lin, Rui Zhang, Wangmeng Zuo, and Lei Zhang. Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. *IEEE Transactions on Image Processing*, 24(12) :4766–4779, 2015. 32, 35

[ZLZW17] Liming Zhao, Xi Li, Yueting Zhuang, and Jingdong Wang. Deeply-learned part-aligned representations for person re-identification. In *ICCV*, pages 3239–3248, 2017. 92, 95

[ZOW13] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Person re-identification by salience matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2528–2535, 2013. 91, 92

[ZPS12] Jianming Zhang, Liliana Lo Presti, and Stan Sclaroff. Online multi-person tracking by tracker hierarchy. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 379–385. IEEE, 2012. 15, 16

[ZQFX09] Chunrong Zhang, Yuansong Qiao, Enda Fallon, and Chiangqiao Xu. An improved camshift algorithm for target tracking in video surveillance. In *9th. IT & T Conference*, page 12, 2009. 20, 22

[ZST+15] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification : A benchmark. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1116–1124, 2015. 85

[ZWW+15] Shun Zhang, Jinjun Wang, Zelun Wang, Yihong Gong, and Yuehu Liu. Multi-target tracking by learning local-to-global trajectory models. *Pattern Recognition*, 48(2) :580–590, 2015. 65

[ZYH16] Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification : Past, present and future. *arXiv preprint arXiv :1610.02984*, 2016. 28, 90

[ZZK+17] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv :1708.04896*, 2017. 34, 35