Apprentissage neuronal profond pour l'analyse de contenus multimodaux et temporels

Valentin Vielzeuf

▶ **To cite this version:**

Valentin Vielzeuf. Apprentissage neuronal profond pour l'analyse de contenus multimodaux et temporels. Bio-informatique [q-bio.QM]. Normandie Université, 2019. Français. NNT : 2019NORMC229 . tel-02437035

HAL Id: tel-02437035
https://theses.hal.science/tel-02437035

Submitted on 13 Jan 2020

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

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

Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité INFORMATIQUE

Préparée au sein de l'Université de Caen Normandie

Apprentissage Neuronal Profond pour l'Analyse de Contenus Multimodaux et Temporels

Présentée et soutenue par
Valentin VIELZEUF

Thèse soutenue publiquement le 19/11/2019 devant le jury composé de		
M. MATTHIEU CORD	Professeur des universités, Université Paris 6 Pierre et Marie Curie	Rapporteur du jury
M. CHRISTIAN WOLF	Maître de conférences HDR, INSA Lyon	Rapporteur du jury
Mme ISABELLE BLOCH	Professeur des universités, Université Paris-Saclay	Président du jury
M. ALEXIS LECHERVY	Maître de conférences, Université Caen Normandie	Membre du jury
M. STÉPHANE PATEUX	Ingénieur de recherche, Orange Lab	Membre du jury
M. FREDERIC JURIE	Professeur des universités, Université Caen Normandie	Directeur de thèse

Thèse dirigée par FREDERIC JURIE, Groupe de recherche en informatique, image, automatique et instrumentation

UNIVERSITÉ CAEN NORMANDIE

GREYC

Remerciements

Si ces trois années m'ont beaucoup apporté, c'est principalement par les belles rencontres que j'ai pu y faire. Ainsi, j'aimerais exprimer toute ma gratitude à ces personnes qui ont permis de concrétiser mon travail de thèse en ce manuscrit.

Je tiens à remercier tout d'abord mes trois excellents encadrants, qui m'ont donné toutes les clés pour faire de mon doctorat une belle expérience. Frédéric a dirigé cette thèse, avec sa très grande expertise du domaine et son expérience de chercheur renommé. Je lui suis très reconnaissant de m'avoir transmis notamment son goût pour la recherche, pour la rigueur et pour l'honnêteté scientifique. Alexis, que j'ai rencontré au cours de mon premier séjour à Caen, a accepté d'apporter également sa vision scientifique et ses conseils pertinents à ma thèse et je le remercie sincèrement de m'avoir posé les bonnes questions au bon moment et de m'avoir accordé beaucoup de temps. Stéphane m'a encadré du côté d'Orange, malgré ses importantes responsabilités par ailleurs, avec son immense culture scientifique, son humour et sa bienveillance. Merci d'avoir stimulé ma curiosité par tous ces échanges, d'avoir été toujours disponible et de m'avoir beaucoup appris sur de nombreux plans.

J'aimerais ensuite remercier les membres du jury, qui ont accepté d'analyser et de discuter ce travail. Je remercie tout d'abord Madame Isabelle Bloch, Professeur à Telecom Paris Tech, d'avoir présidé le jury de cette soutenance, avec expérience et gentillesse. Je remercie également Monsieur Matthieu Cord, Professeur au LIP6 et Monsieur Christian Wolf, Maître de Conférences à l'INSA de Lyon, d'avoir rapporté mon manuscrit et qui, de par leurs questions enrichissantes, leur bienveillance et leurs conseils, ont contribué à l'amélioration de ce manuscrit.

Au cours de ma thèse, j'ai été accompagné par un comité de suivi, composé de Patrick Bouthemy et Pierrick Philippe. Merci de m'avoir accordé votre temps et vos conseils pertinents à plusieurs reprises.

J'ai eu l'occasion de passer un quart de chaque année à Caen, au sein de l'équipe Image du GREYC et le reste de l'année à Rennes, au sein de l'équipe MAS d'Orange Labs. Ces deux périodes qui ont rythmé mes années de thèse ont été une véritable chance, m'amenant à rencontrer des personnes exceptionnelles. Ainsi, je tiens à remercier tous mes collègues d'Orange et plus particulièrement Grigory, Corentin, Moez, Emmanuel, Claudia, Olivier, Nicolas, Gaël, Philippe pour toutes les discussions intéressantes et les bons moments que j'ai passés grâce à vous.

Je remercie ensuite Ryan, Shiwang, Julien, Loïc, Luc, Sébastien ainsi que toute l'équipe Image pour les échanges que nous avons pu avoir, tant scientifiques qu'humains. Merci également à Marie Meleux et Sandrine Halvas, qui m'ont beaucoup aidé dans toutes les démarches administratives qui peuvent parsemer la réalisation d'une thèse.

Toute ma gratitude va à ma famille, mon frère et mes parents, qui m'ont toujours soutenu dans mes choix et à qui je dois énormément.

Enfin, Lucie, merci pour tout, pour m'avoir poussé à donner le meilleur de moi-même et pour avoir rendu la vie encore plus lumineuse.

Glossaire

Action Units Activations du visage définies par le Facial Action Coding System (FACS).

AE Auto-Encodeur ou *Auto-Encoder*.

AFEW Acted Facial Expression in the Wild.

agrégation agrégation ou *pooling*.

Animaux Audioset, sous-ensemble des animaux.

AV-MNIST AudioVisual MNIST.

batch lot ou *batch*.

BN Normalisation par groupement ou *Batch Normalization* [134].

C3D Réseau de neurones à convolution 3D.

CNN Réseau de Neurones Convolutif ou *Convolutional Neural Network*.

DAE Auto-Encodeur Débruitant ou *Denoising Auto-Encoder*.

dropout masquage des connections ou *dropout* [260].

ENAS Recherche Efficace d'Architectures Neuronales ou *Efficient Neural Architecture Search*.

EPNAS Recherche Progressive et Efficace d'Architectures Neuronales ou *Efficient Progressive Neural Architecture Search*.

epoch epoch (correspond à une passe complète sur l'ensemble d'entraînement).

FACS Facial Action Coding System.

FER2013 Facial Expression Recognition challenge 2013.

GAN Réseau Génératif Adverse ou *Generative Adversarial Network*.

GPU Processeur Graphique ou *Graphics Processing Unit*.

IMDb Internet Movie Database.

LLL Apprentissage tout au Long de la Vie ou *Life Long Learning*.

LOPO *Leave One Person Out*.

LSTM Réseau de Neurones Récurrent à Mémoire Court et Long Terme ou *Long Short Term Memory*.

MAE Erreur Moyenne Absolue ou *Mean Absolute Error*.

MFAS Recherche d'Architectures de Fusion Multimodale ou *Multimodal Fusion Architecture Search*.

MLP Perceptron Multi-Couches ou *Multi Layer Perceptron*.

mmIMDb Multimodal Internet Movie Database.

MNIST Mixed National Institute of Standards and Technology.

Montalbano Montalbano version 2, RGB+D+audio.

MSE Erreur Moyenne au Carré ou *Mean Squared Error*.

MT Multi-Tâche ou *Multi-Task*.

NTU-RGB+D Nanyang Technological University's Red Blue Green and Depth information.

PCA Analyse en Composantes Principales ou *Principal Components Analysis*.

PNAS Recherche Progressive d'Architectures Neuronales ou *Progressive Neural Architecture Search*.

RAF Real-World Affective Faces.

reLU rectified Linear Unit.

ResNet Réseau de neurones résiduel.

RMSE Racine de l'Erreur Moyenne au Carré ou *Root Mean Squared Error*.

RNN Réseau de Neurones Récurrent ou *Recurrent Neural Network*.

rRMSE Racine de l'Erreur Moyenne au Carré Relative ou *relative Root Mean Squared Error*.

SFEW Static Facial Expressions in the Wild.

SVM Machine à Vecteurs de Support ou *Support Vectors Machine*.

VAE Auto-Encodeur Variationnel ou *Variational Auto-Encoder*.

VGG Réseau de Neurones proposé par le Groupe de Géométrie Visuelle d'Oxford.

Notations

$\mathbf{a}$ Symbole en gras : vecteur.

$\mathbf{A}$ Symbole en gras et majuscule : matrice.

$\mathcal{A}$ Symbole en majuscule et calligraphié : ensemble.

$\#$ Cardinal d'un ensemble.

L Fonction de coût (loss).

θ Ensemble des poids d'un réseau de neurones.

$\mathbf{W}$ Matrice de poids (weights) d'une couche entièrement connectée.

σ Fonction d'activation, non-linéarité.

$\mathbf{x}$ Données en entrée d'un modèle.

$\mathbf{x}, \mathbf{y}$ Données issues de deux modalités et mises en entrée d'un modèle.

$\mathbf{h}$ Couche cachée (hidden) d'un réseau de neurones.

$\hat{\mathbf{z}}$ Approximation de la vérité terrain par un modèle.

$\mathbf{h}$ Annotation, vérité terrain, label.

$\tilde{\mathbf{x}}$ Reconstruction d'une donnée d'entrée par un modèle.

i.e. Id Est / c'est-à-dire.

e.g. Exempla Gratia / par exemple.

etc. Et Cetera / et d'autres choses manquantes.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction générale

Table des matières

1.1 Contexte

Intelligence artificielle et apprentissage automatique Chaque minute, plus de 2.4 millions de recherches sont lancées sur Google tandis que plus de 500 heures de vidéos sont mises en lignes sur Youtube. Twitter compte 350 000 tweets par minute, Instagram plus de 50 millions d'images et Facebook plus de 350 millions de photos. Associés à la mise en ligne de cette impressionnante quantité d'information, s'ajoutent les commentaires et réactions de la part d'autres utilisateurs. Ces éléments constituent des sources de renseignements supplémentaires, avec par exemple plus de 5 milliards de commentaires postés par mois sur Facebook. De manière plus générale, le volume de données numériques augmente constamment, avec des sources très variées, telles que la météorologie, l'astronomie, ou encore les archives de différents programmes et actions (ou logs).

Ces quelques chiffres soulignent la masse actuelle de l'information disponible actuellement sous différentes formes (texte, image, son, chiffres, *etc.*), réceptacle d'un ensemble de connaissances inestimable et difficile à exploiter. Un moyen de disposer de certaines de ces connaissances passe par l'apprentissage statistique, qui consiste à estimer à partir des données un modèle permettant d'inférer automatiquement la réponse à une tâche complexe et difficile à formuler algorithmiquement. Par exemple, la reconnaissance automatique de l'identité d'une personne à partir de son visage est rendue possible par des modèles ayant bénéficié de larges bases de données composées de visages et des identités des utilisateurs associés [130, 248, 75].

Les réseaux de neurones profonds peuvent être considérés comme une sous-branche de l'apprentissage statistique [166] et se révèlent particulièrement efficaces pour traiter ce type de problème [110]. L'entraînement de tels modèles consiste en un problème d'optimisation. Si nous reprenons l'exemple précédent, pour chaque visage, le réseau de neurones proposera un vecteur, auquel il sera possible d'associer une identité. Il sera alors possible de calculer l'erreur effectuée par ce réseau et de la minimiser sur l'ensemble des données, conduisant à l'apprentissage du concept de l'identité. Au fur et à mesure de cet apprentissage, le réseau construit ses propres représentations internes (ou cachées) des données, permettant de développer différentes couches d'abstraction, d'une manière analogue à nos processus d'apprentissage de la perception. La Figure 1.1.1 illustre ce procédé des représentations cachées dans le cadre plus simple de la classification d'images de chiffres, une démonstration en ligne permettant de visualiser les représentations cachées obtenues pour différentes entrées (*e.g.* les représentations cachées d'un 2 seront très différentes de celles obtenues pour un 4).

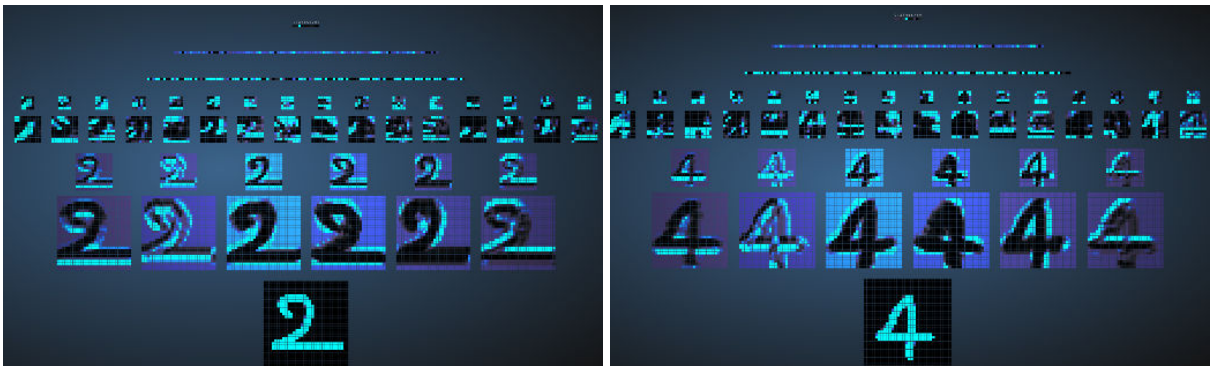


FIGURE 1.1.1 – Visualisations proposées par [120] (démonstration en ligne) des différentes couches de représentations d'un chiffre donné en entrée d'un réseau de neurones profonds, qui a été entraîné à classifier les chiffres entre 0 et 9.

Perception et multimodalité Néanmoins, notre perception repose sur nos sens : nous disposons de plusieurs sources d'information, apportées par différents canaux, entre lesquels nous créons des liens [193] afin de résoudre un problème donné. Prenons l'exemple d'une discussion que nous aurions avec un interlocuteur. Celui-ci cherche à partager avec nous de l'information, qui est donc présente dans les mots

qu'il utilise. Mais l'information va également être communiquée à travers d'autres modalités, telles que le ton de sa voix, ses expressions faciales ou encore ses gestes, constituant une communication non-verbale [192], contenant plus d'information que de simples mots. La perception que nous avons du monde qui nous entoure est donc *multimodale* et elle permet de reconstituer une information riche et complexe à partir de différentes modalités.

Analyse de données multi-modales et application Une grande partie des contenus disponibles sur Internet peuvent être considérés comme des données multimodales. Par exemple, les vidéos sur Youtube contiennent de l'image, du son et des commentaires (donc du texte), tandis que les images de Facebook sont souvent accompagnées d'une localisation ou de textes rédigés par leurs auteurs, et que les tweets contiennent régulièrement des émoticônes et des liens vers des images, du son ou des vidéos.

Pour améliorer la perception des systèmes neuronaux actuels, il est alors intéressant de prendre en compte les différentes modalités disponibles dans les données. Il est par exemple possible d'améliorer la transcription automatique de la parole (*i.e.* transformer la voix en texte) en se servant du mouvement des lèvres [68, 298] pour limiter les erreurs dues par exemple au bruit environnant.

Un sujet particulièrement complexe et où l'usage de différentes modalités peut se révéler très pertinent est l'analyse des émotions d'une personne. En effet, si nous reprenons l'exemple de la communication non-verbale, les émotions sont en grande partie contenues dans des modalités autre que le texte [46] et elles ont alors différentes formes d'expression (vocales, faciales, corporelles). Pour pouvoir percevoir une émotion, nous utilisons l'ensemble de ces modalités et l'apprentissage d'un réseau de neurones utilisant ces différentes modalités permettrait d'améliorer substantiellement la qualité des systèmes unimodaux (*i.e.* n'utilisant qu'une modalité). La Figure 1.1.2 illustre l'intérêt d'une telle démarche : l'expression faciale de cette femme peut aussi bien correspondre à de la joie que de la tristesse ou de la peur, tandis que son expression corporelle et le contexte sonore (sa voix, les applaudissements) permettent de comprendre que Serena Williams célèbre sa victoire.

FIGURE 1.1.2 – Émotion exprimée par Serena Williams à travers différentes modalités.

Améliorer la reconnaissance (et la génération) des émotions constitue un enjeu découlant de plusieurs intérêts et applications :

— l'amélioration des *interfaces homme-machine* en ajoutant une coloration émotionnelle réaliste, par exemple dans le cas des agents virtuels [218] ou en personnalisant l'interface suivant l'humeur de son utilisateur
— *une aide* face au trouble autistique (qui implique l'incapacité à reconnaître les émotions [183]), par exemple en proposant aux personnes autistes un outil permettant d'apprendre à reconnaître et à exprimer des émotions [274]
— la *ré-utilisation* de certaines modalités, telles que l'expression faciale, mais appliquées à des tâches connexes, telles que l'estimation de la douleur [186]
— l'amélioration du marketing client, en identifiant les clients mécontents ou en se servant des émotions exprimées par les clients face à un produit pour évaluer celui-ci

— l'indexation de contenus, pour par exemple retrouver des moments clés au sein de films et ainsi améliorer la qualité des résumés automatiques de ceux-ci [295]

Nous nous intéresserons donc à travers ce manuscrit à l'application des réseaux de neurones dans un contexte multimodal, avec une volonté d'application particulière au domaine de l'analyse des émotions. Cela entraîne diverses problématiques que nous détaillons dans la section suivante.

1.2 Problématiques

Traiter des contenus multimodaux avec des réseaux de neurones profonds implique de se confronter à différentes problématiques :

Fusion entre modalités Un aspect clé de la résolution d'un problème multimodal réside dans la fusion des modalités. Il existe une grande variété de méthodes de fusion multimodale, souvent adaptée pour un problème spécifique. Il est alors très difficile de savoir quelle méthode adopter face à un nouveau problème.

Grande dimensionnalité du signal d'entrée et faible quantité de données annotées le fait de disposer de plusieurs modalités et de chercher à les utiliser signifie que le signal en entrée du réseau de neurones contient beaucoup plus d'informations mais possède donc beaucoup plus de dimensions. Cette dimensionnalité importante rend l'extraction d'une représentation neuronale pertinente plus difficile, surtout dans un contexte comme la reconnaissance d'émotion, où les données disposant d'annotations sont peu nombreuses et souvent bruitées. Il est alors coûteux d'annoter plus de données : nous allons donc explorer des techniques de transfert de connaissances (*i.e.* utilisation de réseaux de neurones déjà entraînés sur d'autres tâches et bases de données) et d'apprentissage faiblement supervisé (*i.e.* en utilisant beaucoup de données dont peu sont annotées).

Interprétabilité Le fait d'utiliser un apprentissage neuronal peut souvent conduire à un effet "boîte noire", *i.e.* à une difficulté à interpréter les décisions produites par le modèle et à analyser ses représentations cachées. Nous chercherons à proposer des solutions permettant d'obtenir des informations sur la qualité de l'apprentissage plus complètes qu'un simple indicateur de performance.

Complexité et sélection des modèles Le fait de traiter des signaux de grandes dimensions et de disposer d'un grand nombre de données implique de disposer d'une bonne capacité d'apprentissage de la part du modèle et donc d'une complexité importante et d'un large nombre de paramètres. Or, cette complexité peut rendre le modèle inapte (en termes de temps de calcul) à être ré-utilisé dans des applications réelles par la suite, voire l'amener à ne pas généraliser au-delà des données vues pendant son entraînement (on parle alors de sur-apprentissage). Nous donnerons de l'importance tout au long du manuscrit à la taille des modèles utilisés et chercherons quand cela nous est possible à réduire celle-ci.

Émotion dans le monde réel La reconnaissance des émotions est un problème résolu dans un cadre contrôlé et en se basant sur des expressions faciales exagérées [185]. En revanche, les techniques qui obtiennent d'excellents résultats dans ce contexte deviennent bien moins efficaces lorsque les expressions sont plus naturelles et dans des conditions plus réalistes (mouvements, orientation du visage, occlusions du visage, *etc.*) [80], ce qui nécessite l'utilisation de nouvelles approches, permettant notamment de construire des représentations plus robustes des émotions.

Ces problématiques nous ont guidés dans nos travaux. Nous reviendrons dans le Chapitre 7 de conclusion générale sur les réponses que nos contributions (détaillées ci-dessous) ont permis d'y apporter.

1.3 Contributions et organisation du manuscrit

État de l'art Le Chapitre 2 introduit des notions et travaux relatifs aux problématiques abordées dans les autres chapitres. Ainsi, nous proposons de présenter les réseaux de neurones profonds et un certain nombre de techniques associées, qui seront utilisées tout au long du manuscrit. Dans le cadre de cette thèse, nous nous sommes particulièrement intéressés à l'analyse des émotions et nous donnons donc également quelques notions d'informatique affective, *i.e.* l'étude et le développement de systèmes permettant de modéliser, reconnaître et générer les émotions humaines ainsi que leur contexte. Enfin, nous revenons sur la multimodalité et sur une taxonomie des approches de fusion multimodale, qui permettra de mieux situer le Chapitre 6. Nous illustrons également les liens forts qui existent entre les techniques de fusion multimodale et celles de résolution d'un problème avec plusieurs tâches, permettant de préparer le Chapitre 5.

Analyse des émotions Nous commençons par traiter une application concrète du problème multimodal dans le contexte de la reconnaissance d'émotion. Notre but est d'aboutir à une solution robuste de reconnaissance d'émotion dans un contexte audiovisuel, en exploitant notamment l'aspect multimodal du problème. Le Chapitre 3 constitue une étude préliminaire et empirique, qui relate nos deux soumissions à une compétition internationale de reconnaissance d'émotion dans des contenus audio-visuels. Nous y étudions notamment l'intérêt d'utiliser des modèles légers et des techniques de transfert de connaissances pour compenser le problème de la grande dimensionnalité des entrées et du peu de données disponibles. Nos participations à la compétition ont permis d'obtenir une évaluation objective de nos solutions et d'ainsi nous comparer de façon directe à l'état de l'art actuel. Ces soumissions nous ont permis d'atteindre à deux reprises une troisième place sur une trentaine d'équipes et ont donné lieu à deux publications [a,b].

De plus, cette étude préliminaire ouvre des pistes de recherche, en illustrant certains problèmes particuliers, tels que le fait qu'une méthode de fusion ne fonctionnera pas sur tous les problèmes multimodaux, que le choix d'un bon transfert de connaissances peut s'avérer très complexe et fastidieux et que le choix d'une bonne représentation de l'émotion est crucial et non trivial pour reconnaître celle-ci.

C'est pourquoi le Chapitre 4 revient sur les représentations des émotions proposées par les psychologues et illustre le lien entre ces représentations et celles apprises par un réseau de neurones dans le cadre de la reconnaissance d'expressions faciales. Cela nous conduit à proposer un modèle de reconnaissance et génération d'expressions faciales basé sur l'utilisation d'une représentation neuronale compacte et efficace. Ces travaux ont été effectués en collaboration avec Corentin Kervadec, qui a effectué une grande partie des expérimentations relatives à la reconnaissance d'expressions faciales lors de son stage de Master. Ils ont donné lieu à deux publications, l'une sur la représentation des expressions faciales [c] et l'autre sur leur génération [d], ainsi qu'à une démonstration en ligne de l'édition d'expression faciale.

Représentation et multimodalité L'importance de la notion de représentation développée dans la Chapitre 4 et la difficulté posée par le choix d'un bon modèle pour le transfert de connaissances lors du Chapitre 3 conduisent naturellement au Chapitre 5, qui propose une méthode non traditionnelle pour effectuer un transfert de connaissances à partir de plusieurs sources, *i.e.* plusieurs modèles pré-entraînés. Pour cela, une représentation neuronale plus générale est obtenue et permet de rassembler la connaissance contenue dans les modèles pré-entraînés en un seul modèle unique, conduisant à des performances à l'état de l'art sur une variété de problèmes d'analyse de visages. Nous avons choisi une application à l'analyse de visage du fait des autres problèmes que nous avons traités jusqu'ici mais la méthode appliquée dans ce chapitre pourra être généralisée à d'autres applications. Ce chapitre est associé à une publication en cours de soumission au moment de la rédaction de ce manuscrit.

Enfin, à travers le Chapitre 6, nous repartons des conclusions du Chapitre 3, et en simulant différents problèmes multimodaux, nous cherchons notamment à vérifier qu'il n'existe pas de technique de fusion multimodale universelle. Nous proposons et validons ensuite deux méthodes pour trouver une architecture neuronale de fusion efficace pour un problème multimodal donné, la première se basant sur

un modèle central de fusion et ayant pour visée de conserver une certaine interprétation de la stratégie de fusion adoptée, tandis que la seconde adapte une méthode de recherche d'architecture neuronale au cas de la fusion, explorant un plus grand nombre de stratégies et atteignant ainsi de meilleures performances. La première méthode a donné lieu à deux publications [e,f], tandis que la seconde a été réalisée en collaboration avec Juan-Manuel Perez, pendant son post-doc au sein de l'équipe, et a donné lieu à une publication [g].

Liste des publications

[a] V. Vielzeuf*, S. Pateux, and F. Jurie. Temporal multimodal fusion for video emotion classification in the wild. *ICMI*, 2017.

[b] V. Vielzeuf*, C. Kervadec, S. Pateux, A. Lechervy, and F. Jurie. An Occam's razor view on learning audiovisual emotion recognition with small training sets. *ICMI*, 2018.

[c] C. Kervadec*, V. Vielzeuf*, S. Pateux, A. Lechervy, and F. Jurie. CAKE : Compact and Accurate K-dimensional representation of Emotion. *BMVC-W*, 2018.

[d] V. Vielzeuf*, C. Kervadec, S. Pateux, and F. Jurie. The many variations of emotion. *FG*, 2019.

[e] V. Vielzeuf*, A. Lechervy, S. Pateux, and F. Jurie. CentralNet : a multilayer approach for multimodal fusion. *ECCV-W*, 2018.

[f] V. Vielzeuf*, A. Lechervy, S. Pateux, and F. Jurie. Multilevel sensor fusion with deep learning. *Sensors*, 2018.

[g] J.-M. Pérez-Rúa*, V. Vielzeuf*, S. Pateux, M. Baccouche, and F. Jurie. MFAS : Multimodal Fusion Architecture Search. *CVPR*, 2019.

Chapitre 2

État de l'art général

Table des matières

Étant donnée la quantité conséquente de travaux publiés dans les domaines traités par le manuscrit, cet état de l'art ne se prétend pas exhaustif et a pour fin première de fournir au lecteur des outils et pointeurs qui pourront lui être nécessaires lors de la lecture du manuscrit.

L'ensemble des chapitres exploite des approches d'apprentissage automatique neuronal et c'est pourquoi nous traitons tout d'abord en Section 2.1 de notions liées aux réseaux de neurones en général (architectures, régularisation, recherche des hyper-paramètres) et aux *représentations* qu'ils permettent d'extraire d'un signal donné, menant à des techniques variées (transfert de connaissances, distillation).

Les Chapitres 3 et 4 s'intéressent plus particulièrement à l'application des réseaux de neurones pour l'analyse des émotions. C'est pourquoi nous présentons en Section 2.2 le domaine de l'informatique affective et l'inscrivons dans le contexte des réseaux de neurones.

Enfin, nous proposons une formulation de ce qu'est un problème dit multimodal (*i.e.* basé sur des données composées de plusieurs modalités, telles que le son ou l'image). Nous reportons par la suite une taxonomie possible des méthodes actuelles de fusion multimodale et des approches multi-tâches. Cette Section 2.3 a pour but de mieux situer les méthodes employées dans les Chapitres 5 et 6.

2.1 Réseaux de neurones et représentations

Cette section s'intéresse à la méthode d'apprentissage particulière que constitue les réseaux de neurones. De manière à bien présenter celle-ci, nous prenons le parti de tout d'abord mentionner quelques éléments généraux relatifs à l'apprentissage statistique. Nous considérons ensuite certaines des composantes principales des réseaux de neurones telles que les architectures, les méthodes de régularisation et la recherche d'hyper-paramètres. Ces premiers éléments permettent ainsi de plus facilement présenter les techniques utilisées dans l'ensemble des chapitres. Enfin, concernant la notion de *représentation*, nous proposons quelques éléments de définition et explorons ensuite diverses applications relatives aux *représentations* neuronales.

2.1.1 Apprentissage statistique

Comme mentionné dans l'introduction générale, les approches neuronales peuvent être vues comme une sous-partie du domaine de l'apprentissage statistique. Celui-ci peut lui-même être défini comme *une discipline de l'intelligence artificielle, qui concerne la conception, l'analyse, le développement et l'implémentation de méthodes permettant à une machine (au sens large) d'évoluer par un processus systématique, et ainsi de remplir des tâches difficiles ou impossibles à remplir par des moyens algorithmiques plus classiques* [1]. Plus sommairement, nous pourrions définir l'apprentissage statistique comme la recherche et l'obtention d'une fonction permettant de résoudre une tâche à partir d'un ensemble de données associées. Plusieurs ouvrages [100, 39, 254, 8] permettent d'approfondir cette définition et d'explorer ce domaine très large.

Formalisation du problème de l'apprentissage supervisé Nous proposons maintenant une définition plus formelle, issue de [254]. Si nous considérons que $\mathcal{X}$ est l'ensemble des entrées possibles et $\mathcal{Z}$ l'ensemble des sorties possibles, il existe une distribution de probabilité inconnue sur l'ensemble $\mathcal{X} \times \mathcal{Z}$, que nous noterons $p(\mathbf{x}, \mathbf{z})$ avec $(\mathbf{x}, \mathbf{z}) \in \mathcal{X} \times \mathcal{Z}$.

Nous pouvons définir le concept d'apprentissage comme la recherche d'une fonction $f : \mathcal{X} \to \mathcal{Z}$ telle que $f(\mathbf{x}) \sim \mathbf{z}$. Par la suite, nous pourrons noter $\hat{\mathbf{z}} = f(\mathbf{x})$ et $L(\hat{\mathbf{z}}, \mathbf{z})$ une fonction de coût permettant d'évaluer l'écart entre valeur estimée $\hat{\mathbf{z}}$ par f et valeur réelle $\mathbf{z}$.

Cela permet alors de définir un risque attendu R tel que :

$$R(f) = \int_{\mathcal{X} \times \mathcal{Z}} L(\hat{\mathbf{z}}, \mathbf{z}) p(\mathbf{x}, \mathbf{z}) \mathbf{dxdz} \tag{2.1.1}$$

1. https://fr.wikipedia.org/wiki/Apprentissage_automatique

Le but est alors de trouver f^* telle que $R(f^*)$ soit minimal. Mais il se trouve que la distribution de probabilité $p(\mathbf{x}, \mathbf{z})$ est inconnue. Il est alors nécessaire de définir un sous-espace de $\mathcal{X} \times \mathcal{Z}$, que nous appellerons ensemble d'entraînement $\mathcal{D}$ composé de N éléments issus de $p(\mathbf{x}, \mathbf{z})$. Ainsi dans cet ensemble, à chaque $\mathbf{x_i}$ est associé une sortie $\mathbf{z_i}$, formant N couples. Cet ensemble $\mathcal{D}$ permet alors d'obtenir une approximation du risque attendu R, que nous appellerons le risque empirique $R_{\mathcal{D}}$:

$$R_{\mathcal{D}}(f) = \frac{1}{N} \sum_{i}^{N} L(\hat{\mathbf{z}}_\mathbf{i}, \mathbf{z_i}) \tag{2.1.2}$$

Notion de modèle Par la suite, nous utiliserons le terme de modèle pour désigner les fonctions f et considérerons que les différentes fonctions f peuvent être exprimées par des paramètres θ. Ainsi, nous considérons qu'un modèle M est composé de paramètres θ et permet de réaliser une inférence, prenant en entrée un vecteur $\mathbf{x}$ et estimant une sortie $\hat{\mathbf{z}}$:

$$\hat{\mathbf{z}} = M(\theta, \mathbf{x}) \tag{2.1.3}$$

La minimisation de la fonction de coût présentée précédemment s'écrira alors :

$$\theta^\star = argmin_\theta \frac{1}{N} \sum_{i}^{N} L(\mathbf{z_i}, \hat{\mathbf{z}}_\mathbf{i}) \tag{2.1.4}$$

Apprentissage non supervisé Nous nous sommes précédemment placés dans le cas où nous disposions d'éléments de l'espace $\mathcal{Z}$, permettant de modéliser une supervision, *i.e.* que nous avons considéré que nous disposions d'*annotations* ou *vérité terrain*, par exemple produites par un être humain.

Il est aussi possible de se placer dans un cas où on ne dispose pas de cette supervision et où l'on cherche alors souvent à estimer, de manière explicite ou implicite, la distribution de probabilités $p(\mathbf{x})$. Nous parlerons alors d'apprentissage non supervisé, qui regroupe différentes méthodes et catégories spécifiques [254].

Classification, régression Au sein de l'apprentissage supervisé, nous pouvons distinguer différentes problématiques et notamment des problèmes de régression et de classification.

La classification consiste en des annotations par catégories, *i.e.* que les z_i correspondent à des valeurs discrètes et qu'il en existe un nombre fini. Par exemple, un problème de classification classique serait de déterminer à partir d'une photo d'animal s'il s'agit d'un chien, d'un chat ou d'un oiseau.

La régression consiste en des annotations par valeurs continues, *i.e.* que les z_i correspondent à des valeurs comprises dans un intervalle donné. Par exemple, un problème de régression pourrait être l'estimation de la position d'un animal dans une image, modélisée par deux valeurs continues à régresser . Les méthodes utilisées et notamment la fonction de coût employée présentent des différences entre ces deux types de problématiques, comme nous le verrons par la suite.

Composantes de l'apprentissage statistique De manière générale, nous allons considérer qu'un modèle d'apprentissage statistique dépend de différents éléments :

— les paramètres (ou poids) θ, *i.e.* des valeurs numériques que l'on cherche et qui permettent de définir M et de réaliser l'inférence

— une fonction de coût L permettant d'évaluer l'inférence, *i.e.* la qualité de l'estimation de l'annotation par le modèle

— une méthode d'optimisation permettant de minimiser la fonction de coût et de mettre à jour les poids du modèle

— des hyper-paramètres permettant de contrôler l'ensemble, *e.g.* la vitesse de convergence de la méthode d'optimisation

Nous nous concentrons dans cette thèse sur le cas spécifique des réseaux de neurones profonds [110]. Ceux-ci sont des modèles d'apprentissages statistiques particuliers, qui présentent notamment une structuration (ou architecture) des valeurs numériques θ du modèle en différentes couches, permettant de construire automatiquement et couche par couche une représentation du signal d'entrée de plus en plus appropriée pour la résolution de la tâche donnée.

2.1.2 Architectures de réseaux de neurones

Nous considérons donc que le réseau de neurones (et plus particulièrement les réseaux de neurones profonds) peut être défini comme un ensemble de poids muni d'une structure particulière, optimisés pour traiter un signal d'entrée de manière à résoudre une tâche donnée. Nous allons voir dans cette partie que la structure de cette famille de poids dépend de l'architecture du réseau de neurones. Nous décrivons l'élément d'architecture à la base d'un réseau de neurones, le perceptron, puis poursuivons par la description de différentes familles d'architectures plus sophistiquées et plus récentes.

Perceptron Une première formulation datant de 1958 est celle du perceptron [239], proposée au départ dans le cas de la classification binaire. La Figure 2.1.1 illustre la structure de cet élément d'architecture. L'entrée x est ici un vecteur de dimension n. Chacun des éléments est multiplié par un poids w_i, puis sommé (*i.e.* un produit scalaire est effectué entre l'entrée $\mathbf{x}$ et les poids). Une fonction d'activation σ (*e.g.* une fonction échelon qui associe 0 à son entrée h si $h < 0.5$ et 1 sinon) est alors appliquée au résultat pour obtenir une valeur z permettant de décider à quelle classe (0 ou 1) appartient $\mathbf{x}$. De manière plus formelle, il est possible de ré-écrire cette opération sous la forme :

$$\hat{\mathbf{z}} = \sigma\left(\begin{pmatrix} w_1 & w_2 & \dots & w_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}\right) \tag{2.1.5}$$

Cette formulation [2] peut être étendue à une sortie de taille supérieure à 1, en réécrivant simplement :

$$\hat{\mathbf{z}} = \sigma(\mathbf{W}x) \tag{2.1.6}$$

avec $\hat{z}$ la sortie, vecteur de taille m, $\mathbf{x}$ l'entrée, vecteur de taille n, σ la fonction d'activation (qui est appliquée indépendamment sur chaque sortie) et $\mathbf{W}$ une matrice de taille $m \times n$. Le perceptron ainsi défini correspond à la solution neuronale typique pour résoudre un problème de classification (ou de régression) linéaire simple.

Perceptron Multi-Couches ou *Multi Layer Perceptron* (MLP) Le perceptron ne permet pas en revanche de résoudre certains types de problèmes, se révélant inefficace par exemple dans le cas de la classification non-linéaire. La Figure 2.1.2 illustre un ensemble de données avec des annotations non linéairement séparables et qui nécessite donc d'exhiber des frontières non linéaires. Une manière de résoudre ce problème est de projeter de manière non linéaire les données dans un espace où elles seront linéairement séparables, en exhibant par exemple plus de dimensions que dans l'espace précédent (astuce du noyau [4]). Ce type d'approche est très utilisée dans d'autres approches d'apprentissage statistique telles que les Machine à Vecteurs de Support ou *Support Vectors Machine* (SVM) [70].

L'idée du MLP [243] est alors, comme le nom l'indique, de rassembler plusieurs couches de perceptrons, séparées par des non-linéarités, de manière à créer un espace de représentation des données d'entrée qui soit linéairement séparable. Par exemple, pour un MLP à une couche cachée, l'entrée x est traitée par un premier perceptron et donne une sortie intermédiaire, dite cachée, $\mathbf{h}$. Celle-ci est injectée dans un second perceptron, qui permet d'obtenir une sortie finale $\mathbf{z}$.

2. Notons que pour simplifier la notation, nous n'utilisons pas de terme additif de biais (qui pourrait d'ailleurs être simplement pris en compte en considérant que quelque soit x, $x_n = 1$).

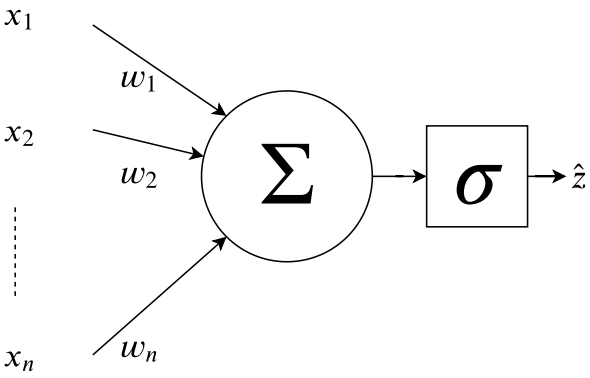


FIGURE 2.1.1 – Représentation schématique du perceptron défini en 1958. L'entrée **x** est composée de n éléments x_i, qui sont chacun multipliés par un w_i, avant d'être tous sommés, puis traités par une fonction d'activation σ.

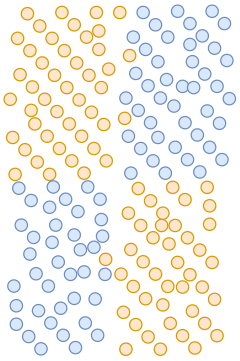


FIGURE 2.1.2 – Exemple de problème non linéairement séparable (impossible de séparer les deux classes avec une seule droite). Les cercles représentent les données **x**, leur couleur correspondant à leur annotation **z**.

La fonction d'activation (*e.g.* tangente hyperbolique, sigmoïde, rectified Linear Unit (reLU) [202]), utilisée au sein des couches cachées, permet d'introduire la non-linéarité et d'ainsi faciliter la résolution de problèmes non linéairement séparables.

Réseau de Neurones Convolutif ou *Convolutional Neural Network* (CNN) De manière à pouvoir par exemple traiter des entrées de taille plus large, des modèles de réseau de neurones ré-introduisant la notion de filtre ont fait leur apparition. Les images sont des exemples d'entrées avec un grand nombre d'éléments, nécessitant donc un nombre de paramètres très conséquents pour les traiter avec un MLP. Fukushima [101] puis Le Cun *et al.* [167] proposent une architecture qui prend en compte la topologie spatiale de l'image et permet notamment grâce à cet a priori de réduire le nombre de paramètres nécessaires.

Un CNN peut être composé de 3 types de couches : (1) une couche de convolution, qui consiste en un ensemble de noyaux de convolution appliqués à l'image pour en extraire des *cartes de convolution*, (2) une couche d'agrégation, qui consiste à sous-échantillonner les cartes préalablement obtenues, par exemple en prenant la moyenne ou le maximum d'un ensemble de valeurs, (3) une couche totalement connectée, *i.e.* un perceptron avec une activation non-linéaire. Un exemple classique d'architecture convolutive proposée par Le Cun *et al.* pour la reconnaissance automatique de chiffres dans les images [167] est *LeNet-5*. Il est constitué d'une succession de couches des trois types, comme illustré par la Figure 2.1.3.

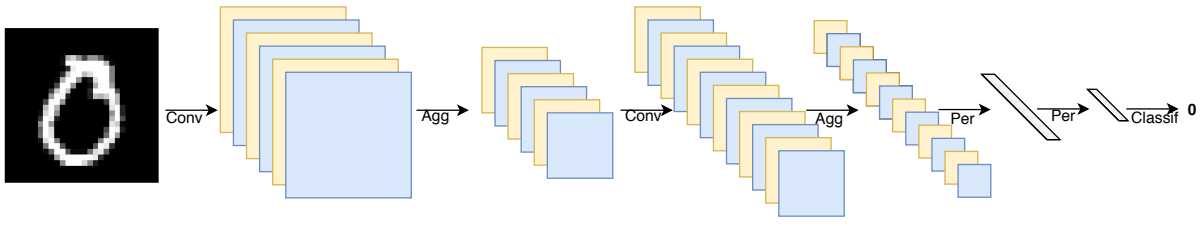


FIGURE 2.1.3 – Représentation schématique d'un réseau convolutif similaire au *LeNet-5* [167]. Les couches convolutives, d'agrégation et totalement connectée sont respectivement désignées par les acronymes Conv, Agg et Per. La dernière couche est un perceptron simple qui permet la classification finale de l'image parmi 10 classes.

Le réseau de neurones convolutif présente plusieurs caractéristiques qui expliquent son succès sur les problèmes relatifs aux images. Nous pouvons notamment mentionner :

1. la connectivité locale : un élément de sortie d'une couche de convolution est lié à des éléments d'entrée qui sont proches spatialement, du fait de la taille limitée du noyau de convolution.
2. le partage des poids des noyaux de convolution, ceux-ci parcourant l'ensemble de l'image.
3. l'invariance à la translation, par nature même de la convolution.

Par la suite, des ensembles de données plus larges [74] émergent et permettent des architectures plus "profondes", *i.e.* avec plus de couches. Parmi celles-ci, nous pouvons noter par exemple AlexNet [161], Réseau de Neurones proposé par le Groupe de Géométrie Visuelle d'Oxford (VGG) [253] et Réseau de neurones résiduel (ResNet) [122].

Le réseau de neurones convolutif peut être généralisé à des entrées de dimensions supérieures à deux, pour traiter par exemple des séquences d'images (qui contiennent donc une dimension temporelle) [22, 271]. Nous illustrons l'intérêt de cette approche en Figure 2.1.4 et nous verrons l'application d'une telle architecture au problème de la reconnaissance d'émotions à partir d'une séquence d'images dans le Chapitre 3.

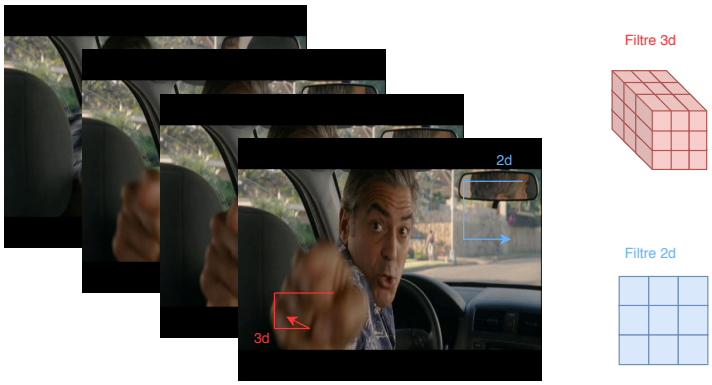


FIGURE 2.1.4 – Couches convolutives 2d et 3d appliquées à une séquence d'images. La convolution 3d permet d'extraire une unique description de toute la séquence et donc de prendre en compte la notion de proximité temporelle (ou connectivité locale). La convolution 2d traite chaque image séparément.

Il est également intéressant de noter l'utilisation du réseau de neurones convolutif à une dimension pour traiter du son [83, 273] et des séries temporelles [42].

Enfin, nous pouvons mentionner l'émergence des éléments spécifiques d'architecture. Szegedy *et al.* [264] proposent un module d'Inception, qui consiste à appliquer en parallèle plusieurs blocs convolutifs différents (par exemple avec des noyaux de tailles différentes) et à concaténer leurs sorties. He *et al.* [122] proposent ResNet, qui permet d'améliorer la convergence. C'est une approche qui est dite résiduelle car elle consiste à sommer (ou concaténer), à la sortie d'un bloc, l'entrée de celui-ci (résidu). Autrement dit, si une couche classique s'écrit $\mathbf{h} = \mathbf{W}\mathbf{x}$, nous pouvons définir la couche résiduelle par $\mathbf{h} = \mathbf{W}\mathbf{x} + \mathbf{x}$, *i.e.* il s'agit de la combinaison d'un bloc et de la fonction identité. Han *et al.* [117] améliorent cette idée en travaillant sur la couche d'agrégation et en proposant une approche pyramidale (PyramidNet). Un nombre croissant d'autres approches telles que les DenseNets [129] ou les Squeeze-Excitation Networks [126] sont également validées de manière empirique. Une étude récente [147] dresse une liste (non exhaustive) de ces approches.

Cette large variété d'architectures et de techniques justifiées par l'intuition et l'expérimentation ramène naturellement la communauté en direction d'une interprétation plus théorique pour justifier les choix d'architectures. Ainsi, nous pouvons noter l'émergence de plusieurs interprétations (pouvant s'opposer) basées sur la théorie de l'information, discutant par exemple l'influence de la compression de l'information sur la capacité de généralisation d'un réseau de neurones [269, 246]. Un aspect plus traditionnel, comme proposé dans le cours de Stéphane Mallat [190], est de percevoir l'effet des différents blocs comme une reparamétrisation d'espaces très complexes et non linéairement séparables en espaces plus facilement manipulables. Cette vision n'est pas sans évoquer l'astuce du noyau, qui consiste à reparamétrer les données dans un espace de plus grande dimension où elles sont linéairement séparables.

Enfin, des travaux comme ceux d'Escorcia *et al.* [95] s'intéressent à la localisation de l'information "utile" au sein d'un réseau de neurones et montrent par exemple que les attributs visuels permettant de résoudre une tâche donnée sont distribués dans l'ensemble des couches d'un réseau de neurones convolutif.

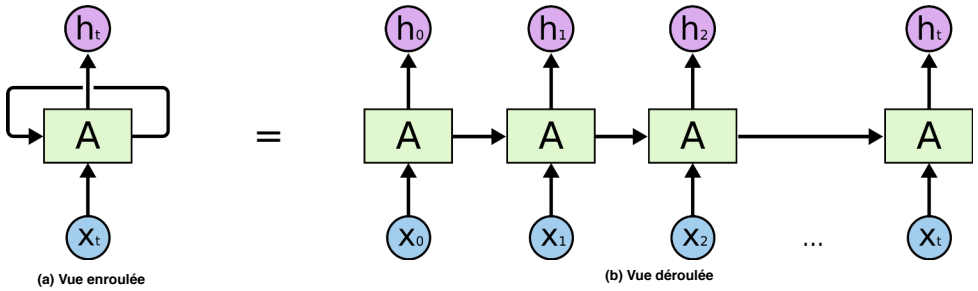


FIGURE 2.1.5 – Illustration d'un réseau de neurones récurrent. *A* est une cellule, $\mathbf{x_t}$ un élément de la séquence d'entrée et $\mathbf{h_t}$ la sortie associée. Schéma réalisé à partir du site https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Réseau de Neurones Récurrent ou *Recurrent Neural Network* (RNN) Une autre forme d'architecture connaissant beaucoup de succès est le RNN. Cette forme provient d'un besoin de traiter des séquences d'éléments, par exemple des textes (dont les éléments sont des caractères ou des mots) et qui peuvent avoir une longueur variable. La Figure 2.1.5 présente l'architecture classique d'un RNN. L'entrée de ce réseau, nommée $\mathbf{x}$, est une séquence de T éléments (notés $\mathbf{x_t}$). Le réseau peut être vu de deux manières : enroulée et déroulée. Concentrons-nous d'abord sur la partie déroulée. Le principe est d'associer à chaque élément de la séquence une cellule. Cette cellule prend en entrée l'élément $\mathbf{x_t}$ et un état $\mathbf{s_{t-1}}$, provenant de la cellule précédente. Elle permet alors d'obtenir une sortie $\mathbf{h_t}$ et un état $\mathbf{s_t}$; qui sera transmis à la cellule suivante. La cellule classique est souvent composée d'un réseau de neurones de type MLP qui prend la concaténation de $\mathbf{x_t}$ et $\mathbf{s_{t-1}}$ en entrée et donne une unique sortie $\mathbf{h_t}$, également utilisée comme état $\mathbf{s_t}$ par la cellule suivante. Il y a donc transmission de l'information tout au long de la séquence, ce qui permet d'inclure les observations passées dans le processus de décision de l'instant *i*. La vision enroulée du RNN trouve sa justification dans le fait que les poids du réseau de neurones permettant d'obtenir les sorties sont partagés entre les cellules.

Des variantes de cellules existent, telles que les Réseau de Neurones Récurrent à Mémoire Court et Long Terme ou *Long Short Term Memory* (LSTM) [113] et les Gated Recurrent Linear Units (GRU) [65], qui permettent d'oublier certaines informations contenues dans l'état *s* et d'ainsi éviter que l'information contenue dans celui-ci ne disparaisse lorsqu'on traite des séquences très longues [3]. Suivant les types d'entrées traitées (séquences textuelles, audio, d'images, autres), des techniques et cellules spécifiques ont été développées comme le décrivent Lipton *et al.* [176] en 2015. À cette période, Karpathy fournit également une vision des diverses applications possibles d'un réseau de neurones récurrent à travers son blog : http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

Dans le Chapitre 3, nous présenterons également une LSTM traitant les sorties d'un CNN, pour effectuer de la reconnaissance d'expression faciale dans des séquences d'images. Nous confirmerons la tendance récente (exemple des meilleures approches du challenge Youtube-8M [196] en analyse de vidéos) à éviter cette approche pour un certain nombre de problèmes pour lesquels les données sont trop peu nombreuses et/ou la notion d'ordre est peu présente.

Auto-encodeur et modèles génératifs Les architectures présentées précédemment cherchent à résoudre une tâche souvent proche de la sémantique (classification ou régression par exemple). L'idée

3. Il s'agit concrètement d'une "disparition du gradient" progressive pendant la phase de mise à jour des poids. Pour des détails généraux sur l'apprentissage d'un réseaux de neurones, voir la partie 2.1.3

de l'AE, développée très tôt [24], est de proposer d'apprendre une représentation compressée du signal d'entrée. Pour cela, la première partie est un réseau de neurones classique, par exemple un MLP ou un CNN, compressant l'information contenue dans l'entrée (par exemple une image comme dans la Figure 2.1.6) dans une couche cachée h. À partir de celle-ci, le décodeur cherche à reconstituer le signal d'origine. Il y a compression car la couche cachée $\mathbf{h}$ est de dimension beaucoup plus faible que le signal d'entrée, ce qui implique que seules les informations les plus significatives sont conservées. De nombreuses variantes, comme les Auto-Encodeur Débruitant ou *Denoising Auto-Encoder* (DAE) [282] ou les Auto-Encodeur Variationnel ou *Variational Auto-Encoder* (VAE) [156] existent et permettent par exemple de la génération de contenu. Ces méthodes seront reprises dans le Chapitre 5 où nous utilisons un AE pour regrouper les représentations issues de plusieurs modèles en une seule représentation plus compacte et plus générale.

Enfin, il est important de noter parmi les modèles génératifs la technique de l'approche adverse ou Réseau Génératif Adverse ou *Generative Adversarial Network* (GAN) [111]. L'idée principale est d'entraîner simultanément deux réseaux : un *générateur* et un *discriminateur*. L'objectif du *générateur* est de créer des contenus synthétiques les plus réalistes possibles, tandis que le *discriminateur* apprend à distinguer ces contenus factices des contenus réels. L'entraînement simultané permet d'amener le *générateur* à améliorer de plus en plus la qualité des contenus générés, pendant que le *discriminateur* doit trouver des détails de plus en plus fins pour identifier les contenus factices. La compétition qui a lieu entre ces deux réseaux adversaires joue un rôle essentiel dans l'entraînement, l'un ne pouvant être entraîné sans l'autre. Le *générateur* finalement obtenu permettra d'échantillonner des contenus et pourra être utilisé dans diverses applications telles que l'augmentation des données ou la modification de contenus (super-résolution, inpainting, *etc.*). Le *discriminateur* permettra d'obtenir une représentation particulièrement robuste d'un contenu et pourra notamment être affiné pour résoudre de nouvelles tâches relatives à ce type de contenu.

La convergence d'un GAN peut se révéler particulièrement difficile et un grand nombre de techniques ont été proposées pour faciliter celle-ci, telle qu'une reformulation de la fonction de coût [19] ou un entraînement progressif [145].

Plus de détails sur ce sujet vaste et complexe peuvent être trouvés dans des ouvrages tels que le tutoriel de Goodfellow [109]. Le Chapitre 4 exploite notamment une méthode de génération adverse pour modifier les expressions du visage.

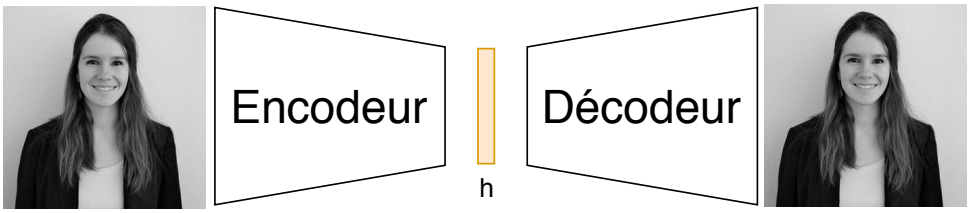


FIGURE 2.1.6 – Représentation schématique du principe d'un AE. L'encodeur permet de représenter le signal d'entrée en une représentation cachée $\mathbf{h}$. Le décodeur doit alors reconstruire à partir de cette représentation l'image d'origine.

2.1.3 Apprentissage d'un réseau de neurones

Nous avons jusqu'ici détaillé la phase d'inférence d'un réseau de neurones, essentiellement définie par son architecture et pouvant être modélisée comme une fonctionnelle f telle que :

$$\hat{\mathbf{z}} = f(\mathbf{W}, \mathbf{x}) \tag{2.1.7}$$

Ainsi, cette fonctionnelle utilise un ensemble de poids $\mathbf{W}$ pour calculer une sortie estimée $\hat{\mathbf{z}}$ à partir d'une entrée $\mathbf{x}$. Mais pour pouvoir résoudre une tâche donnée, il est alors nécessaire d'entraîner notre modèle

sur un ensemble de N données $\mathcal{D} = (x_i)_{(i\in 1..N)}$, de manière à estimer des valeurs optimales des poids $\mathbf{W}$ suivant une fonction de coût L.

Concernant l'optimisation, $\hat{\mathbf{z}}$ va être injectée dans une fonction de coût L, qui permet d'estimer quelle est la qualité du modèle sur la tâche choisie par rapport à une annotation $\mathbf{z}$. L'objectif va ensuite être de minimiser cette fonction de coût sur l'ensemble des données d'entraînement $\mathcal{D}$, *i.e.* minimiser le risque empirique (*cf.* introduction).

Fonction de coût De manière générale, il est possible d'écrire que L est une fonction ayant pour entrées f, $\mathbf{W}$, $\mathbf{z}$ et $\mathbf{x}$. En effet, $L(f,\mathbf{W},\mathbf{x},\mathbf{z}) = erreur(f(\mathbf{W},\mathbf{x}),\mathbf{z})$, avec *erreur* une fonction adaptée au problème à résoudre. Par exemple, pour les problèmes de régression et de classification, il est usuel d'utiliser respectivement la moyenne des erreurs quadratiques ($= \sum_i (z_i - \hat{z}_i)^2$) et l'entropie croisée ($= -\sum_i \hat{z}_i log(z_i)$).

Rétro-propagation du gradient Le fait de minimiser la fonction de coût correspond finalement à trouver une valeur optimale des poids du réseau. Il est important de noter que $L(f,\mathbf{W},\mathbf{x},\mathbf{z})$ est une fonction non-convexe, ne permettant donc pas de garantir l'unicité d'un optimum. Ainsi, l'existence d'un optimum n'est pas garantie et le réseau de neurones peut conduire à exhiber des optima locaux. Nous verrons tout au long du manuscrit qu'il est important d'utiliser des aides à la convergence, permettant notamment de ne pas stagner au sein d'un optimum local.

Concrètement, nous appliquons des méthodes de mise à jour des poids de type descente de gradient. Il s'agit d'une méthode usuelle, consistant à mettre à jour chacun des poids dans la direction opposée au gradient de la fonction de coût. Plus formellement, pour chacun des poids $w \in \mathbf{W}$, la différence Δw entre l'ancienne et la nouvelle valeur de w s'écrit :

$$\Delta w = -\alpha \frac{\partial L}{\partial w} \tag{2.1.8}$$

α correspond ici à un *taux d'apprentissage* permettant de contrôler la vitesse de mise à jour des poids. Pour pouvoir effectuer le calcul de la dérivée partielle $\frac{\partial L}{\partial w}$, une propagation inverse est effectuée. En effet, il est possible de calculer cette dérivée partielle séquentiellement, en partant des poids de la couche de sortie (calcul immédiat) puis en propageant le calcul couche par couche en direction de l'entrée. Il s'agit d'une application du théorème de dérivation des fonctions composées ou *chain rule*.

Optimisation stochastique Dans la pratique, la mise à jour des poids peut être effectuée de plusieurs manières. La *descente de gradient groupée* consiste à faire la mise à jour des poids à partir du gradient moyen sur tout l'ensemble d'entraînement, impliquant un coût important en mémoire et le risque de converger dans un point-selle [102]. La *descente de gradient stochastique par lot ou batch (batch) unique* consiste à traiter les données de l'ensemble d'entraînement une par une et à effectuer itérativement la mise à jour. Puisque la direction de mise à jour choisie à chaque itération est basée sur un unique exemple, le procédé d'optimisation est alors très bruité. Dans le cas des réseaux de neurones profonds (et plus précisément de l'optimisation non-convexe), ce bruit peut être bénéfique pour sortir d'un minimum local ou d'un point selle [43, 102]. Mais le temps de convergence qu'implique cette solution est très important. C'est pourquoi l'approche la plus utilisée dans le cas des réseaux de neurones profonds est la *descente de gradient stochastique par batch*, un compromis entre approche groupée et approche stochastique à batch unique. Elle consiste à faire la mise à jour des poids à partir du gradient moyen calculé sur des petits batch de données. Cela permet de conserver les bénéfices d'une optimisation bruitée/stochastique, mais de tout de même disposer d'une convergence plus rapide.

Différentes améliorations de la descente de gradient stochastique peuvent être utilisées pour accélérer et stabiliser la direction de convergence (Momentum [228], gradient accéléré de Nesterov [205], RMSprop, Adagrad [87], Adadelta [310], Adam [155], *etc.*).

Des approches du second ordre, comme la méthode de Newton, restent peu utilisées du fait de la taille importante des ensembles de données à traiter et du bruit important sur l'estimation de la dérivée seconde.

2.1.4 Aide à la convergence

La nature non-convexe du problème d'optimisation a poussé la communauté à développer des méthodes permettant d'aider à la convergence, à la fois en termes de stabilité, de vitesse et de qualité de l'optimum trouvé. Cette notion est également intrinsèquement liée à celle de généralisation, *i.e.* de trouver un optimum suffisamment global pour être toujours valable quand le modèle est appliqué sur des données n'ayant pas été vues durant l'entrainement.

Statistique du signal Ainsi il existe notamment des méthodes visant à préserver la statistique du signal (moyenne et variance par exemple) au cours de l'inférence. La normalisation des entrées ou une initialisation normalisée des poids [108] permettent de commencer à partir d'un point potentiellement plus proche des optima locaux et d'ainsi éviter une divergence dès le départ. Il est également possible de contraindre l'apprentissage pour assurer la préservation de la statistique du signal, avec des méthodes telles que la Normalisation par groupement ou *Batch Normalization* [134] (BN) [134], la normalisation groupée [292], *etc.* . Ces approches permettent d'accroître la stabilité de la convergence en évitant par exemple des valeurs extrêmes, ce qui permet donc également d'en améliorer la vitesse.

Régularisation Un autre aspect important est la capacité de généralisation du modèle obtenu. Du fait du grand nombre de paramètres contenus dans un modèle, celui-ci est capable de "sur-apprendre" l'ensemble d'entraînement. Lorsque le modèle est confronté à un ensemble de test, avec des entrées non vues auparavant, sa performance va être extrêmement dégradée. Pour éviter ce phénomène, une approche classique de l'apprentissage statistique consiste en une régularisation de l'apprentissage [265].

Cette régularisation peut prendre la forme d'une modification de la fonction de coût, avec l'ajout d'une contrainte. Par exemple, il est possible d'ajouter la norme L1 ou L2 des poids à la fonction de coût, ce qui a pour effet de réduire leurs valeurs. Or un modèle avec des petits poids générera des variations de plus faibles amplitudes, impliquant une meilleure stabilité et une meilleure généralisation [162].

Généralisation Un autre point important est d'obtenir une bonne généralisation, *i.e.* d'obtenir un modèle conservant sa performance sur d'autres données que celles utilisées lors de l'entraînement. Une technique intuitive consiste alors à agrandir l'ensemble d'apprentissage. La collecte et l'annotation de données étant une tâche lourde, une possibilité est de les augmenter artificiellement. Par exemple, plusieurs versions d'une même image peuvent être générées : en modifiant sa taille, sa couleur, en la tournant, *etc.* . Le but de ces transformations est de mieux couvrir l'espace des possibles pour une même annotation et d'ainsi aider le réseau à généraliser. Une autre interprétation de ces techniques d'augmentation est qu'il s'agit d'un ajout de bruit sur les entrées lors de l'apprentissage, de manière à obtenir du réseau de neurones des réponses invariantes à ce bruit. De cette idée, dérive un grand nombre de techniques consistant à bruiter les entrées (comme le cutout [77]), mais aussi les couches cachées (dropout [260], blockout [200], shakeout [141]).

Fonctions d'activations Le choix des fonctions d'activations permet également d'améliorer la convergence. Un grand nombre de fonctions existe. Au départ, les architectures ont souvent été présentées avec des sigmoïdes ou des tangentes hyperboliques. Il se trouve qu'utiliser des fonctions telles que la reLU permet d'accélérer la convergence. En effet, un neurone ayant une activation forte aura tendance à propager une dérivée nulle avec la sigmoïde ou la tangente hyperbolique, alors qu'elle sera constante avec une reLU. Ce type de neurone aura donc tendance à être plus difficile à modifier pour la sigmoide ou la tangente hyperbolique.

Enfin, principalement pour pallier le fait que la reLU a une tendance importante à créer des neurones "morts" (*i.e.* nuls, du fait de son annulation pour des valeurs négatives), d'autres activations ont été proposées telles que la leaky reLU [189] ou la PReLU [121].

2.1.5 Recherche des paramètres

Un réseau de neurones peut donc être vu comme un modèle complexe, qui dépend d'un grand nombre d'hyper-paramètres : architecture (type, nombre de neurones, nombre de couches cachées, activation, agencement, *etc*.), fonction de coût, régularisation, algorithmes d'optimisation, taux d'apprentissage, taille des groupements... Il est donc nécessaire de fixer ces éléments de la manière la plus adaptée au problème traité.

La majorité des approches s'est souvent contentée d'effectuer cela manuellement, en se basant sur l'a priori des expériences passées. Pourtant, nous sommes face à une tâche qui pourrait être automatisée. En effet, il est possible de reformuler cette recherche d'hyper-paramètres en une tâche d'optimisation vers une performance maximale du réseau de neurones. Nous décrivons ici différentes méthodes de manière générique, mais il est important de noter que l'optimisation d'hyper-paramètres se concentre souvent sur un aspect en particulier : recherche d'une architecture, d'une méthode d'augmentation des données, d'un ensemble d'hyper-paramètres haut niveau (taux d'apprentissage, pondération de la régularisation, *etc*.), d'une méthode d'optimisation.

Recherche par exploration Cette technique se base uniquement sur de l'exploration des différentes configurations d'hyper-paramètres. Notons que générer des combinaisons de valeurs aléatoires semble apporter de meilleurs résultats [37] que d'explorer une grille de tous les paramètres. L'intuition derrière cette constatation s'illustre dans la Figure 2.1.7, où la méthode par grille échoue à trouver une combinaison optimale de paramètres.

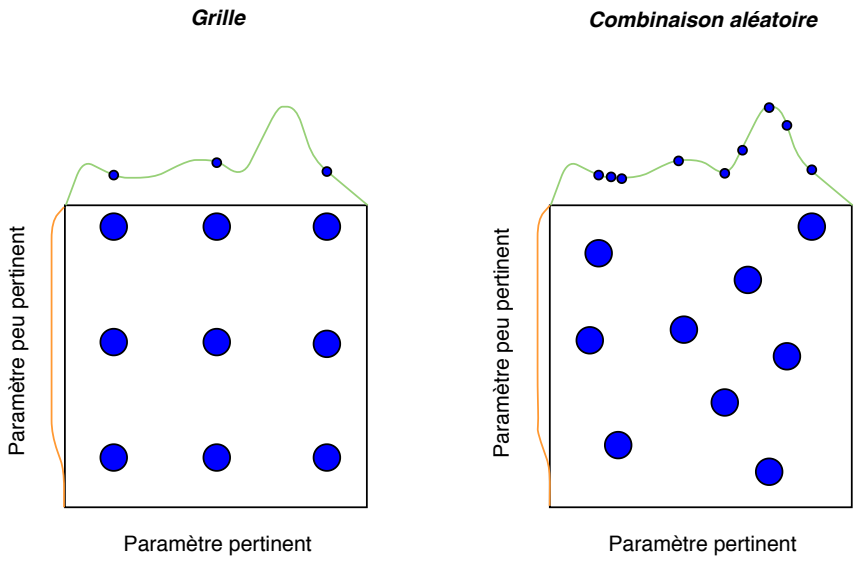


FIGURE 2.1.7 – Représentation des deux types de recherche par grille (classique et aléatoire) pour deux paramètres, l'un pertinent et changeant la performance sur la tâche donnée, l'autre n'ayant que peu d'influence. La recherche par grille n'explore que trois valeurs du paramètre pertinent, tandis que la recherche aléatoire permet d'obtenir une bonne couverture de l'ensemble des valeurs (schéma inspiré de [37]).

Recherche par algorithmes génétiques Pour ne pas se contenter d'une exploration, une méthode proposée dès 2002 [261] est l'utilisation d'un algorithme génétique pour trouver une configuration d'hyper-paramètres optimale. Pour cela, une "population" de configurations est choisie, mutée, et évaluée sur un ensemble de validation. Seules les meilleures configurations donnant les meilleurs résultats sont conservées et ré-injectées dans une nouvelle itération [293].

Recherche par apprentissage par renforcement L'apprentissage par renforcement se prête également parfaitement à cette situation, permettant de définir un agent proposant différentes configurations [329, 326, 331, 71]. Celles-ci sont évaluées sur un ensemble de validation et permettent ensuite de récompenser et d'améliorer l'agent pour ses prochaines propositions, mêlant exploitation et exploration.

Autres approches D'autres approches se concentrent sur l'apprentissage effectif d'une fonction permettant d'estimer la performance d'une configuration donnée. Pour cela, des techniques d'optimisation bayésienne [256] ou séquentielle [178] sont souvent appliquées.

Enfin, une autre façon de formuler le problème est le méta-apprentissage [281], qui consiste à considérer qu'on "entraîne un méta-modèle à entraîner un modèle". Pour cela, l'ensemble de données devient un méta-ensemble, divisé en différents ensembles. Sur chacun d'eux, le méta-modèle doit choisir la configuration optimale pour l'entraînement du modèle. Ainsi, le méta-modèle est entraîné à prédire la meilleure configuration pour chacun des ensembles et permet d'entraîner efficacement un nouveau modèle sur un nouvel ensemble. Cette approche a souvent pour but de permettre un apprentissage avec peu d'exemples ou face à des problématiques telles que l'apprentissage en "un coup" [234] (*i.e.* une seule passe sur l'ensemble d'entraînement) ou en "zéro coup" (*i.e.* être capable de prédire des labels jamais vus auparavant en se servant de connaissances a priori, ici contenues dans le méta-modèle).

2.1.6 Transfert de connaissances

Nous avons précédemment formulé le problème de l'entraînement d'un réseau de neurones dans l'optique de résoudre une tâche donnée. Mais il se trouve qu'il existe d'autres intérêts dans l'entraînement d'un modèle. En effet, une vue possible d'un réseau de neurones entraîné pour résoudre une tâche est celle d'un réservoir de connaissances. Lorsque nous sommes confrontés à une nouvelle tâche, il est possible et parfois pertinent de ré-utiliser la connaissance contenue dans le premier modèle pour en entraîner un nouveau. Une formulation alors utilisée est celle du transfert de connaissances. Ce transfert peut être utile dans de nombreux cas, par exemple lorsque la première tâche comprend beaucoup d'exemples et est très complexe, il est probable qu'une partie de la connaissance apprise par le réseau de neurones sera utile lors de l'apprentissage de la nouvelle tâche.

Notion de représentation Une définition importante pour traiter de transfert de connaissances est la notion de représentation extraite par un réseau de neurones. Cela revient à considérer que les couches cachées d'un réseau de neurones sont autant de représentations du signal d'entrée, à différents niveaux de sémantique [311, 95]. Ainsi, chaque couche va contenir des informations de plus en plus sémantiques en s'éloignant de l'entrée [199]. Et va donc souvent être de plus en plus spécialisée. Néanmoins, les couches cachées relativement bas-niveau d'un réseau de neurones entraîné sur un très grand nombre d'exemples constituent souvent une représentation très générale et complète du signal d'entrée.

Un exemple plus concret est la visualisation des couches apprises par un réseau de neurones convolutif sur un large corpus, par exemple ImageNet [74], qui permettent de modéliser toute la complexité des entrées à travers une large banque de filtres. Cela explique la tendance à ré-utiliser les parties bas-niveau de réseaux appris sur de large corpus pour extraire des représentations plus stables du signal d'entrée.

Approche classique Pour formaliser le transfert de connaissances, écrivons qu'un modèle M_a est appris sur une tâche t_a associée à un ensemble de données $\mathcal{D}_a$. On désigne par M_a^i le modèle sans les couches situées au-dessus de la couche i.

Pour une entrée donnée $\mathbf{x}$, il existe alors n couches cachées $\mathbf{h_i} = M_a^i(\mathbf{x})$. Ces couches cachées peuvent être considérées comme des représentations du signal $\mathbf{x}$. Nous cherchons maintenant à résoudre une tâche t_b associée à un ensemble de données $\mathcal{D}_b$ qui contient des données de nature similaire à celles du premier ensemble $\mathcal{A}$ (*e.g.* $\mathcal{A}$ et $\mathcal{B}$ contiennent des images).

Pour cela, une première possibilité est d'extraire pour tout $\mathbf{x}$ appartenant à $\mathcal{D}_b$ une représentation $\mathbf{h_i} = M_a^i(\mathbf{x})$. À partir de celles-ci, il est alors possible d'entraîner un nouveau modèle (souvent de petite taille) sur la tâche t_b. La "hauteur sémantique" optimale i à laquelle la représentation doit être extraite peut être sujet à discussion et a souvent été choisie empiriquement, comme le montrent certains papiers de compétitions récentes [99].

Une autre possibilité, qui apporte souvent de meilleures performances, est d'ajuster M_a sur $\mathcal{D}_b$. Pour cela, une approche classique [33] consiste à construire un réseau M_b composé d'un M_a^i pré-entraîné et avec i bien choisi, et de le compléter par des couches initialisées aléatoirement. Dans un premier temps, seules ces dernières couches sont entraînées sur la nouvelle tâche et le nouveau domaine. Dans un second temps, l'erreur est également propagée dans tout M_b avec un taux d'apprentissage plus faible, ce qui permet d'affiner les paramètres de M_a^i.

Tâches Comme mentionné précédemment, pour effectuer un transfert pertinent, il est intéressant d'utiliser des tâches présentant certains liens avec la tâche cible. Pour identifier ces liens, Zamir *et al.* [309] proposent une "Taskonomy". Celle-ci permet de sélectionner la meilleure combinaison de modèles existants (*i.e.* la combinaison des représentations extraites par ces modèles) pour résoudre une nouvelle tâche. Notons également que Ying *et al.* [306] proposent une approche similaire mais avec des transferts du premier ordre uniquement (*i.e.* transfert d'une tâche vers une autre). Pour transférer des connaissances issues de plusieurs tâches, Geyer*et al.* [105] fusionnent des modèles pré-entraînés en utilisant des méthodes des moments incrémentales.

Jusqu'ici nous avons supposé que les ensembles $\mathcal{D}_a$ et $\mathcal{D}_b$ étaient annotés respectivement pour les tâches t_a et t_b. Mais il est également possible de transférer des connaissances depuis un ensemble de données non annotées. Pour cela, l'idée est d'effectuer un apprentissage auto-supervisé, en sélectionnant des tâches "gratuites" en termes d'annotation et pertinentes pour résoudre ensuite t_b. Ces *tâches intermédiaires*, pour qu'elles ne soient pas coûteuses, exploitent souvent la nature des données et sont corrélées à la tâche t_b. Un grand nombre de tâches intermédiaires appliquées à des images peuvent être citées [318, 319, 214, 215, 106, 231], l'idée étant de dégrader l'image, puis d'entraîner M_a à annuler cette dégradation (*e.g.* re-colorisation, estimation de la rotation de l'image, reconstitution d'une image découpée en plusieurs morceaux).

D'autres tâches dites "non supervisées" peuvent être considérées comme entrant dans cette catégorie des tâches intermédiaires, telles que l'approche de Caron *et al.* [53], qui consiste à entraîner un réseau de neurones à prédire les groupements obtenus par un algorithme d'agrégation (ou *clustering*).

Multi-tâches Les approches décrites précédemment sont souvent appliquées dans un contexte proche de la notion de multi-tâches. L'intuition de l'approche multi-tâches réside dans l'idée qu'il est intéressant d'apprendre une représentation permettant de répondre à plusieurs tâches, car celle-ci présentera une meilleure capacité de généralisation [55]. Ainsi ce type d'approche est utilisée avec succès dans beaucoup de domaines, tels que le traitement naturel du langage [69], la reconnaissance de la voix [76], la détection d'objet [107], la détection et l'alignement de visages [315], ou même la recherche de médicaments [233]. Doersch *et al.* [85] ont également appliqué l'idée des tâches intermédiaires développée au paragraphe précédent dans un contexte multi-tâches, tout comme les auteurs de la "Taskonomy" [309]. Nous détaillerons en fin de Chapitre (section 2.3) quelques méthodologies utilisées en multi-tâches.

Domaines Un autre élément d'importance dans le transfert de connaissances est la généralisation à travers différents domaines. Cela signifie que les ensembles $\mathcal{D}_a$ et $\mathcal{D}_b$ sont différents et définissent des statistiques d'observation spécifiques, pouvant entraîner des difficultés de généralisation lorsqu'on change de domaine ou lorsqu'on souhaite étendre celui-ci (*e.g.* un ensemble de photos de visages prises en intérieur et un autre ensemble prises en extérieur), tandis que les tâches t_a et t_b sont identiques (*e.g.* reconnaissance d'expression faciale). Dans cette situation, il est possible de directement appliquer M_a sur $\mathcal{D}_b$. Mais la performance risque d'être dégradée. Ainsi des méthodes spécifiques à ce problème existent, consistant par exemple à combiner plusieurs réseaux spécialisés sur différents domaines [191, 63].

Distillation Enfin, nous avons présenté la technique du transfert de connaissances, en ré-utilisant et affinant un modèle (ou des modèles) entraîné(s) sur une tâche liée. Mais il est aussi possible de procéder par distillation [123]. L'idée est d'utiliser deux modèles : un maître et un élève. Le maître est souvent un modèle complexe avec un grand nombre de paramètres préalablement entraînés. L'élève cherche alors à prédire les sorties ou même les représentations extraites [237] par le maître pour chaque entrée d'un ensemble de données, qui n'a donc pas besoin d'être annoté. Ce type d'approche permet non seulement d'obtenir un modèle élève beaucoup plus léger, mais également d'améliorer la performance. D'autres approches proposent également d'utiliser plusieurs maîtres entraînés différemment [58, 232, 173] et observent une amélioration, ce qui rejoint les résultats observés dans les paragraphes précédents en combinant plusieurs tâches intermédiaires. Enfin, l'intérêt d'un tel procédé est qu'il permet de transférer des connaissances sur des problématiques contenant des données de natures différentes [296]. Par exemple, le modèle SoundNet [21] permet de classifier des sons et a été entraîné à partir d'une annotation visuelle d'un grand nombre de contenus audiovisuels.

2.2 Informatique affective

L'informatique affective [222] consiste à étudier et développer des systèmes permettant de modéliser, reconnaître et générer les émotions humaines ainsi que leur contexte. Il s'agit donc d'une discipline à la croisée de plusieurs domaines : robotique sociale, interaction homme-machine, psychologie, animation et cinéma, informatique. Dans cette partie, nous proposons d'en présenter quelques aspects clés, qui permettront de faciliter la lecture des Chapitres 3 et 4. Nous donnons donc quelques définitions et modèles de représentation essentiels, puis détaillons diverses méthodes de reconnaissance et génération des émotions.

2.2.1 Définitions

Avant de chercher à reconnaître ou générer une émotion, il est essentiel de savoir la définir et la représenter. Nous présentons donc certains des concepts utilisés par la communauté à des fins de modélisations théoriques et pratiques.

Affect, humeur, émotion Les notions d'affect, d'humeur et d'émotion [32] constituent un vocabulaire de base. L'affect est défini par la communauté en psychologie par "un état neurophysiologique consciemment accessible en tant qu'un simple sentiment primitif non réflexif". De manière plus concrète, l'énergie, la relaxation, ou la fatigue sont des exemples d'affect. Il s'agit d'états qu'une personne va connaître de manière constante et qui ne proviennent pas forcément d'un stimulus extérieur mais peuvent aussi être expliqués par sa personnalité. La notion d'émotion, quant à elle, est définie de manière prototypique [88] comme un "épisode complexe de sous-événements liés autour d'un même objet ou stimulus (personne, événement, chose réelle ou imaginaire)". Ainsi, il s'agit d'une réaction sur une période courte face à une situation ou un stimulus donné. Enfin, l'humeur peut être vue comme une émotion sur une période plus longue [32] et dont la cause n'est pas immédiate (*e.g.* une personne peut être d'humeur triste du fait d'une situation vécue la veille).

Dans nos travaux, nous nous concentrons sur l'émotion notamment au travers de sa perception visuelle et auditive, mais il reste important de noter que celle-ci fait partie d'un ensemble plus vaste et ne permet pas en elle seule la compréhension de l'état affectif d'une personne.

Expressions Pour pouvoir percevoir une émotion, il faut partir du postulat que celle-ci est exprimée [72]. Une formulation simplifiée du problème de la reconnaissance d'émotions serait celle représentée en Figure 2.2.1. L'émotion est un signal interne (en réaction à un stimulus), qui va provoquer diverses manifestations externes dites expressions. Celles-ci peuvent être gestuelles, vocales, ou encore faciales. C'est en utilisant ces diverses expressions qu'il est possible de reconnaître l'état interne de la personne.

Quant à la tâche de génération d'émotion, le principe revient finalement à générer des expressions pouvant correspondre à un état interne donné.

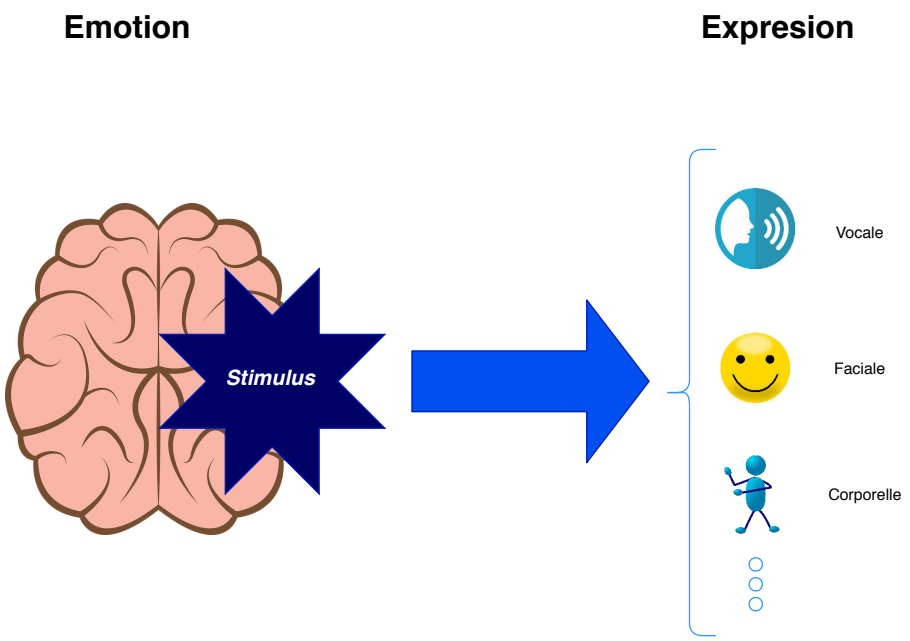


FIGURE 2.2.1 – Représentation de l'expression d'une émotion. Celle-ci est un signal interne provoqué par un stimulus et restitué sous forme de différentes modalités d'expression.

Représentation des émotions De manière à pouvoir qualifier et quantifier l'émotion ressentie et ainsi apprendre à la reconnaître et la générer, il est d'abord nécessaire de la représenter de manière cohérente. C'est pourquoi il s'agit d'un sujet très exploré par la communauté, qui propose différentes méthodes. Une approche basique proposée par Ekman *et al.* [88] est celle des émotions discrètes. Elle consiste à proposer des catégories dites "universelles" (dans le sens où elles sont communes à toutes les cultures), telles que la peur, la surprise, la colère, la joie, et le dégoût. Bien qu'étant facile à comprendre et à utiliser, ce système présente l'inconvénient d'être ambigu, puisqu'une émotion peut très bien être un mélange de joie et de surprise par exemple. C'est pour cela que des émotions mélangées [86] (*e.g.* joyeusement surpris) ont été proposées. Néanmoins, pour décrire toute la complexité d'une émotion, il faudrait alors employer une infinité de mots. Une solution à ce problème, proposée par Russell *et al.* [244] est alors de considérer l'émotion comme un espace continu. Pour contrôler cet espace, ils proposent d'utiliser des directions telles que le niveau de plaisir (valence), d'excitation (arousal) ou même de confiance en soi [195] (dominance). Annoter et interpréter les espaces ainsi créés devient moins intuitif et facile que dans le cas des émotions discrètes, mais apporte une meilleure précision et qualité d'annotation. Enfin, dans le cas spécifique de l'expression faciale, une représentation beaucoup plus objective propose un encodage du visage par Activations du visage définies par le FACS (Action Units) [90], permettant d'identifier certains éléments clés relatifs à l'émotion (*e.g.* sourcils levés, bouche ouverte, *etc.*).

2.2.2 Reconnaissance d'émotions

La reconnaissance automatique d'émotion est un sujet qui a été abordé depuis plusieurs dizaines d'années et qui peut prendre plusieurs formes. Nous nous intéressons ici aux méthodes récentes utilisées pour les différents types de modalités de l'émotion.

Expression faciale Le visage est un excellent moyen d'expression de l'émotion. De plus, il existe un savoir-faire sur ce type de données dans le domaine de l'apprentissage statistique. La création d'un

grand nombre de corpus, dès les années 2000, est donc logique. Des ensembles de données en conditions contrôlées (*e.g.* visages frontaux, expressions non spontanées, illumination identique) avec peu de diversité ethnique sont d'abord proposés [188, 185, 114, 325]. Puis grâce à l'explosion de la quantité de données disponibles sur Internet et au début de l'apprentissage profond, des ensembles de grande tailles et en conditions non contrôlées voient le jour. Par exemple, FER-2013 [112] contient environ 30 000 visages annotés en émotion discrète, tout comme RAF [172] qui est annoté avec une meilleure qualité (30 annotateurs par image environ). Ensuite, EmotioNet [98] rassemble un million de visages annotés avec des Action Units et AffectNet [198] propose également un million de visages annotés à la fois en émotion discrète et en plaisir-excitation.

La diversité et la largeur des ensembles de données a donc permis de développer de nouvelles techniques d'apprentissage. Ainsi les auteurs de la base de données AffectNet [198] utilisent trois AlexNet [161] pour apprendre respectivement les émotions discrètes, l'excitation et le plaisir. De plus, le transfert de connaissances entre bases de données est beaucoup utilisé [209]. Enfin, des techniques plus sophistiquées voient le jour pour prendre en compte par exemple les importantes variations au sein d'une même classe d'émotion discrète. Ainsi, une tendance importante est d'essayer rapprocher au maximum dans l'espace de représentation les éléments d'une même classe, en intégrant par exemple cette contrainte dans la fonction de coût [172, 47]. Acharya *et al.* [2] implémentent une *agrégation par covariance*, ce qui permet d'utiliser des statistiques d'ordre deux lors de la fusion des différentes cartes de convolutions, et d'ainsi explorer des relations plus complexes entre ces cartes.

Plusieurs travaux s'intéressent également aux bénéfices apportés par l'étude des liens entre les diverses représentations. Ainsi, Khorrami *et al.* [148] montrent qu'un réseau de neurones entraîné avec des visages sur une tâche de reconnaissance d'émotions discrètes apprend implicitement dans ses couches cachées des Action Units. Confirmant cette première analyse, Pons *et al.* [226] entraînent un modèle à satisfaire deux tâches : la prédiction d'émotion discrète et d'Action Units. En revanche, peu de bénéfices semblent provenir de l'entraînement d'un modèle pour satisfaire l'estimation d'émotion discrète et de plaisir-excitation. Ces deux représentations sont pourtant très liées, les émotions discrètes pouvant être projetées dans l'espace de l'excitation-plaisir [198], comme l'illustre par exemple la Figure 2.2.2.

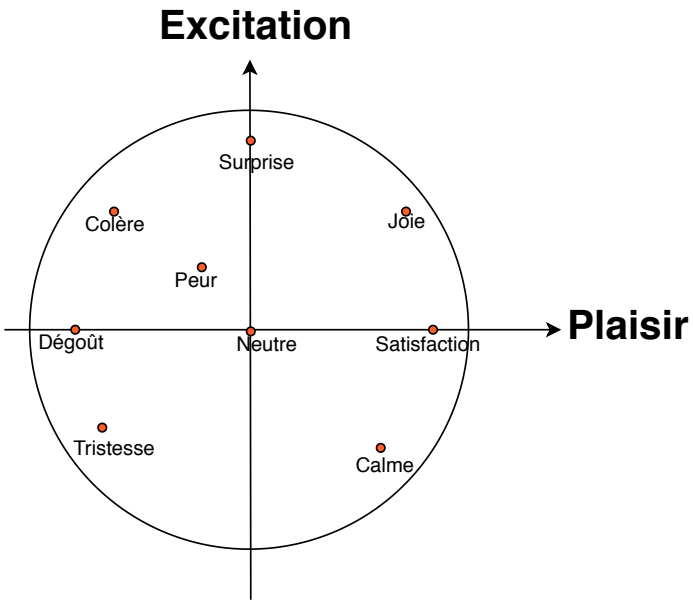


FIGURE 2.2.2 – Exemple de projection de classes d'émotion discrète dans l'espace excitation-plaisir. Il est important de noter que cette projection correspond à un barycentre des éléments de cette classe, différentes expressions appartenant à une même classe occupant différentes positions dans l'espace. Pour réaliser cette projection, nous avons utilisé un sous-ensemble de la base de données AffectNet [198], qui sera étudiée plus en détail dans le Chapitre 4

Enfin, des ensembles de données présentant non plus des images mais des séquences d'images ont également émergé, permettant de travailler sur les notions de temporalité, à la fois avec des annotations globales d'émotion discrète [82] et avec des annotations continues en excitation-plaisir [194, 236, 160].

Expression vocale Les corpus disponibles en expression vocale sont plus limités en taille que dans le cas de l'expression faciale [91, 249]. Cela explique également la prépondérance d'approches par extraction de descripteurs manuels [97]. Néanmoins, des approches par distillation multimodale (en se servant de modèles visuels pour entraîner des modèles audio) ont permis d'utiliser des corpus plus larges [6].

Expressions gestuelles et autres D'autres modalités peuvent être utilisées pour exprimer une émotion. Ainsi des études recensent l'apport de diverses modalités, telle que l'expression corporelle [157, 213]. Le problème de ce type de modalité réside dans la faible taille des ensembles de données disponibles, rendant difficile des approches neuronales profondes.

Multimodal Comme présenté dans les paragraphes précédents, il existe plusieurs modalités d'expression de l'émotion, ce qui en fait naturellement un problème multimodal. C'est pourquoi de plus en plus de travaux cherchent à adresser la reconnaissance d'émotions avec des approches de fusion multimodale. Ainsi, un grand nombre de techniques provenant de la littérature de fusion sont appliquées à ces problématiques, souvent pour combiner les prédictions finales d'un réseau dédié à une entrée visuelle et d'un autre dédié à une entrée audio [78, 277]. De nombreuses méthodes de fusion existent et seront détaillées dans la partie de ce chapitre dédiée aux problèmes multimodaux. Un aspect difficile auquel sont confrontées les approches multimodales est la faible taille des ensembles de données disponibles [236, 79, 139, 323], combinée à la dimensionnalité conséquente des entrées à traiter (*e.g.* une vidéo est beaucoup plus lourde à manipuler qu'une image).

2.2.3 Génération d'émotion

La génération d'émotion est un autre aspect important pour la communauté de l'informatique affective. Celle-ci permet d'une part de mieux comprendre les mécanismes inhérents à la reconnaissance des émotions, mais aussi de simuler des comportements humains.

Expression faciale L'expression faciale est la modalité la plus explorée également pour le problème de génération. La communauté de la modélisation 3D (*e.g.* pour des applications cinématographiques) s'est intéressée très tôt à l'animation d'un visage à partir d'un modèle 3D [40, 297]. Plus récemment, Soladié *et al.* [257] utilisent 4 dimensions pour animer un visage en utilisant également des descripteurs issus d'un modèle à apparence active. Une méthode plus générale est également proposée par Weber *et al.* [288], consistant à entraîner un modèle de manière non supervisée et spécifique à une personne et à s'adapter ensuite à une nouvelle personne cible à la fois. Enfin, des approches hybrides entre apprentissage profond et modèle spécifique à la personne existent. Par exemple, Susskind *et al.* [263] a construit un réseau de neurones profonds basé sur les Action Units et sur l'information relative à l'identité pour générer des expressions faciales. Plus récemment, une autre approche [258] utilise des points fiduciels pour contrôler l'animation du visage, tandis que Tulyakov *et al.* [276] apprennent directement à partir d'une séquence d'images en séparant "contenu" et "dynamique".

D'autres travaux proposent d'améliorer les transitions entre diverses émotions en utilisant des points d'intérêt du visage [229]. Enfin, Kim *et al.* [152] génèrent une animation séquentielle du visage en se basant sur l'exemple d'une autre vidéo de visage.

Toutes ces approches travaillent sur la forme même du visage et permettent de contrôler finement les changements imposés à l'expression. En revanche, il est plus difficile de s'adapter à des conditions non contrôlées sans des modifications complexes. C'est pourquoi des approches entièrement neuronales ont également été développées. Ainsi, Choi *et al.* [66] améliorent la robustesse de la génération en apprenant sur plusieurs domaines et tâches à la fois (*i.e.* attributs du visage et expression faciale). Ding *et al.* [84]

proposent également une nouvelle méthode permettant de contrôler l'intensité des expressions générées, ce qui permet de couvrir une plus large variété d'expressions faciales. Enfin, une autre approche développée par Pumarola *et al*. [227] fait usage d'Action Units pour entraîner un réseau génératif autorisant des modifications très précises et contrôlées de l'expression faciale.

Expression vocale La génération d'expression vocale de l'émotion est une tâche difficile. En effet, les grandes avancées en synthèse automatique de la voix [279, 285] sont récentes. Néanmoins, l'idée du contrôle de l'émotion vocale n'est pas étrangère à la communauté et plusieurs méthodes de modification des générateurs existants sont proposées [124, 270], mais souvent à partir d'ensembles de données de très faibles tailles (inférieurs à dix exemples), rendant toute généralisation difficle.

Multimodal Bien que l'idée soit déjà présente [212], la génération automatique de contenus multimodaux émotionnels reste un domaine peu exploré. En revanche, il existe des travaux sur la génération de contenus multimodaux [60, 16] de manière plus générale. Et ces approches utilisent souvent des modèles génératifs inspirés de techniques existantes en fusion multimodale [283].

2.3 Multimodalité et multi-tâche

Cette partie est consacrée au domaine de la multimodalité dans un contexte d'apprentissage profond. Nous donnons dans un premier temps des exemples de problèmes multimodaux et discutons quelques définitions. Nous nous concentrons ensuite sur l'aspect spécifique de la construction d'une architecture neuronale multimodale. Enfin, une dernière sous-partie présente quelques approches multi-tâches et s'intéresse au lien entre multimodalité et multi-tâches.

2.3.1 Problème multimodal

Un problème multimodal consiste en un problème qu'il est possible de résoudre en utilisant plusieurs modalités. Il est alors nécessaire de définir ce qu'est une modalité. Mais cette définition peut présenter une certaine ambiguïté. En effet, partons du principe que notre problème multimodal consiste en l'analyse d'un signal caché (*e.g*. l'émotion décrite dans la partie précédente). Pour pouvoir effectuer cette analyse, différentes vues ou modalités du signal sont utilisées (*e.g*. image, son). Mais il est également commun de considérer le flot optique, la profondeur ou des éléments de gestes comme autant de modalités [207]. Or les gestes ou le flot optique sont issus des images. Pourtant, il s'agit de deux vues différentes du signal, ne donnant pas accès à la même information. De la même manière, il est tout à fait possible d'utiliser plusieurs réseaux de neurones entraînés sur des tâches différentes pour extraire des représentations du signal d'origine. Ces représentations sont-elles alors également des modalités ?

La littérature récente [25] étudie des modalités qui restent souvent liées au langage (*e.g*. texte), à la vision (*e.g*. image) et à la voix (*e.g*. son, prosodie). C'est pourquoi nous considérerons essentiellement dans le Chapitre 6 des problématiques multimodales relatives à ces modalités et non des vues abstraites. Mais nous verrons également dans le Chapitre 5 que les techniques présentées peuvent tout de même être appliquées dans des contextes différents, en manipulant par exemple des représentations issues de réseaux de neurones.

Un problème multimodal peut alors être formulé à partir de ces modalités et va être constitué de différents objectifs [25] :

— **Représentation** ou comment exploiter au mieux les relations entre les modalités pour extraire une représentation pertinente du signal.

— **Traduction** ou comment traduire une modalité en une autre (*e.g*. comment décrire une image avec du texte et inversement).

— **Alignement** ou comment faire correspondre deux modalités (*e.g*. retrouver les différentes étapes écrites d'une recette de cuisine avec celles montrées dans une vidéo).

— **Fusion** ou comment tirer parti de plusieurs modalités pour résoudre un même problème (*e.g.* utiliser le son et l'image pour reconnaître l'émotion, car celle-ci peut être présente dans les deux modalités).

— **Co-apprentissage** ou comment transférer de la connaissance entre plusieurs modalités (*e.g.* apprendre la classification de son à partir d'un modèle visuel déjà entraîné [21]).

Ces objectifs sont présents dans diverses applications, telles que la reconnaissance et la synthèse audiovisuelles de la parole, de l'émotion [81, 235, 128], la détection et la classification d'événements, d'actions, de contenus multimédias [252, 1, 286, 149], la génération automatique de résumés [96], ou encore la réponse à des questions visuelles [14, 34] (*i.e.* répondre à des questions à propos d'une image donnée).

Nous faisons ici le choix de présenter essentiellement des problématiques de fusion et de représentation dans des contextes de classification, celles-ci étant au coeur de nos travaux. Concernant les autres objectifs et applications, la très complète taxonomie proposée par Baltruvsaitis *et al.* [25] permettra de trouver plus de références.

2.3.2 Fusion multimodale

Il existe un grand nombre de méthodes de fusion multimodale, qui peuvent être regroupées de différentes manières.

Fusion par projection Une première taxonomie qui a été très utilisée [20, 11, 164] est de séparer les méthodes par le niveau de représentation auquel les modalités sont fusionnées : l'idée est d'opposer les méthodes de fusion précoce (fusion proche de l'entrée) aux méthodes de fusion tardive (fusion proche de la sortie) comme l'illustre la Figure 2.3.1. Cela présuppose qu'il existe un opérateur de fusion, tel que la somme [204], la concaténation [252] ou le produit bilinéaire [34] des représentations. Cet opérateur de fusion permet alors de projeter deux représentations "unimodales" dans le même espace multimodal.

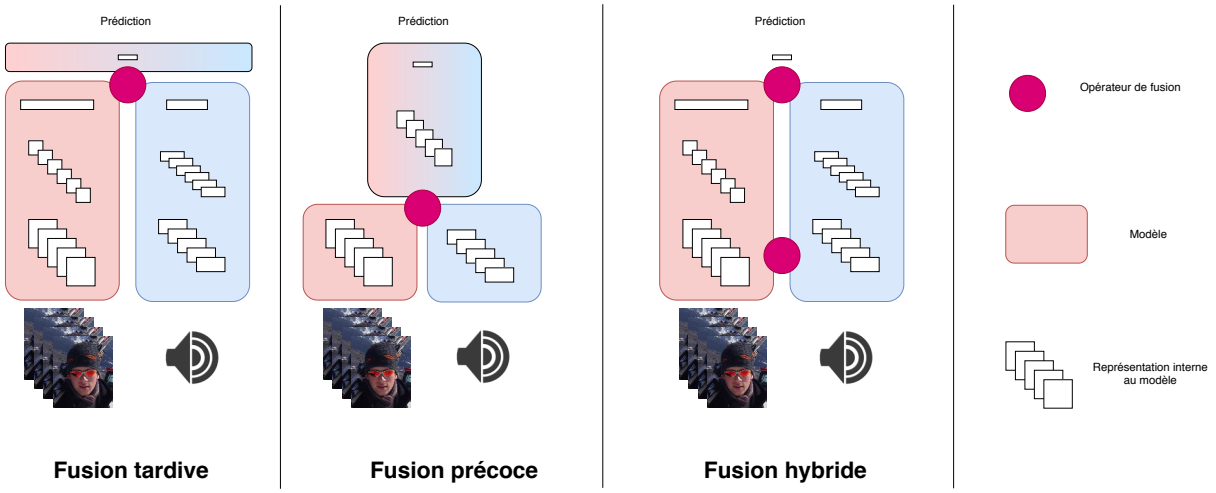


FIGURE 2.3.1 – Exemple d'une taxonomie d'architectures de fusion multimodale, opposant fusion précoce à fusion tardive. Les méthodes combinant plusieurs niveaux de représentations tombent alors dans la catégorie de la fusion hybride. L'opérateur de fusion peut prendre plusieurs formes : concaténation, somme/produit, produit bilinéaire, MLP, AE. Notons que ces méthodes peuvent être étendues à n modalités.

Il n'y a pas de consensus sur le niveau idéal auquel appliquer l'opérateur de fusion, puisque celui-ci semble dépendre de la tâche et des données à traiter. En effet, même si la fusion tardive semble en général donner des résultats légèrement meilleurs [255], il arrive qu'elle soit largement dépassée par les approches précoces. Par exemple, Simonyan *et al.* proposent un réseau convolutif bimodal [252] pour

la reconnaissance d'activité humaine en fusionnant tardivement. De manière similaire, les approches donnant les meilleurs résultats en reconnaissance d'émotions [151] et en détection d'événements [204] sont des fusions tardives. De même, des méthodes d'apprentissage à noyaux multiples [23, 291, 304] sont très utilisées et s'appliquent souvent à proximité des sorties.

Pourtant, d'autres auteurs proposent de fusionner l'information à bas niveau. Zhou *et al.* [327] concatènent des représentations à bas niveau et utilisent une Analyse Multi-Discriminante. De même, Arevalo *et al.* [18] ou Chen *et al.* [61] introduisent des méthodes de fusion beaucoup plus proches des entrées, en utilisant de plus des systèmes de portes permettant de favoriser certaines modalités. Cette idée de portes trouve ses origines dans une méthode plus générale de "mélange d'experts" proposée par Jacobs *et al.* [136]. L'idée est d'entraîner un modèle annexe à sélectionner à partir d'une information contextuelle l'expert le plus approprié parmi plusieurs afin de traiter une portion d'entrée donnée. Cette notion se retrouve également dans les mécanismes d'attention, qui peuvent être appliqués à des approches multimodales et temporelles [125, 182].

Néanmoins, effectuer une taxonomie en se basant uniquement sur le niveau de fusion donne une vision limitée du domaine. En effet, une grande partie des méthodes n'entre pas dans les catégories de fusion tardive ni précoce et elles sont alors regroupées sous le terme de fusion hybride. Une bonne illustration est la méthode heuristique de fusion par étape proposée par Neverova *et al.* [207], cherchant à fusionner d'abord les modalités très corrélées (carte de profondeur et image) à bas niveau et ensuite les modalités moins corrélées (audio et vision) à plus haut niveau. D'autres méthodes de fusion à plusieurs niveaux existent. Notamment, Yang *et al.* [299] proposent de sélectionner plusieurs niveaux de fusion par une méthode de boosting [230]. Cătălina Cangea *et al.* [48] développent une fusion multi-niveau avec des représentations de dimensionnalités différentes. Enfin, des approches multi-niveaux introduisent également des éléments de régularisation [115, 142, 137, 9].

Fusion par contraintes Nous avons montré que la fusion par projection conduit à construire une unique représentation multimodale. Mais il est aussi possible de simplement imposer des contraintes sur les représentations unimodales (Baltruvsaitis *et al.* [25] utilisent le terme de représentations coordonnées). L'Analyse Canonique des Corrélations ou *Canonical Correlation Analysis* (CCA) en est un premier exemple [119] non neuronal cherchant à maximiser la corrélation entre deux représentations. Plus formellement, cela revient à écrire :

$$(\mathbf{a}^*, \mathbf{b}^*) = \underset{\mathbf{a},\mathbf{b}}{argmax}\ corr(\mathbf{a}^T\mathbf{X}, \mathbf{b}^T\mathbf{Y}) \tag{2.3.1}$$

avec $\mathbf{X}$ et $\mathbf{Y}$ les représentations unimodales, $\mathbf{a}$ et $\mathbf{b}$ deux vecteurs de dimensions appropriées et *corr* la fonction de corrélation.

Un exemple orienté réseau de neurones est l'autoencodeur bimodal proposé par Ngiam *et al.* [210] qui prend en entrée la concaténation des deux modalités et contraint la représentation cachée obtenue à permettre de reconstruire les deux modalités à la fois.

Mais il est possible par exemple d'essayer de maximiser la corrélation entre les représentations unimodales, comme cela a été proposé par Andrew *et al.* [11] avec une approche neuronale profonde de l'Analyse Canonique des Corrélations [119]. Mais aussi par Chandar *et al.* [56] avec une approche appelée CorrNet (pour Correlational Network) basée sur des AEs dont les représentations internes sont contraintes à être corrélées. Engilberge *et al.* [93] proposent d'utiliser une similarité cosinus pour appliquer une contrainte moins forte sur les représentations. Enfin, Shahroudy *et al.* [251] utilisent des cascades de factorisation orthogonales pour contraindre les représentations unimodales à présenter à la fois une partie redondante et une partie totalement orthogonale aux autres (et donc spécifique à une modalité).

Il existe également des méthodes avec des contraintes plus structurelles, telles que celle proposée par Neverova *et al.* [206], dont l'idée principale consiste à régulariser en masquant aléatoirement certaines modalités pendant l'entraînement. L'idée est ensuite étendue par Li *et al.* [170] qui proposent d'apprendre un masque stochastique. Une autre méthode de contrainte structurelle beaucoup utilisée dans

les domaines liés à la modalité textuelle consiste en la factorisation de tenseurs, de manière à pouvoir exhiber toutes les relations intermodales tout en conservant une dimensionnalité raisonnable [34].

2.3.3 Lien avec les approches multi-tâches

L'ensemble des méthodes de fusion présente beaucoup de similarités avec des techniques utilisées dans le contexte du multi-tâches. En effet, une taxonomie classique des approches de multi-tâches est de diviser celles-ci en deux catégories : partage dur des paramètres [55], qui correspond à extraire une représentation pour les différentes tâches avec les mêmes paramètres, et partage des paramètres par contraintes [301], qui consiste à extraire plusieurs représentations mais à établir des relations de similarités entre celles-ci. Cela n'est pas sans rappeler la taxonomie de la fusion multimodale que nous venons de présenter avec la fusion par projection, où une seule représentation multimodale regroupe l'ensemble de l'information et la fusion par contraintes, où les représentations sont séparées mais reliées par des contraintes de corrélation.

En poussant l'analogie entre multi-tâches et multimodal, il est également possible de trouver des méthodes de combinaison des représentations précoces [184] ou tardives [181], des techniques de factorisation de tenseurs [300], l'utilisation de méthodes d'attention [143] ou encore des partages de représentations dans plusieurs des couches [197, 242].

Ainsi nous verrons dans le Chapitre 6 que notre approche de fusion présente beaucoup de similarités avec des techniques développées pour des problématiques de multi-tâches et également qu'il existe un intérêt à combiner les deux problématiques.

2.4 Conclusions

Ce chapitre a permis de présenter quelques éléments clés des réseaux de neurones, tout en développant notamment les notions de tâches et représentations. Celles-ci permettent de traiter des problématiques diverses telles que le transfert de connaissances ou la distillation. Ces concepts seront utilisés tout au long du manuscrit et nous nous référerons donc à ce Chapitre à plusieurs reprises pour éviter des répétitions.

De manière plus spécialisée, nous avons ensuite donné un aperçu du large domaine de l'informatique affective, qui permettra de contextualiser les travaux présentés dans les Chapitres 3 et 4. Enfin, la taxonomie des techniques de fusion multimodale servira de base pour aborder le Chapitre 6, tandis que les notions relatives aux approches multi-tâche permettront de donner un regard différent sur le Chapitre 5.

Chapitre 3

Approches neuronales pour la reconnaissance d'émotion

Table des matières

3.1 Introduction

Ce chapitre introduit le problème de la reconnaissance d'émotions dans des contenus audiovisuels. Il peut être vu comme une étude préliminaire au reste de la thèse et permet d'introduire les différentes approches explorées lors de deux participations successives à une compétition internationale de reconnaissance d'émotion dans des vidéos. En ce sens, nous choisissons ici de conserver une présentation des travaux réalisés proches de ce qu'ils étaient lors de nos participations, afin de pouvoir ensuite mieux discuter les enseignements que nous en avons tirés. Cela nous a permis d'ouvrir plusieurs pistes de travail, qui sont explorées dans les chapitres suivants de cette thèse.

3.1.1 Motivations

La reconnaissance d'émotion automatisée est un sujet qui suscite beaucoup d'intérêt par ses applications dans plusieurs domaines. Il est par exemple possible d'imaginer faciliter le marketing par une meilleure compréhension du client [159], mais aussi de mieux personnaliser les interfaces homme-machine [38], ou encore d'apporter des solutions dans le domaine de la santé [287].

Les approches actuelles de reconnaissance d'émotion se concentrent sur une sous-tâche qui consiste en la reconnaissance d'une expression sociale de l'émotion. Et les approches d'apprentissage profond, rencontrent beaucoup de succès dans le traitement des images et peuvent expliquer le traitement très important accordé à l'analyse de l'expression faciale.

De ce fait, il existe de nombreux travaux sur le sujet, et notamment plusieurs compétitions et bases de données. Tout d'abord, la tâche de reconnaissance d'expression faciale dans des images statiques dominent dans les compétitions [112, 278, 79, 35], utilisant des bases de données de taille (100 à 1 000 000 d'images), de qualité (résolution de 48x48 à 256x256) et de diversité variables (prise d'image en laboratoire à prise d'image en conditions non controllées et bruitées) [81, 198, 172, 35]. Par la suite, la tendance se dirige vers la reconnaissance d'expression dans des contenus plus complexes, tels que des séquences d'images [185], des séquences d'images en trois dimensions [320], et des contenus audiovisuels [82] ou multimodaux au sens large [277, 323].

À travers toutes ces bases de données, il est intéressant de noter qu'il existe différents types d'annotation. En effet, même si une méthode très directe consiste à utiliser des classes discrètes représentées par des mots (*e.g.* joie, peur) [88], il peut y avoir un bénéfice certain à utiliser des valeurs continues suivant les axes excitation et plaisir [30, 225]. Tandis que dans le cas particulier de l'expression faciale, le FACS permet de modéliser des activations particulières du visage, dites Action Units (*e.g.* sourcils froncés, bouche ouverte) [89].

3.1.2 Le challenge Emotion in the Wild

Cette compétition repose en particulier sur une base de données nommée AFEW [81], composée de courtes vidéos extraites principalement de films et de téléréalité. Chaque vidéo dure quelques secondes et contient le plus souvent un acteur exprimant divers signaux sociaux. Le but de la compétition est de déterminer automatiquement l'émotion exprimée parmi sept classes (colère, dégoût, peur, joie, tristesse, surprise et neutralité).

La première édition de la compétition a eu lieu en 2013 et, bien que remportée par une approche neuronale profonde [140], de nombreuses méthodes s'appuyaient sur des descripteurs calculés à la main (repris en détail par [139]), tels que des motifs binaires locaux ou des descripteurs de Gabor pour modéliser les visages, et des spectrogrammes ou des coefficients cepstraux pour modéliser le son.

Dans les éditions qui suivent, les approches neuronales dominent. Ainsi, les gagnants des quatre dernières éditions utilisent des réseaux de neurones [302, 99, 128, 177] et s'intéressent principalement au traitement des visages et à leur fusion temporelle. Du fait de la faible taille de la base de données, des approches de transfert de connaissances sont proposées, avec des réseaux de neurones au préalable entraînés sur de la reconnaissance d'identité et d'émotion discrète [99, 158], de la détection de FACS [302], ou même de la classification de vidéos de sport avec le concept de convolutions 3D [99]. Concernant

le traitement du son, la plupart des approches utilisent les mêmes descripteurs manuels [97], même si certains travaux s'intéressent à l'utilisation de réseaux de neurones [224].

La technique la plus répandue pour la fusion des différents descripteurs ou scores obtenus consiste en une simple moyenne pondérée [99, 128, 177]. La littérature présente tout de même d'autres approches : une fusion à noyaux multiples [59], une fusion hiérarchique des descripteurs [262], une SVM appliquée à la concaténation des scores [28] ou encore une fusion à double canaux avec mécanisme d'attention [57]. Toutefois, il semble difficile d'évaluer l'apport de ces différentes approches de fusion, celles-ci étant appliquées à des descripteurs ou des scores donnant des performances très différentes pour chaque travail.

Enfin, il est important de noter que cette base de données présente certaines difficultés inhérentes à sa faible taille / grande dimensionnalité, son annotation bruitée [139] et ses ensemble de validation et de test présentant des distributions très différentes.

3.2 Reconnaissance d'émotions à partir de différentes modalités

Lors de notre première participation à cette compétition, nous avons fait un certain nombre de choix pragmatiques basés sur les résultats obtenus par les approches précédentes. En rupture avec cette méthodologie, notre seconde participation propose une philosophie de simplification des modèles. Néanmoins, nous faisons le choix dans cette section et dans ce chapitre de présenter l'ensemble des techniques utilisées pour les deux participations, qui peuvent être schématiquement représentées de la même manière Ainsi, nous étudions tout d'abord plusieurs types de descripteurs de l'expression faciale et du son. Par la

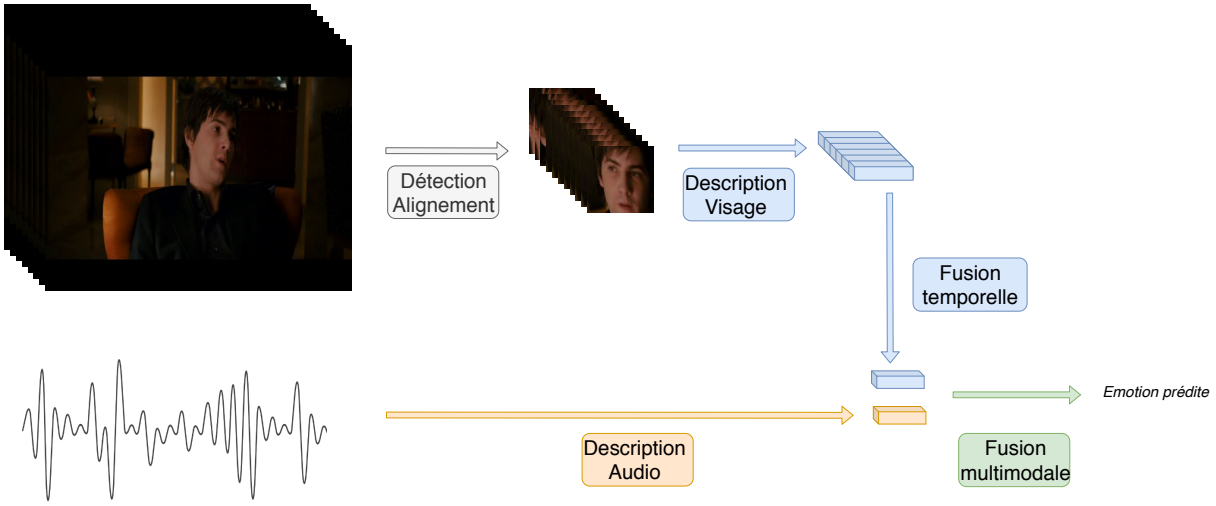


FIGURE 3.2.1 – Vue d'ensemble de la méthode utilisée. La modalité visuelle correspond d'abord a une *extraction des visages* et une *description visuelle* pertinente de ceux-ci. Les représentations obtenues sont *fusionnées temporellement*. Une *description de l'audio* est également obtenue par une chaîne plus simple. Enfin, les deux modalités (vision et audio) sont combinées par *fusion multimodale* pour prédire une classe d'émotion.

suite, concernant plus particulièrement la partie visuelle, diverses méthodes de sélection et fusion temporelles sont détaillées. Enfin, l'ensemble des modalités permet d'obtenir une prédiction finale en utilisant des techniques de fusion multimodale.

3.2.1 Extraire des descripteurs de la modalité visuelle

Détection et alignement Dans un premier temps, une approche courante lorsqu'on traite des visages est de les détecter et de les aligner (pour que les points d'intérêt tels que les yeux soient toujours à la même position). Nous appliquons donc un détecteur de visages interne à Orange sur chaque image

FIGURE 3.2.2 – Exemple de visages extraits (avec un pas de 8 images) d'une vidéo de AFEW pour chacune des 7 classes. Les variations de luminosité, de positions du visage ou encore de contexte constituent une illustration des difficultés inhérentes à cette base de données. Notez que les visages obtenus par notre détecteur (à gauche) sont en couleur, avec un cadre plus large et une meilleure qualité que ceux fournis par les organisateurs de la compétition (à droite).

contenue dans les clips vidéos. L'alignement est effectué également avec un outil interne, similaire à celui proposé par Dlib [154].

Les organisateurs de la compétition [78] proposent également leur propre pré-traitement des visages [328, 294], mais celui-ci présente plusieurs désavantages, comme le montre la Figure 3.2.2. En effet, les visages proposés sont en niveau de gris (moins d'information) et avec un cadre très serré (plus de sensibilité au bruit d'alignement). De plus, le détecteur interne à Orange est plus facilement utilisable à large échelle, ce qui nous permet de traiter d'autres bases de données de visage et d'ainsi pouvoir facilement pré-entraîner des modèles. Nous considérerons dans la suite du chapitre que l'ensemble des visages a été pré-traité par notre détecteur et notre aligneur.

Descripteurs de l'expression 2D Le but est de tout d'abord extraire pour chaque visage une représentation pertinente de son expression. Pour obtenir cet extracteur de représentation, notre approche se base sur les réseaux de neurones et le transfert de connaissances. L'idée est d'entraîner un réseau de neurones à prédire l'expression faciale d'un visage dans une image. Les dernières couches cachées extraites par ce modèle constituent alors des descriptions de l'expression faciale, utilisables pour décrire de manière plus compacte les images contenues dans les vidéos de AFEW.

Lors de notre première participation en 2017, il n'existait pas de base de données de large taille contenant des images de visages avec leur annotation d'expression. Une approche courante est d'utiliser un modèle pré-entraîné auparavant sur une large base de données de visages pour résoudre une autre tâche. Notre approche s'inspire alors des éditions précédentes (*e.g.* , [99, 146]) et utilise un modèle *VGG* [217] pré-entraîné à la reconnaissance d'identité. L'avant-dernière couche de ce modèle (entièrement connectée de 4096 vers 4096) est ré-initialisée aléatoirement et la dernière couche de ce modèle est remplacée par une couche entièrement connectée avec sept sorties, correspondant aux 7 classes d'expressions. Les images de la base de données FER2013 [112] (environ 28 000 visages) sont alors utilisées pour affiner le *VGG* sur le problème de classification d'expression faciale. D'abord seule la dernière couche est entraînée, puis dans un deuxième temps l'ensemble des poids mais avec un taux d'apprentissage plus faible. Ce réseau obtient alors sur la partie privée de test de FER2013 une performance de 71.2% d'expressions faciales correctement classifiées, légèrement supérieure à l'état de l'art de 2013 (71.1%) [112] et supérieure aux résultats obtenus avec la même architecture en 2016 (70.74%) [99]. Ce *VGG* peut alors être utilisé pour extraire un vecteur de taille 4096 à partir d'une image de visage, constituant une

représentation de l'expression faciale plus compacte qu'une image de résolution moyenne.

Lors de notre seconde participation en 2018, l'émergence de base de données plus conséquentes permet un apprentissage direct d'un réseau de neurones à la prédiction d'expression faciale. Ainsi, nous choisissons d'entraîner un ResNet-18 [122] sur la base de données AffectNet [198] (300 000 visages utilisables). Cette base dispose de deux types d'annotations : 8 labels d'émotions discrètes et des valeurs d'*excitation / plaisir*. Cette double annotation permet un entraînement multi-tâche, en remplaçant la dernière couche du *ResNet* par deux couches entièrement connectées : un classifieur prédisant les huits labels et un régresseur linéaire estimant les valeurs d'*excitation / plaisir*. La somme des fonctions de pertes du classifieur (entropie croisée) et du régresseur (erreur quadratique moyenne) est minimisée pendant l'entraînement de manière à bénéficier d'un effet de régularisation. Ce *ResNet* permet alors d'extraire un vecteur de taille 512 relatif à l'expression faciale. Mais les différences d'annotation étant très importantes entre les vidéos d'AFEW et la base de données AffectNet, utiliser directement la représentation extraite ne semble pas pertinent. C'est pourquoi nous avons choisi d'affiner le *ResNet* en deux étapes (comme pour le *VGG*) sur un regroupement de deux bases de données : SFEW et RAF. SFEW [80] contient moins de 2 000 images mais il s'agit d'images extraites des vidéos de AFEW et ré-annotées avec des labels appropriés (qui ne sont pas toujours identiques au label unique de la vidéo). Tandis que RAF [172] est une base de données plus large (12 271 images pour l'entraînement) avec les mêmes labels que SFEW mais une meilleure qualité d'annotations due à son grand nombre d'annotateurs.

Enfin un aspect important pour l'apprentissage de cette première représentation de l'expression faciale est la régularisation. En effet, les deux réseaux (*VGG* et *ResNet*) ont été entraînés avec de l'augmentation de données (changement d'échelle, rotation, effet miroir, flou) pour améliorer la généralisation et notamment pallier les échecs du détecteur-aligneur de visages. Pour le *VGG*, notons également l'ajout de *dropout* [260] entre l'avant-dernière et la dernière couche avec un taux de non-activation des nœuds à 95%, de manière à régulariser drastiquement l'apprentissage. Pour le *ResNet*, la technique de *cutout* [77] est aussi utilisée. Elle consiste à masquer des parties aléatoires de l'image d'entrée et peut être vue comme une modélisation des occlusions.

Descripteurs de l'expression 3D Nous avons précédemment vu qu'il est possible d'extraire une représentation dite statique de l'expression. Néanmoins, les vidéos contiennent une notion de dynamique que les modèles précédents risquent de négliger. C'est pourquoi une approche avec un réseau convolutionnel 3d peut paraître pertinente, ayant déjà été utilisée dans cette compétition [99] et face à des problèmes similaires [22, 271]. L'idée de ce type de réseau est d'effectuer la convolution non plus sur une image (deux dimensions) mais sur une suite d'images, qui correspond donc à un volume (trois dimensions). Comme Fan *et al.* [99], nous avons choisi un modèle convolutionnel 3d originellement proposé par Tran *et al.* [271] et entraîné sur un million de vidéos de sports [144]. Ce modèle, appelé *C3D* par la suite, prend en entrée une suite de 16 images. En remplaçant la dernière couche entièrement connectée par un classifieur avec une sortie de taille 7, nous ré-entraînons le C3D en deux étapes (comme effectué pour précédemment pour le *VGG* et le *ResNet*).

Bien que l'idée qu'il existe des liens entre les vidéos de sports et les séquences de visages étudiées ne soit pas immédiate, il est possible de faire le parallèle avec l'approche classique de réseaux entraînés sur ImageNet [74] et donnant de bons résultats sur des tâches très différentes, notamment en reconnaissance d'expression dans des images [209]. Une explication plausible serait la qualité des descripteurs bas niveaux appris grâce au très grand nombre d'exemples fournis lors de l'entraînement.

Notre approche diffère des précédentes en s'intéressant à la sélection des entrées du réseau. En effet, Fan *et al.* [99] entraîne le C3D en choisissant des fenêtres de 16 images aléatoirement. Au moment du test, ils utilisent la fenêtre centrale de la vidéo. Le problème de cette approche est qu'il n'y a aucune garantie que la fenêtre centrale corresponde à un moment pertinent de la vidéo. De plus, lors de l'entraînement, une grande partie des émotions des fenêtres sélectionnées ne correspondent absolument pas au label de la vidéo, ce qui créé un bruit important. En effet les vidéos sont annotées comme un tout et certaines fenêtres temporelles peuvent tout à fait correspondre à des labels différents. En prenant l'exemple des visages de la Figure 3.2.2 nous retrouvons cette notion de différence entre le label local et le label

global. En effet, à partir de la séquence associée à la joie, il est tout à fait possible d'interpréter les deux premiers visages (chacun pris à 8 images d'intervalle) comme une expression de tristesse.

Pour répondre à ce problème, nous proposons une méthode inspirée de l'apprentissage multi-instance. Dans un premier temps, le *C3D* est ré-entraîné à partir de toutes les fenêtres temporelles de toutes les vidéos, ce qui est proche d'une sélection aléatoire des fenêtres mais à l'avantage de garantir que toutes les fenêtres ont été observées. Ensuite, pour être capable d'identifier les fenêtres temporelles les plus utiles, nous utilisons les scores du réseau pour pondérer chaque fenêtre de plus en plus finement, à chaque epoque. De manière plus formelle, nous notons :

$$w_{i,j} = e^{\frac{-s_{i,j}}{T(t)}} \tag{3.2.1}$$

avec $w_{i,j}$ le poids pour la $i^{ème}$ vidéo et la $j^{ème}$ fenêtre de cette vidéo, $T(t)$ un paramètre de température diminuant à chaque epoch (correspond à une passe complète sur l'ensemble d'entraînement) (epoch) t et $s_{i,j}$ le score (*i.e.* entropie croisée) de la $j^{ème}$ fenêtre de la $i^{ème}$ vidéo.

Les poids sont normalisés de manière à ce que pour chaque vidéo d'indice i, la relation $\sum_j w_{i,j} = 1$ soit respectée. Ainsi, le réseau de neurones obtenu, nommé *W-C3D* dans la suite, permet d'obtenir une représentation robuste de taille 4096 d'une fenêtre temporelle de 16 images.

Cette méthode présente beaucoup de similarités avec l'Apprentissage Multi-Instance (MIL) [7]. Celui-ci, dans sa version basique, consisterait à considérer chaque vidéo comme un sac d'objets, en l'occurrence de fenêtres temporelles, avec un seul label. Une façon directe d'appliquer le MIL serait alors d'entraîner le *C3D* en traitant chaque sac de fenêtres en une seule inférence et en choisissant comme prédiction le score maximal du batch. La fonction de perte serait alors calculée à partir de cette unique prédiction. Cela correspond au cas extrême de notre algorithme, où un seul poids serait non nul pour chaque vidéo.

Une perspective intéressante serait d'appliquer cette technique également au *VGG* ou au *ResNet*. En pratique, la difficulté d'une telle approche réside dans le fait que le nombre d'éléments dans un sac serait beaucoup plus élevé que pour le *W-C3D*, rendant très difficile d'identifier une "fenêtre" (en l'occurrence un visage) plus pertinent que les autres.

3.2.2 Sélection et fusion temporelle

Nous considérons dans cette sous-section que les modèles précédents (*VGG*, *ResNet*, *W-C3D*) ont permis d'extraire des représentations de l'expression des visages contenus dans les différentes vidéos d'AFEW. Nous disposons donc pour chaque séquence de visages des représentations associées (resp. de taille 4096, 512 et 4096), ainsi que des scores pour chaque visage (*i.e.* le vecteur de taille 7 après activation softmax).

Nous cherchons maintenant à obtenir une représentation générale de l'expression tout au long de la séquence. Une approche basique consiste à travailler directement avec les scores obtenus par un modèle pour chaque visage et à en prendre la moyenne temporelle comme score final associé à la vidéo. Il est aussi possible de prendre le score maximal suivant l'axe temporel.

Nous détaillons ici deux autres approches se servant des descripteurs plus bas niveau : celle de 2017, dite approche récurrente et celle de 2018, dite approche heuristique.

Approche récurrente Pour extraire une description temporelle d'une séquence, l'utilisation de réseaux récurrents et plus particulièrement de LSTM (réseau de neurones récurrents à mémoire court-terme et long terme) [22, 104] est fréquente. À la différence des travaux existants dans cette compétition [99, 78], nous avons choisi d'utiliser une LSTM prenant des séquences de longueur variable en entrée, de manière à pouvoir traiter chaque séquence dans son ensemble. Pour éviter un sur-apprentissage dû au faible nombre de vidéos (773) vues lors de l'entraînement, nous avons appliqué un dropout sur les cellules de la LSTM et pénalisé la norme L2 des poids du classifieur final.

En pratique, nous avons entraîné plusieurs modèles de LSTM, chacun composé d'une partie récurrente retournant un vecteur de description et d'une partie classifieur retournant un score final.

— *VGG-LSTM* prend en entrée les représentations des visages extraites par le *VGG* et retourne une description de taille 297 ainsi qu'un score.

— *C3D-LSTM* prend en entrée les représentations des fenêtres temporelles extraites par le *W-C3D* et retourne une description de taille 304 ainsi qu'un score. Les fenêtres temporelles peuvent se chevaucher ou être totalement séparées comme dans la partie basse de la Figure 3.2.3.

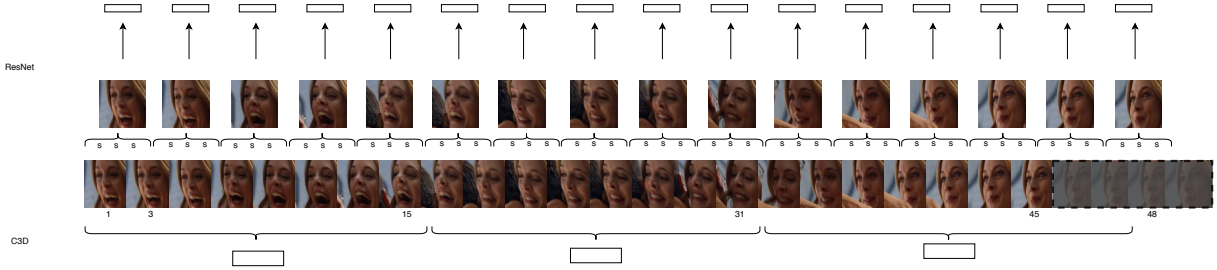


FIGURE 3.2.3 – Illustration du processus de sélection des visages dans une séquence de 45 images. Pour des raisons de mise en page, seul un visage sur deux est affiché. La partie grisée à droite correspond à des visages rajoutés à la séquence pour atteindre un nombre divisible par n. En haut, la méthode heuristique suivie en 2018 avec le *ResNet* consiste à récupérer la valeur maximale du score **s** pour chaque visage. Puis au sein de chaque petite fenêtre le visage qui a le score le plus élevé est choisi. Cela permet d'obtenir finalement une séquence de n visages. En bas, des exemples de fenêtres temporelles données en entrée de la *C3D* en 2017, ici sans recoupement entre les fenêtres. Pour le *VGG*, la description est extraite de tous les visages.

Approche heuristique Lors de notre deuxième participation, notre approche est basée sur la recherche d'un modèle le plus simple possible, avec le moins de paramètres possibles entraînés sur AFEW. C'est pourquoi nous utilisons uniquement la description issue du *ResNet*.

Ainsi une vidéo d'indice i est pour l'instant représentée par L_i vecteurs de taille 512 et des scores associés. Nous définissons alors une heuristique pour échantillonner ces vidéos en un nombre fixe n de vecteurs ($n = 16$ dans le reste du papier). Pour cela, comme illustré en haut de la Figure 3.2.3, nous proposons simplement de diviser la séquence de taille L en n groupements. Du fait que L n'est pas forcément divisible par n (*e.g.* dans la Figure 3.2.3, $L = 45$), la taille des groupements est un arrondi à l'entier le plus proche. Il est alors nécessaire de ne pas tenir compte des derniers visages de la séquence dans le cas d'un arrondi à l'inférieur, ou de compléter la séquence (en répétant le dernier visage comme dans la Figure 3.2.3 partie grisée) dans le cas d'un arrondi au supérieur. Pour chaque groupement, nous récupérons donc un arrondi de $\frac{L}{n}$ vecteurs représentant des visages et autant de scores associés. À l'intérieur d'un groupement, un seul vecteur est choisi en comparant les scores ; chaque score **s** étant un vecteur de 7 nombres, l'idée est de le remplacer par sa valeur maximale et d'ensuite comparer les valeurs maximales et de choisir le vecteur associé à la plus élevée. Autrement dit le vecteur sélectionné correspond au score maximal à la fois sur l'axe des classes et sur l'axe temporel.

Cette heuristique permet donc de représenter chaque vidéo par une matrice de dimensions $(n, 512)$. Cette matrice est alors utilisée pour la classification en utilisant possiblement trois méthodes de réduction temporelle :

— une moyenne simple sur l'axe temporel, donnant un vecteur de taille 512, qui devient l'entrée d'un classifieur linéaire prédisant une des 7 classes d'émotions.

— une moyenne pondérée, qui peut être vue comme une forme d'attention, chacun des n vecteurs possédant un poids. Ces poids sont calculés à partir des valeurs excitation-plaisir (soit un vecteur de contexte de taille $16 \times 2 = 32$) que le *ResNet* entraîné en multi-tâche sur AffectNet a la possibilité de fournir.

— un LSTM complété par un simple classifieur.

3.2.3 Extraire des descripteurs du son

Comme son nom l'indique, la base de donnée AFEW s'intéresse principalement aux expressions faciales des acteurs, d'où l'énorme intérêt porté aux visages. Néanmoins, il est essentiel d'être capable de prendre en compte l'information de contexte qu'apporte le son. En effet, dans un contexte plus que bruité, plusieurs approches précédentes [302, 79, 99] évaluent un gain significatif à la prise en compte de cette modalité.

Pour extraire un contexte émotionnel à partir du son, nous avons fait le choix de reprendre une méthode particulièrement simple basée sur des descripteurs extraits "à la main".

Descripteurs manuels La grande majorité des participants des trois dernières éditions [79] utilisent le logiciel OpenSmile [97], qui permet notamment d'extraire des descripteurs relatifs à l'émotion. Ces descripteurs sont rassemblés en un vecteur de taille 1582. Un classifieur telle qu'une machine à vecteur de support [70] est alors entraîné à prédire les labels des vidéos [302, 99].

Lors de notre première participation, nous avons choisi d'utiliser un réseau de neurones à 2 couches comme classifieur. Nous l'avons entraîné en prenant en entrée les descriptions extraites par OpenSmile. Pour aider à la convergence, nous avons utilisé de la normalisation par batch, ainsi que du dropout. Une fois le réseau entraîné, il est alors possible soit de directement obtenir un score de taille 7, soit un vecteur de description de taille 279.

Lors de notre seconde participation, nous avons décidé d'étudier une autre alternative, consistant à utiliser des données externes. Pour cela, nous récupérons la partie audio de la base de données IEMOCAP [44]. Celle-ci contient 12 heures de vidéos de personnes exprimant des émotions lors de diverses intéractions. L'enregistrement est fait en laboratoire (donc en condition contrôlées) et une annotation par catégories d'émotions (proches de AFEW) est disponible. Nous utilisons également OpenSmile pour extraire des vecteurs de description de chaque élément de IEMOCAP, puis nous entraînons à partir de ceux-ci un réseau de neurones avec une couche cachée de taille 64, utilisant de la normalisation par batch et du dropout. Finalement, ce réseau (avec sa dernière couche remplacée par une couche linéaire appropriée) est ré-entraîné en deux étapes sur AFEW. Une variante également testée a été d'entraîner un modèle de Random Forest [44] sur la représentation de taille 64.

3.2.4 Fusion multimodale

Par définition, le problème traité est multimodal, dans le sens où au moins deux modalités entrent en jeu : visuel et audio. Différentes méthodes de fusion sont étudiées lors de la première participation, où nous considérons que le *VGG-LSTM* et le C3D-LSTM traitent deux modalités différentes. Lors de la seconde participation, nous nous limitons à une moyenne des scores finaux et nous expliciterons ce choix dans la section relative aux expériences.

Moyenne et pondération Une première approche basique consiste simplement à prendre les scores de chacune des modalités et à définir le score final comme leur moyenne. Les différentes modalités peuvent également être pondérées, les poids étant choisis par exemple pour maximiser la performance sur le jeu de validation. D'autres méthodes utilisant les scores peuvent être envisagées, telles qu'effectuer un vote majoritaire ou concaténer les scores et entraîner un classifieur à prédire un score finale.

Concaténation de vecteurs Nous proposons de concaténer les vecteurs issus des différentes modalités en une seule représentation puis d'apprendre un perceptron multi-couches sur AFEW.

Moddrop Pour améliorer l'approche précédente, nous nous sommes inspirés de la méthode proposée par Neverova *et al.* [206], qui obtient d'excellents résultats pour de la reconnaissance de gestes multimodale. L'idée est de masquer aléatoirement une ou des modalités pendant l'entraînement, et d'ainsi éviter le sur-apprentissage et permettre d'être plus robuste à la dégradation de certaines modalités. Nous avons

utilisé trois modalités (issues du classifieur audio, du *C3D-LSTM* et du *VGG-LSTM*) et avons appliqué cette technique.

En revanche, un élément important pour assurer la bonne convergence du réseau est qu'il faut que les paramètres propres à chacune des modalités soient appris en premier avant de chercher à apprendre des corrélations entre modalités. Neverova *et al.* [206] conditionnent la matrice des poids de la première couche du réseau pour que les blocs diagonaux (donc contenant des paramètres relatifs uniquement à une modalité spécifique) soient les seuls non-nuls durant les premières itérations.

Plus formellement, nous pouvons écrire la première couche du réseau comme :

$$\begin{pmatrix} \mathbf{h_1} \\ \mathbf{h_2} \\ \mathbf{h_3} \end{pmatrix} = \begin{pmatrix} \mathbf{W_{1,1}} & \mathbf{W_{1,2}} & \mathbf{W_{1,3}} \\ \mathbf{W_{2,1}} & \mathbf{W_{2,2}} & \mathbf{W_{2,3}} \\ \mathbf{W_{3,1}} & \mathbf{W_{3,2}} & \mathbf{W_{3,3}} \end{pmatrix} \begin{pmatrix} \mathbf{x_1} \\ \mathbf{x_2} \\ \mathbf{x_3} \end{pmatrix}$$

Avec W la matrice des poids de la première couche du réseau, $\mathbf{h}$ le vecteur caché du réseau de neurones, et $\mathbf{x}$ l'entrée du réseau. Chaque modalité est représentée par un indice allant de 1 à 3 et ainsi $\mathbf{W_{i,j}}$ représente un bloc de paramètres allant de la modalité i à la modalité j.

Le fait d'appliquer la contrainte $\forall i \neq j \quad \mathbf{W_{i,j}} = 0$ puis de la relâcher brutalement perturbe l'apprentissage du réseau sur AFEW. Nous proposons donc d'appliquer cette contrainte d'une manière plus continue. Pour cela, nous ajoutons dans la fonction de perte une pénalisation L2 des poids appartenant aux blocs non diagonaux. La contribution de cette pénalisation à la fonction de perte est multipliée par un facteur γ diminuant exponentiellement au cours de l'apprentissage. Ainsi, une valeur importante de γ en début d'apprentissage assure que les relations unimodales sont bien apprises, tandis qu'une valeur quasi nulle en fin d'apprentissage permet de progressivement ré-introduire les relations entre modalités.

Arbre de fusion par les scores Le but de cette approche est de fusionner l'information sémantique apportée par les scores avec des informations plus bas niveau contenues dans les vecteurs de description des modalités. Elle s'inspire d'une méthode qui a été très utilisée en segmentation d'image, appelée autocontexte [275]. L'idée est de tout d'abord entraîner un modèle à prédire la segmentation des images (exemple de ce qu'est une segmentation d'image en Figure 3.2.4). Ensuite, chaque image est accompagnée de la segmentation obtenue et remise en entrée d'un nouveau modèle. En ré-itérant cette opération, le processus converge à la manière d'une chaîne de Markov (*cf.* [275]) et apporte des bénéfices importants, provenant d'un mélange entre l'information sémantique et l'information d'entrée.

Nous proposons une méthode entièrement neuronale qui s'inspire de l'autocontexte. Pour cela, nous nous plaçons dans le cas où chaque modèle unimodal a été entraîné et permet d'extraire pour une entrée donnée, un vecteur de représentation et un score (qu'on appellera *score initial*).

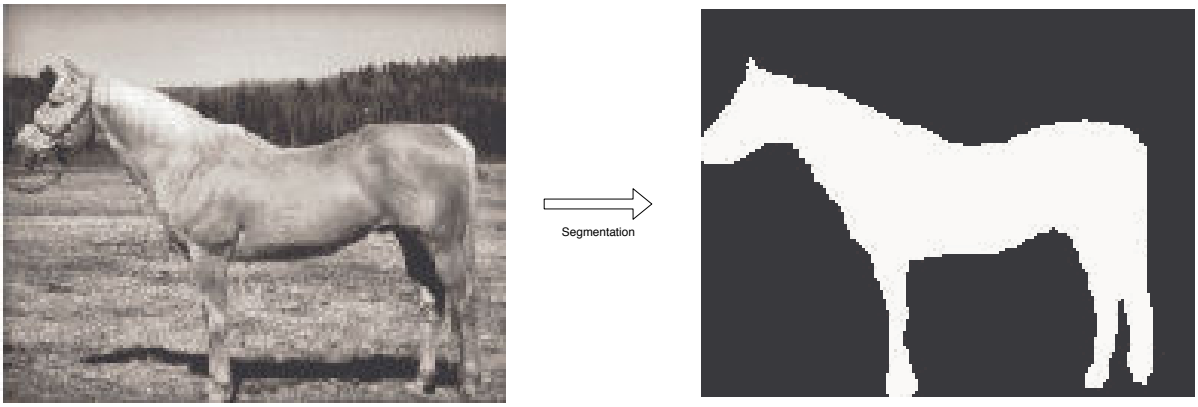


FIGURE 3.2.4 – Exemple de segmentation d'une image. Figure issue de [275]

Comme illustré en Figure 3.2.5, à chaque vecteur issu d'une des trois modalités, nous appliquons alors une couche entièrement connectée pour prédire à nouveau un score (vecteur de taille 7), qu'on

appellera *score affiné*. Or pour chaque modalité nous disposons déjà du *score initial*, qui dans le cadre de l'autocontexte peut être vu comme la première segmentation obtenue. A chaque score affiné d'une modalité sont alors concaténés les scores initiaux des deux autres modalités, permettant d'obtenir un vecteur de taille $3 \times 7 = 21$.

Ce vecteur est alors mis en entrée d'une couche entièrement connectée, qui permet d'obtenir un vecteur de taille 7 identifiable à un score (rectangle grisé sur la Figure 3.2.5). Les trois vecteurs obtenus sont encore concaténés et une couche entièrement connectée permet d'obtenir le score final. Cette étape peut être comparée à une seconde itération de l'autocontext, permettant un affinage du score.

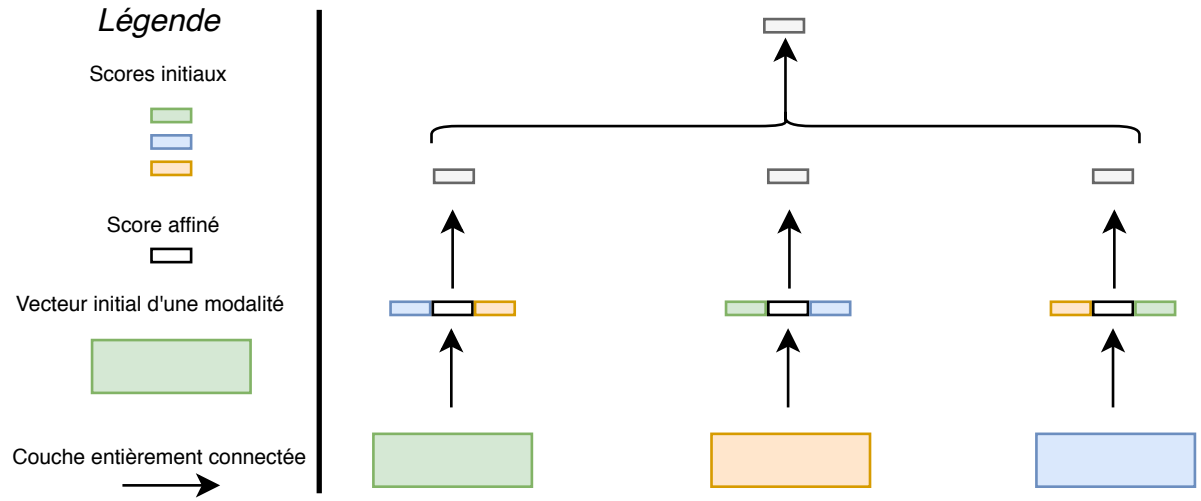


FIGURE 3.2.5 – Principe général de l'arbre de fusion par les scores. Les trois modalités sont représentées par les trois couleurs. Cette méthode est présentée avec trois modalités mais se généralise pour un nombre quelconque de modalités

3.2.5 Sélection des modèles

Deux approches différentes Nos deux participations diffèrent fondamentalement dans les critères de choix utilisés.

Lors de la première participation à la compétition, l'idée fondamentale qui guide la méthode est d'obtenir une performance la plus élevée possible sur un ensemble de validation.

Au moment de la seconde participation, la philosophie que nous avons choisi d'appliquer est celle du rasoir d'Occam, pouvant se résumer en "Les multiples ne doivent pas être utilisés sans nécessité". De manière plus pragmatique, cela signifie que le nombre de paramètres entraînés sur cette base de données de faible taille doit être le plus limité possible et donc, qu'entre deux solutions donnant des résultats proches sur l'ensemble de validation, nous préférerons toujours la plus simple (en termes de nombre de paramètres mais aussi d'implémentation). Cette façon de procéder trouve également son inspiration dans des approches parcimonieuses telles que le critère d'information d'Akaike [5].

Architectures et paramètres Les méthodes proposées contiennent un grand nombre d'hyperparamètres et d'élements d'architecture (*e.g.* nombre et taille des couches cachées d'un réseau) à fixer. Lors de la première édition, nous avons choisi ces éléments à partir d'une recherche par quadrillage aléatoire évaluée par la performance sur le jeu de validation de AFEW. Mais l'annotation de l'ensemble de validation est bruitée. Et de plus les distributions des ensembles de validation et de test sont connues pour être différentes [139, 158]. Cela implique qu'une bonne performance sur l'ensemble de validation ne se propage pas toujours sur l'ensemble de test et nous a poussé pour notre seconde participation à s'intéresser à trois évaluations alternatives :

1. entraîner plusieurs fois le même modèle mais avec des initialisations différentes et conserver la performance moyenne et l'écart-type.
2. rassembler les ensembles d'entraînement et de validation et appliquer une validation croisée.
3. prendre en compte la distribution de l'ensemble de test (très différente de celle des ensembles d'entraînement et de validation) en calculant la performance par classe et en les pondérant par le nombre d'éléments de cette classe dans l'ensemble de test. Nous parlerons d'ensemble de validation pondéré au moment de l'évaluation.

Ensemble Les techniques d'ensemble sont très présentes dans les approches des gagnants de nombreuses compétitions récentes [314, 196, 150, 12, 307].

Lors de notre première participation, nous avons simplement utilisé deux *VGG-LSTM* avec des initialisations différentes, et avons aussi défini deux modèles de *C3D-LSTM* en utilisant des fenêtres temporelles sans recoupement pour l'un et avec 8 visages en commun pour l'autre. Lors de notre seconde participation, nous avons implémenté un ensemble de classifieurs avec différentes initialisations aléatoires pour la partie visuelle, en calculant la moyenne de leurs scores comme score global.

3.3 Expérimentations et participations au challenge EmotiW

Cette section présente une validation expérimentale des différentes méthodes. Pour cela, nous proposons en premier lieu une analyse de la base de données AFEW, de manière à mieux comprendre les enjeux de la compétition. Ensuite, l'ensemble des descripteurs utilisés est évalué et les apports des fusions temporelles et multimodales sont étudiés. Nous détaillons également les résultats obtenus par le modèle global lors de nos deux participations au challenge Emotion in the Wild. Enfin, nous ouvrons une discussion sur les résultats obtenus et de possibles perspectives.

3.3.1 La base de données AFEW

Cette base de données présente plusieurs difficultés. Un élément essentiel réside dans le fait que le nombre de vidéos disponibles pour l'entraînement est particulièrement faible (773) comparé à de nombreux autres problèmes du même type, avec des entrées de très grandes dimensionnalités. De plus, il est intéressant de noter, à travers la Table 3.3.1, l'importante différence entre les distributions des ensembles de validation et de test, pouvant poser de gros problèmes d'adaptation du modèle et donc de généralisation.

Classe	**Entraînement**	**Validation**	**Test**
Colère	133 (17.2 %)	64 (16.7 %)	99 (15.2 %)
Dégoût	74 (9.6 %)	40 (10.4 %)	40 (6.1 %)
Peur	81 (10.4 %)	46 (12 %)	70 (10.7 %)
Joie	150 (19.4 %)	63 (16.4 %)	144 (22 %)
Tristesse	117 (15.1 %)	61 (15.9 %)	80 (12.3 %)
Neutralité	144 (18.6 %)	63 (16.4 %)	191 (29.2 %)
Surprise	74 (9.6 %)	46 (12 %)	29 (4.4 %)
Total	773	383	653

TABLE 3.3.1 – Distribution des classes pour les ensembles d'entraînement, de validation et de test de la base de données AFEW.

Un autre point important consiste en la fiabilité de l'annotation. En effet, celle-ci a été réalisée en suivant un processus particulier [81]. Un ensemble de vidéos a été sélectionné en se servant de mots-clés relatifs à l'émotion et en cherchant ceux-ci dans les didascalies des scripts de films, séries et émissions de téléréalité. Les passages vidéos sélectionnés et "automatiquement" annotés ont alors été visionnés

et triés par quelques annotateurs humains. Mais comme le montrent les auteurs de la base de données RAF [172], le consensus entre les annotateurs n'est pas immédiat. En effet, le fait d'utiliser seulement 7 mot-clés pour décrire un état émotionnel introduit une certaine subjectivité [30]. Par exemple une vidéo peut correspondre à un état de surprise et de joie, et le fait de devoir choisir un des deux états repose alors sur la perception subjective de l'annotateur.

Pour valider cette hypothèse dans le cas de AFEW, nous avons également évalué la performance de l'humain sur l'ensemble de validation. Pour six annotateurs différents, la proportion de prédictions correctes varie de 60% à 82%, ce qui vient compléter l'évaluation à 60% observée dans [139]. Un autre point important est l'accord entre ces six annotateurs, qui est total (six prédictions identiques) sur seulement 52% des vidéos et partagé par au moins deux annotateurs sur 68%. Ainsi, il faut garder à l'esprit lors de la construction et l'évaluation d'un modèle que l'annotation elle-même présente des incohérences pouvant menacer la capacité de généralisation.

3.3.2 Évaluation des descripteurs

Nous reportons ici les résultats obtenus par les différents types de descripteurs au cours de nos deux participations.

Descripteurs visuels Nous reportons les performances des approches *VGG*, *C3D*, *W-C3D* utilisées lors de la première participation et *ResNet* lors de la seconde participation.

Nous comparons tout d'abord dans la Table 3.3.2 les approches traitant chaque visage séparément. L'idée est d'abord d'évaluer les modèles sur des bases de données d'expression faciale statique (SFEW et RAF), sur lesquelles les modèles sont à chaque fois ré-entraînés avant évaluation. Une évaluation plus en lien avec la compétition consiste à extraire pour chaque vidéo de AFEW le score de chaque visage et de prendre comme prédiction le score moyen. Nous évaluons ainsi une performance sur l'ensemble de validation d'AFEW, sans entraînement aucun.

Le *VGG* de notre première participation est aujourd'hui dépassé sur les trois bases de données choisies et fait partie des modèles les plus lourds à manipuler. Il est largement dépassé par le *ResNet*, ce qui s'explique par l'entraînement de celui-ci dès le départ sur une tâche de reconnaissance d'émotions, avec une base de données de taille conséquente et présentant beaucoup de variations [198]. Une étude plus poussée serait également nécessaire pour identifier la contribution du changement d'architecture (VGG vers ResNet) sur la performance.

En comparaison d'autres approches à l'état de l'art, le *ResNet* est dépassé sur les deux bases d'expressions faciales statiques mais il semble présenter une meilleure capacité de généralisation quand il est évalué sur AFEW. De plus, son nombre de paramètres et son temps de calcul restent raisonnables comparés à des approches comme le *Covariance Pooling* [2]. Cela a son importance, étant donné qu'une vidéo de AFEW contient en moyenne 60 visages détectés. Enfin, les résultats obtenus pour le *ResNet* correspondent à une performance moyenne sur 50 entraînements avec des initialisations différentes. L'écart-type associé est de 0.5%.

Modèle	RAF	SFEW	AFEW	FLOPs (millions)	Paramètres (millions)
CNN Ensemble [307]	_	55.96	_	>2000	>500
HoloNet [128]	_	_	46.5	**75**	_
Cov. Pooling [2]	**85.4**	**58.14**	46.71	1600	7.5
VGG	74	45.2	41.4	1550	138
ResNet	80	55.8	**49.4**	180	**1.7**

TABLE 3.3.2 – Comparaison de différents modèles de reconnaissances d'expression faciale sur trois bases de données d'expressions faciales : dans des images avec SFEW et RAF ; dans des séquences d'images avec AFEW. Le nombre d'opérations en virgule flottante et le nombre de paramètres sont également donnés à titre indicatif.

Lors de notre première participation, nous avons également utilisé le *C3D* pour extraire des vecteurs de description de fenêtres temporelles contenant 16 visages. Ainsi nous comparons dans la Table 3.3.3 différentes variantes de traitement des fenêtres temporelles. Les *C3D* et *W-C3D* sont évalués sur l'ensemble de validation en prenant la fenêtre temporelle centrale de chaque vidéo. Lors de l'entraînement, Fan *et al.* [99] propose de sélectionner aléatoirement la fenêtre temporelle lors de l'entraînement. Tout comme d'autres auteurs [158], notre tentative de reproduction de cette approche n'atteint pas les mêmes performances (5.7% de moins). Lorsque nous utilisons la fenêtre temporelle centrale pour l'entraînement également, nous observons un gain de performance (38.7%). Notons tout de même que cette valeur est une performance moyenne sur 10 entraînements avec un pic à 39.4% et un écart-type de 0.9%.

Méthode	Performance sur le validation
C3D par Fan *et al.* [99]	39.7 %
C3D, fenêtre aléatoire	34 %
C3D, fenêtre centrale	38.7 %
W-C3D (pas de recoupement)	**42.1 %**
W-C3D (8 images de recoupement)	40.5 %
W-C3D (15 images de recoupement)	40.1 %

TABLE 3.3.3 – Performance de différentes variantes du *C3D* sur l'ensemble de validation de AFEW.

Ensuite, nous pouvons valider l'intérêt de l'approche *W-C3D* que nous avons proposée, qui dépasse significativement l'approche de Fan *et al.* [99]. Un point intéressant est également que le fait que la meilleure approche consiste à ne pas laisser de recoupement entre les fenêtres temporelles. Cela revient finalement à diminuer le nombre de fenêtres temporelles parmi lesquelles il faut identifier les "bonnes" fenêtres temporelles, ce qui peut expliquer que la tâche du réseau soit alors plus facile.

Descripteurs audio Un aspect très important de l'évaluation des descripteurs audio sur AFEW vient du fait que l'annotation des vidéos est principalement visuelle. Ainsi, le son d'une vidéo peut tout à fait ne contenir aucune information utile pour prédire l'émotion (*e.g.* un claquement de porte dans une scène où la personne est neutre). Néanmoins, bien que les performances reportées pour l'audio soient particulièrement basses comparées à celles reportées pour la modalité visuelle, le son apporte une information de contexte très complémentaire.

Lors des deux participations, nous avons proposé un MLP prenant la description d'OpenSmile en entrée. Les résultats reportés dans la première colonne de la Table 3.3.4 différent. La différence peut venir du nombre de paramètres beaucoup moins importants du modèle de 2018. Un élément très important réside dans le fait que dans le cas d'une évaluation classique, une Random Forest donne les meilleurs résultats sur l'ensemble de validation. En revanche, lorsque nous reprenons l'évaluation sur l'ensemble de validation pondéré pour avoir la même distribution que le test, le modèle de 2017 est dépassé par celui de 2018. La Random Forest est légèrement dépassée par le MLP pré-entraîné sur ce validation pondéré et un autre élément joue en sa défaveur : elle atteint une performance de 100% sur l'entraînement, tandis que les MLP avoisinent tous les 45%, pouvant laisser penser que la Random Forest possède ici une mauvaise capacité de généralisation.

Enfin, comparées à d'autres approches de l'état de l'art, nos approches sont similaires en termes de performance. Bien que relativement bas, les résultats obtenus par le SoundNet [21] (réseaux de neurones traitant des sons par distillation de modèle visuelle) sont encourageants et laisse penser qu'utiliser une approche de distillation similaire à celle d'Albanieé*et al.* [6] serait bénéfique.

3.3.3 Évaluation de la fusion temporelle

Dans cette sous-section, nous présentons les résultats de différentes techniques pour la fusion temporelle des représentations visuelles. Intéressons-nous dans un premier temps aux courbes présentées dans la Figure 3.3.1. Celles-ci correspondent aux scores prédits par le *VGG* et le *ResNet* à partir de la séquence

Méthode	Performance sur le validation	Performance sur le validation pondéré
SoundNet [21]		
ré-entraîné [177]	33.5 %	_
SVM [128]	37.2 %	_
MLP 2017	36.5 %	40.6 %
MLP 2018	33.5 %	42.1 %
MLP 2018 pré-entraîné	35 %	**45.2 %**
Random Forest	**38.8 %**	44.3 %

TABLE 3.3.4 – Performance de différentes méthodes de descripteurs audio sur l'ensemble de validation d'AFEW et l'ensemble de validation pondéré suivant la distribution du test.

de visages fournies. Nous retrouvons bien la différence entre label local et label global déjà abordée dans les méthodes. En effet, bien que la séquence affichée ait pour label global la joie, les visages qui la constituent, si nous sommes privés de contexte temporel, peuvent tout à fait être interprétés comme des manifestations de tristesse, de colère ou de surprise. C'est pourquoi il a paru nécessaire de construire un modèle prenant en compte ce contexte temporel. De plus, confirmant les résultats des descripteurs visuels, les scores du *VGG* sont très bruités comparés à ceux du *ResNet*.

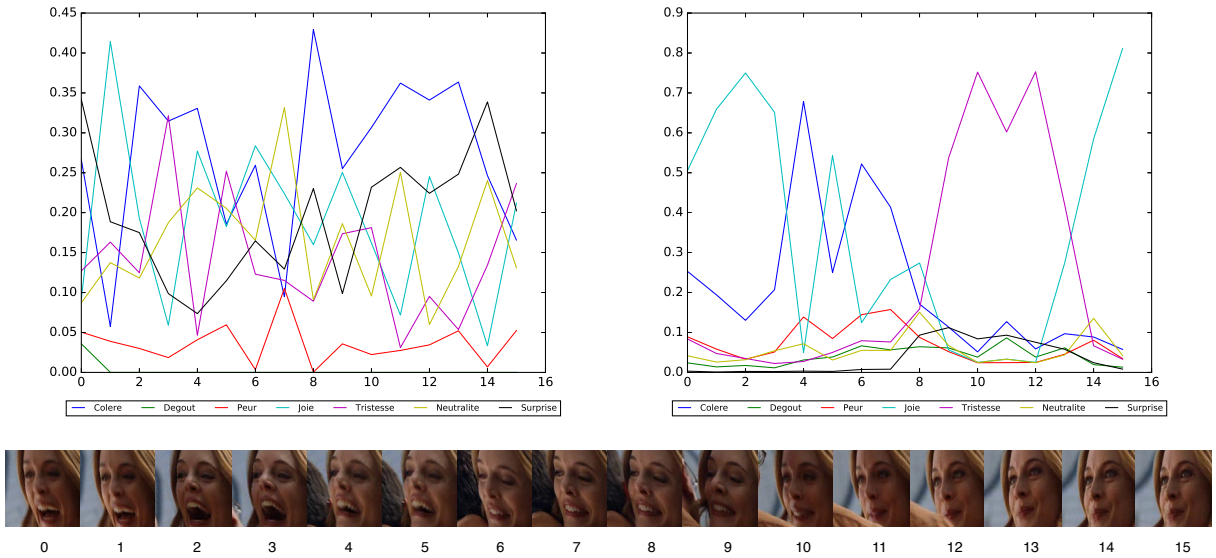


FIGURE 3.3.1 – Représentation des scores associés au visage d'une vidéo, extraits par le *VGG* à gauche et par le *ResNet* à droite. Les indices des visages sont en abscisses et les scores en ordonnées.

La Table 3.3.5 nous permet d'explorer l'efficacité de différentes méthodes. Dans le cadre de notre première participation, nous avons choisi d'utiliser une LSTM. Le modèle de fusion temporelle donnant les meilleurs résultats est alors une LSTM à une couche cachée, ce qui peut s'expliquer encore une fois par le faible nombre d'exemples d'entraînement. Nous pouvons noter une progression importante comparée à des approches similaires telles que celle de Fan *et al.* [99]. Cela peut s'expliquer par l'utilisation de techniques d'augmentation et par la prise en compte de tous les visages dans la vidéo. Nous notons également que l'approche basique d'aggrégation des scores proposée précédemment n'atteint que 41.4% et semble confirmer l'apport de la LSTM. En revanche, le *C3D-LSTM* améliore marginalement la performance du *W-C3D*.

Lorsqu'on utilise le *ResNet*, la Figure 3.3.1 nous permet déjà d'émettre l'hypothèse que la fusion temporelle est une tâche plus simple que dans le cas du *VGG*. Les résultats de la Table 3.3.5 nous le confirment, la performance étant beaucoup plus élevée que pour le *VGG*. En revanche, le fait d'utiliser une LSTM semble beaucoup moins bénéfique que des méthodes beaucoup plus basique, telle qu'une

Méthode	Validation	Validation pondéré
LSTM de Fan *et al*. [99]	45.42 %	_
FR-Net-B [158]	**53.5** %	_
VGG-LSTM, une couche	48.6 %	_
VGG-LSTM, deux couches	46.2 %	_
VGG-LSTM bidirectionnelle, une couche	46.7 %	_
VGG-LSTM bidirectionnelle, deux couches	45.2 %	_
C3D-LSTM, sans recoupement, une couche	43.2 %	_
C3D-LSTM, recoupement de 8 images, une couche	41.7 %	_
ResNet-LSTM, une couche	49.5 %	58.2 %
ResNet + Moyenne, un classifieur	49.7 %	60.5 %
ResNet + Moyenne, 4 classifieurs (ensemble)	50.4 %	61.2 %
ResNet + Moyenne, 50 classifieurs (ensemble)	*52.2 %*	61.7 %
ResNet + Moyenne pondérée, un classifieur	50.2 %	61.1 %
ResNet + Moyenne pondérée, 4 classifieurs (ensemble)	50.3 %	61.5 %
ResNet + Moyenne, 2 classifieurs + Moyenne pondérée, 2 classifieurs	50.1 %	**62.0** %

TABLE 3.3.5 – Résultats de différentes méthodes de fusion temporelle sur l'ensemble de validation et l'ensemble de validation pondéré de AFEW. Le lecteur notera que toutes les performances reportées pour le *ResNet* sont des moyennes sur 50 entraînements.

simple moyenne. De plus, ces approches simples permettent d'élaborer un ensemble de classifieur, améliorant la performance et atteignant sur l'ensemble de validation des performances proches de l'état de l'art actuel. L'utilisation de l'ensemble de validation pondéré apporte également un regard différent, puisque la performance des méthodes complexes semblent dans ce cas se dégrader tandis que les méthodes plus simples les dépassent.

3.3.4 Évaluation de la fusion multimodale

Méthode	Validation	Test
Vote majoritaire	49.3 %	_
Moyenne	47.8 %	_
Moyenne pondérée	50.6 %	**57.58 %**
Moddrop modifié	**52.2 %**	56.66 %
Arbre des scores	50.8 %	54.36 %

TABLE 3.3.6 – Performance des différentes méthodes de fusion en utilisant pour modalités les descriptions issues de *VGG-LSTM*, de *C3D-LSTM* et d'un MLP audio. Les résultats sont reportés sur les ensembles de validation et de test.

La Table 3.3.6 rapporte les résultats obtenus en utilisant les différentes stratégies de fusion employées lors de notre première participation. Les approches de base que sont la moyenne des scores ou le vote majoritaire donnent des résultats inférieurs à celui obtenu par le *VGG-LSTM*, dégradant donc la performance sur l'ensemble de validation.

Les méthodes plus sophistiquées telles que l'arbre des scores ou le Moddrop semblent prometteuses sur l'ensemble de validation mais sont dépassées par une simple moyenne pondérée sur le test. Cela souligne une mauvaise généralisation des approches avec un nombre important de paramètres dans le cas de AFEW. De plus, la question du niveau de fusion employé semble dépendre du problème à résoudre. En effet, dans le cadre d'AFEW, les méthodes les plus efficaces sont des fusions tardives, les seuls essais de fusion plus précoces étant des échecs. Pourtant, ces méthodes, face à d'autres problèmes (*e.g.* Moddrop sur de la reconnaissance de gestes) peuvent donner d'excellents résultats. Ainsi, bien que l'existence

d'une méthode de fusion universelle semble peu probable, la recherche d'un algorithme capable d'adapter la technique de fusion au problème traité semble à ce moment une perspective prometteuse.

La participation de 2018 se contente donc d'exploiter une moyenne pondérée des scores pour la première soumission, puis une simple moyenne pour le reste des soumissions.

3.3.5 Résultats finaux et discussion

Soumission		Test
Modalités	Fusion	
VGG-LSTM + *C3D-LSTM* + audio MLP	Moddrop	55.28 %
VGG-LSTM + 2 x *C3D-LSTM* + audio MLP	Moddrop	56.66 %
VGG-LSTM + *C3D-LSTM* + audio MLP	Arbre des scores	54.36 %
2 x *VGG-LSTM* + 4 x *C3D-LSTM* + audio MLP	Moddrop	56.51 %
2 x *VGG-LSTM* + 4 x *C3D-LSTM* + audio MLP	Moyenne pondérée	57.58 %
2 x *VGG-LSTM* + 4 x *C3D-LSTM* + audio MLP + validation	Moyenne pondérée	58.81 %
Vainqueurs EmotiW 2017 [128]		
ResNet-50+DenseNet-121+HoloNet+SVM+ Descripteurs	Ensemble Pondéré	60.3 %

TABLE 3.3.7 – Détails des soumissions de notre première participation, obtenus sur l'ensemble de test de 2017.

Nous tournons maintenant notre attention vers les résultats obtenus par le modèle dans son entièreté.

La meilleure performance sur l'ensemble de validation lors de la première et seconde participation sont respectivement de 52.2 % et 53.8 %. L'approche de base proposée par les organisateurs atteint quant à elle 38.81% [78]. Lors de cette compétition, les participants reçoivent les vidéos de test sans annotation et ont pendant deux semaines la possibilité de soumettre 7 propositions de prédictions d'émotion aux organisateurs. Pour chaque vidéo, une unique émotion doit être prédite.

Ensemble Pour les deux participations, nous avons pu étudier le bénéfice apporté par les méthodes d'ensemble. Il semblerait que dans le cadre de ce challenge, il est plus pertinent de combiner des modèles différents que de pousser l'ensemble à son extrême en répétant 50 fois le même modèle avec des initialisations différentes. Par exemple, lors de la seconde participation (*cf.* Table 3.3.8), le fait de rajouter une Random Forest en plus du MLP sur l'audio apporte 1.4% d'amélioration sur le test. En revanche, combiner de nombreuses fois le même classifieur visuel (avant-dernière ligne) aboutit à une performance de 59.4%, ce qui correspond à la même performance que la $4^{ème}$ ligne, qui n'utiliser que 2 classifieurs visuels. De plus, notre meilleure soumission (troisième ligne) avec 60.6% correspond à une combinaison de différentes méthodes visuelles et audio et ne totalise pas plus de 4 classifieurs.

Entraînement en ajoutant l'ensemble de validation Puisque le nombre de données d'entraînement est peu élevé, une technique déjà présente dans les éditions précédentes consiste à entraîner le modèle également sur l'ensemble de validation. Cette méthode est utilisée uniquement dans la dernière ligne des deux tables. Pour la Table 3.3.7, nous pouvons observer l'apport de l'entraînement en ajoutant l'ensemble de validation : la différence entre l'avant-dernière et la dernière ligne est de 1.2%, ce qui est significatif à l'échelle de cette base de données. Pour la Table 3.3.8, il est plus difficile d'identifier l'apport de l'ajout de l'ensemble de validation. On note tout de même qu'entre la première (pas d'ensemble, pas de validation) et la dernière soumission (ensemble et validation ajouté), la différence est de 3.3%.

Taille des modèles et temps de calcul Une différence essentielle entre les deux participations est la taille des modèles utilisés. En effet, lors de la première participation notre meilleure soumission contient 2 *VGG-LSTM* (*VGG* identique) et 4 *C3D-LSTM*, ce qui représente un nombre de paramètres énorme (plus de 500 millions) et un temps de calcul irréaliste pour de véritables applications. Avec "seulement"

Soumission	Test
ResNet Moyenne + Audio MLP	57.2 %
ResNet Moyenne + Audio MLP + Random Forest	58.6 %
2 x *ResNet* Moyenne + 2 x *ResNet* Moyenne Pond. + Audio MLP + Random Forest	60.6 %
2 x *ResNet* Moyenne Pond. + 2 x Audio MLP	59.4 %
2 x *ResNet* Moyenne + 2 x *ResNet* Moyenne Pond. + 2 x Audio MLP	60.4 %
50 x *ResNet* Moyenne + 2 x Audio MLP	59.4 %
4 x *ResNet* Moyenne + Audio MLP + validation	60.5 %
Vainqueurs EmotiW 2018 [177]	
Moyenne Pond. + Landmarks + VGG-LSTM + 4 DenseNet + InceptionNet + SoundNet	61.9 %

TABLE 3.3.8 – Détails des soumissions de notre seconde participation sur l'ensemble de test de 2018. La méthode de fusion est une moyenne pondérée pour la première ligne et une moyenne simple pour le reste.

2.8 millions de paramètres, notre dernière soumission de 2018 atteint 60.5%. Ce niveau de performance est proche de l'état de l'art publié en 2018 [177], qui atteint 61.87% et présente l'avantage d'un temps de calcul très inférieur à toutes les approches de ces dernières éditions.

Résultats de la compétition Nos approches, lors des deux éditions, ont atteint la troisième place sur une trentaine d'équipes participantes. La totalité des approches soumises utilisent des réseaux de neurones, avec une tendance à empiler énormément de modèles de grande taille pour gagner des points supplémentaires. Les gagnants de 2017 [128] modifient ainsi un ResNet-50, un DenseNet-121, un HoloNet en ajoutant une méthode "d'ensemble supervisé", qui relient de manière dense plusieurs couches cachées de chacun des réseaux au score final. Ils utilisent également les descripteurs manuels fournis par les organisateurs et une SVM pour la partie audio. Les gagnants de 2018 [177] utilisent les points d'intérêt du visage comme une modalité, ainsi qu'un ensemble d'un *VGG-LSTM* et de 4 DenseNet [133], combiné avec un SoundNet [21] pour l'audio. De plus, les auteurs ont annoté 3553 vidéos à la main, de manière à disposer d'un ensemble d'entraînement permettant de limiter le sur-apprentissage. Cette tendance à rassembler énormément de modèles n'est pas spécifique à ce challenge. Elle est par exemple présente lors de la première édition de la compétition Youtube-8M [196]. Les organisateurs ont alors limité la taille des modèles utilisables lors de la seconde édition, ce qui pourrait également être effectué dans les prochaines éditions d'Emotion In the Wild.

3.4 Conclusions

3.4.1 En résumé

Ce chapitre présente nos deux participations successives à la compétition de reconnaissance d'émotions Emotion in the Wild chez des personnes à partir de courtes séquences audiovisuelles. Nous avons donc développé et étudié un modèle composé de plusieurs blocs permettant de traiter ces séquences.

Un premier bloc permet de traiter la composante visuelle des vidéos, en extrayant des descriptions robustes de l'expression faciale. Pour cela, nous avons utilisé des réseaux de neurones pré-entraînés sur d'autres bases d'expressions faciales plus grandes. En 2017, nous utilisons donc un *VGG* et un *C3D* respectivement pré-entraînés sur une tâche de reconnaissance d'identité puis d'expression faciale, et d'action sportive. En 2018, nous proposons une méthode plus légère en nombre de paramètres, utilisant un *ResNet* entraîné sur une large base de données d'expressions faciales [198]. Le descripteur le plus pertinent semble être ce dernier, atteignant des performances proches de l'état de l'art sur deux bases de données d'expression faciale statique et prouvant sa robustesse en dépassant l'état de l'art de 2018 sur la base de données AFEW, qui est celle de la compétition.

Les modèles qui précédent permettent d'extraire des vecteurs de description des visages de chacune

des vidéos. Il est alors nécessaire de fusionner temporellement ces vecteurs pour obtenir un seul vecteur par vidéo. Nous étudions plusieurs méthodes, d'une simple moyenne temporelle à des réseaux récurrents tels que la LSTM. Les résultats obtenus tendent à prouver que l'ensemble d'entraînement n'est pas suffisamment large pour utiliser des techniques avec un grand nombre de paramètres, tels que des réseaux récurrents, qui sont clairement dépassés par des approches de moyenne simple ou pondérée.

La partie audio ne correspond pas toujours au label des vidéos, celles-ci étant annotées par rapport à l'expression faciale. Néanmoins, elle apporte un contexte complémentaire à la modalité visuelle. C'est pourquoi nous extrayons manuellement des descripteurs réputés pertinents pour la reconnaissance d'émotions et entraînons de simples classifieurs sur AFEW. Des approches totalement neuronales existent également mais restent encore aujourd'hui peu performantes sur cette base de données [177].

Ensuite, pour combiner l'ensemble des modalités nous étudions plusieurs techniques de fusion. Il s'avère finalement que dans le cadre spécifique de cette base de données, bien que donnant de bons résultats sur l'ensemble de validation, les méthodes complexes proposées généralisent mal sur l'ensemble de test et n'atteignent pas les performances d'une simple moyenne pondérée des scores des modalités. Cela s'explique probablement par un sur-apprentissage, mais souligne aussi la spécificité de chaque problème multimodal, impliquant de développer pour chacun une technique particulière.

Les modèles globaux proposés se classent troisièmes de la compétition en 2017 et en 2018, avec une progression de 58.8% (gagnants à 60.3%) à 60.6% (gagnants à 61.9%) [1]. Mais on observe surtout une réduction très importante du temps de calcul et du nombre de paramètres lors de la seconde participation. Cette simplification du modèle permet d'envisager un traitement "temps réel" avec l'aide d'une carte graphique, ce qui constitue une première parmi les meilleurs modèles proposés dans cette compétition. Et cela prouve également qu'une philosophie du rasoir d'Occam, consistant à systématiquement privilégier des solutions simples et légères lors de la construction du modèle global est pertinente dans un contexte aussi bruité et pauvre en données d'entraînement que celui traité dans cette compétition.

Enfin, nous pouvons conclure que cette compétition commence à saturer : l'amélioration des performances dans les dernières éditions est de l'ordre du pourcent, lorsqu'elles étaient au départ de l'ordre de la dizaine de pourcents. Cela peut s'expliquer par le manque de données, la similarité des méthodes utilisées lors des dernières éditions [82, 79, 78], mais également le fait que la performance humaine sur ce type de tâches se situe entre 60% et 80%, du fait de la subjectivité des annotations d'émotion.

3.4.2 Questionnements

Nous avons fait le choix de présenter ce chapitre comme une étude préliminaire. Nous en tirons maintenant un certain nombre de questionnements, nous poussant à explorer d'autre problèmes plus larges.

Comme représenter l'émotion ? Le traitement de cette base de données s'est révélé délicat pour plusieurs raisons, et plus particulièrement par le problème que posent les classes discrètes d'émotion lors de l'évaluation. En effet, comme nous l'avons vu précédemment, une expression peut contenir par exemple un mélange de surprise et de joie, aboutissant à une annotation subjective lorsqu'on utilise un seul mot-clé. Mais en utilisant simultanément un grand nombre de mot-clés, l'annotation discrète deviendrait alors extrêmement coûteuse et plus complexe à comprendre pour un être humain. Enfin, un réseau de neurones appris avec celle-ci présente une modélisation temporelle très bruitée, comme nous l'a montré la Figure 3.3.1.

Cela laisse donc penser qu'une annotation continue, comme les axes excitation-plaisir dans la compétition AVEC [277], pourrait apporter une meilleure qualité d'estimation de l'émotion. Malheureusement, ce type d'annotation n'est pas intuitif [29] et demande donc un effort beaucoup plus lourd pour obtenir une base de données. De plus, du fait de la difficulté de l'annotation, la plupart des bases de données

1. Il est important de noter que l'ensemble de test de 2018 contient celui de 2017, ainsi qu'une centaine de vidéos supplémentaires, considérées comme plus difficiles par les organisateurs (télé-réalité, occlusions multiples, changement de plan) [78]

sont de petite taille et en conditions controllées. Il est donc particulièrement difficile de trouver un bon compromis d'annotation.

Mais d'un autre côté, des travaux [148] se sont intéressés aux représentations cachées apprises par des réseaux convolutionnels pour de la classification d'émotion discrète. Et il s'avère que les cartes convolutives peuvent notamment être reliées à la représentation FACS évoquée en introduction. De ce fait, il serait intéressant d'analyser les liens entre les différents types d'annotations proposées dans la littérature psychologique et les représentations apprises par un réseau de neurones pour la reconnaissance d'émotions. Cette analyse est développée dans le Chapitre 4 dans le cadre restreint de l'annotation de l'expression faciale.

Comment transférer des connaissances d'un réseau de neurones ? Lors de la compétition, le choix des transferts de connaissances a été fait de manière empirique. En effet, nous avons par exemple utilisé le *VGG* pré-entraîné sur de la reconnaissance de l'identité puis de l'expression car celui-ci était connu pour donner de bons résultats [99]. Mais d'autres transferts se sont avérés bien meilleurs lors de notre seconde participation, comme celui effectué avec le *ResNet*.

Ces observations soulèvent alors une question importante : pourrait-on trouver le meilleur transfert de connaissances parmi les modèles pré-entraînés existants ? Certains travaux récents [309] tendent à prouver que cela est possible en construisant un espace permettant de modéliser la qualité de transfert d'une tâche vers une autre.

Dans le Chapitre 5, nous abordons cette problèmatique dans le contexte de la représentation des visages. Au lieu de construire un espace des tâches, nous proposons une approche pour transférer les connaissances d'une banque de modèles pré-entraînés sur diverses tâches de visage en un unique modèle, qui se révèle efficace sur une large variété de tâches liées aux problématiques d'analyse de visage.

Existe-t-il une méthode de fusion universelle ? En appliquant des techniques de l'état de l'art de la fusion multimodale lors de cette compétition, nous nous sommes aperçu que la méthode de fusion à utiliser dépend de la base données et du problème multimodal à traiter. Ainsi, un questionnement naturel serait de se demander s'il existe une méthode de fusion qui serait optimale pour la majorité des problèmes. La définition très large d'un problème multimodal, comme traitée dans le Chapitre 2 ainsi que le théorème NFL [290] laisse entrevoir la réponse négative à cette question. En revanche, en se restreignant à la tâche plus simple de trouver à quels étages des réseaux de neurones fusionner les différentes représentations, le Chapitre 6 propose un ensemble de méthodes pour automatiser le processus de recherche d'une fusion multimodale optimale.

3.4.3 Perspectives

Plus que des questionnements, nos travaux sur cette compétition ouvre également de nouvelles perspectives d'améliorations et de recherche.

Distillation multimodale Pour pallier le manque de données en audio, des techniques de distillation multimodale ont été proposées (*e.g.* SoundNet [21]). L'idée a également été appliquée dans le cadre de la reconnaissance d'émotions dans des vidéos controllées par Albanie *et al.* [6]. Pour améliorer les performances sur la base de données AFEW, cette approche pourrait également être utilisée. Pour cela, il faudrait en premier lieu collecter un large corpus de vidéos en conditions non controllées présentant des similarités avec AFEW. Le *ResNet*+Moyenne Pondérée décrit dans ce chapitre permettrait par exemple une annotation bruitée de ce corpus. Un modèle multimodal incluant l'audio pourrait alors être entraîné sur ce corpus faiblement annoté en complément des labels de AFEW.

Des modalités supplémentaires ? Bien que la partie audio des vidéos contient peu souvent des paroles, celles-ci peuvent s'avérer être un contexte précieux. Il serait donc intéressant de pouvoir extraire le texte associé à chaque vidéo et le traiter avec des approches d'analyses de sentiment, telles que celle

développées pour l'analyse de tweets [203]. Une autre modalité qui pourrait présenter un intérêt serait la scène complète et plus particulièrement la dynamique corporelle des acteurs. En effet, jusqu'ici les approches ne se sont intéressés qu'aux visages et au son, du fait de l'absence de données annotées, par exemple pour l'expression corporelle. Mais en supposant que nous utilisons l'approche semi-supervisée précédente, il serait possible d'ajouter ce type de modalité.

Aspect temporel et multimodal L'aspect temporel, du fait de la petite taille de la base de données, n'a été que brièvement traité. En supposant disposer d'un corpus plus large, un aspect prometteur pourrait être d'appliquer des mécanismes d'attention multimodale [125], qui consistent à utiliser pour contexte une fusion des différentes modalités. Cela impliquerait également ici d'effectuer la fusion multimodale avant la fusion temporelle.

Chapitre 4

Représentation compacte et interprétable de l'émotion

Table des matières

4.1 Introduction

La capacité d'une machine à reconnaître et analyser les émotions présente un intérêt certain pour différents domaines d'application. Par exemple, comme présenté en introduction générale (*cf.* Chapitre 1), l'analyse automatique des émotions peut permettre l'amélioration des interfaces homme-machine, une aide face au trouble autistique ou encore un outil de marketing. Mais pour cela il est nécessaire de pouvoir identifier ce qu'est une émotion (ou au moins une expression faciale), et donc de posséder un vocabulaire approprié, *i.e.* une capacité à représenter une émotion. Dans ce chapitre, nous nous intéressons plus particulièrement à la représentation des expressions faciales, élément clé de l'expression de l'émotion [90].

Nous avons vu dans la Section 2.2 du Chapitre 2 qu'il existe différentes manières de représenter une émotion. Une approche intuitive consiste à utiliser une annotation discrète, basée sur des termes sémantiques tels que "Joyeux", "Surpris" ou "Triste". Cette approche présente le désavantage d'introduire une certaine ambiguïté, un visage pouvant exprimer à la fois de la surprise et de la joie par exemple. Augmenter le nombre de termes utilisés permet de réduire cette ambiguïté, en utilisant par exemple des émotions mixtes, *e.g.* "joyeusement surpris".

Néanmoins, pour décrire et différencier parfaitement les émotions, il serait nécessaire d'utiliser une infinité de termes sémantiques. Cela conduit naturellement à l'usage d'une représentation continue, telle que l'excitation-plaisir proposée par Russell *et al.* [244]. Une telle représentation continue peut alors être vue comme un espace vectoriel, muni de deux directions ayant la particularité d'être facilement interprétables par l'être humain. Elle présente certes l'avantage d'être beaucoup plus précise, mais présente également un coup important d'annotation. De même, d'autres méthodes d'annotation comme les Action Units impliquent un protocole d'annotation spécifique requérant des annotateurs formés pour celui-ci [268]. Cela motive donc l'utilisation de méthodes automatiques pour extraire de telles représentations de l'émotion et éviter une analyse manuelle systématique.

Or les réseaux de neurones permettent naturellement d'extraire des représentations vectorielles en lien avec une tâche donnée (qui permettent alors d'effectuer des transferts de connaissances sur des tâches liées, comme nous le verrons plus en détail dans le Chapitre 5). On retrouve notamment ce type d'approche pour la reconnaissance des émotions, avec des modèles de plus en plus sophistiqués et des bases de données de plus en plus larges [198, 172, 47, 2, 226]. Ces modèles sont souvent entraînés à partir d'annotations discrètes de l'émotion, bien que certains utilisent des corpus annotés avec des Action Units ou plus récemment avec les notions d'excitation-plaisir.

Il existe donc des réseaux de neurones capable d'extraire automatiquement une représentation de l'expression d'un visage. En revanche, peu de travaux s'intéressent à l'interprétation de ce qu'apprend réellement un réseau de neurones entraîné à prédire des émotions. Nous pouvons tout de même citer une étude de 2015 par Khorrami *et al.* [148], consistant à analyser les représentations internes d'un réseau de neurones convolutif lorsque celui-ci est entraîné à prédire des émotions discrètes. Les auteurs montrent notamment que les représentations apprises par le réseau sont très similaires à des Action Units [90].

Ce chapitre cherche à étudier les liens entre la représentation neuronale latente et la représentation psychologique excitation-plaisir. Nous proposons tout d'abord de réduire la dimensionnalité des représentations utilisées afin de plus facilement visualiser l'espace ainsi créé et d'y exhiber des directions pertinentes. Il s'agit du principe même de nombreuses méthodes de visualisation telles que la PCA [138]. De plus, nous montrons qu'utiliser une représentation de dimension réduite ne conduit qu'à une perte de performance marginale pour la reconnaissance des émotions, en comparaison avec l'usage de représentations beaucoup plus larges. Cela est d'ailleurs en accord avec le critère d'Akaike [5], poussant à utiliser des modèles et des représentations de plus petites tailles afin de potentiellement améliorer la généralisation (et accessoirement réduire le temps de calcul).

Nous proposons par la suite d'utiliser la représentation neuronale des expressions comme une véritable annotation de l'expression sur un large corpus et ainsi de s'en servir pour entraîner un GAN ayant pour but de modifier l'expression faciale. Cette technique permettra de contrôler finement la génération d'expression et de plus facilement interpréter les différentes directions exhibées dans l'espace de représentation, au regard des concepts développés en théorie des émotions.

Enfin, cette méthode de modification de l'expression faciale est l'objet d'une démonstration en ligne [1] permettant de tester directement notre modèle sur n'importe quel visage.

4.2 Apprentissage d'une représentation de l'expression faciale

Nous proposons dans cette partie de détailler tout d'abord une approche neuronale de référence, que nous avons adoptée pour apprendre à prédire des expressions discrètes à partir des images. Puis nous revenons à travers quelques expériences préliminaires sur les liens qui existent entre l'espace excitation-plaisir et les expressions discrètes. Enfin, nous confirmons ces premières intuitions à travers différentes variations de l'approche classique, principalement sur les caractéristiques de la représentation cachée du réseau de neurones et exhibons une représentation spécifique, compacte et performante des expressions faciales.

4.2.1 Entraînement d'un réseau de neurones pour la reconnaissance d'expression faciale

Nous avons déjà évoqué dans les chapitres qui précédent des approches d'apprentissage neuronal pour la reconnaissance de l'expression faciale. Nous proposons ici de détailler une approche neuronale classique pour la prédiction d'émotion discrète. Elle constitue une base de départ des travaux effectués dans ce chapitre.

FIGURE 4.2.1 – Images extraites des trois bases de données que nous utiliserons le plus dans ce chapitre.

Données et pré-traitement Nous avons choisi de travailler avec des images d'expression faciale, ce qui permet de simplifier le problème à résoudre en supprimant la dimension temporelle et de disposer d'un grand nombre de données annotées. En effet, il existe une grande diversité de bases de données d'expression faciale. Nous avons plus particulièrement travaillé sur trois d'entre elles :

— *SFEW* [80] : environ 1800 images de visages dans des conditions pouvant être très bruitées (annotation, occlusion, illumination, *etc.*). Chaque image est annotée avec un classe d'expression parmi sept (colère, dégoût, peur, joie, surprise, tristesse, neutralité).

— *RAF* [172] : environ 30 000 images de visages également dans des conditions pouvant être très bruitées. Nous travaillons uniquement sur les images annotées avec les sept classes d'expressions (soit environ 15 000 images), pour faciliter les comparaisons entre bases de données et avec l'état de l'art. L'annotation est très précise, car chaque image a bénéficié de 40 annotateurs.

1. https://many-fe.noprod-b.kmt.orange.com/

— *AffectNet* [198] : environ 500 000 images, dont 300 000 finalement exploitables (beaucoup d'images sans visage ou contenant des occlusions particulièrement fortes). Chaque image est annotée à la fois avec l'expression discrète (nous nous limitons aux 7 mêmes classes que pour les autres bases) et en excitation-plaisir.

De la même manière que dans le Chapitre 3 et pour faciliter l'entraînement d'un réseau de neurones, nous pré-traitons l'ensemble des images de la même manière, en appliquant tout d'abord un détecteur de visages, puis en les alignant par rapport à des points d'intérêt tels que l'oeil.

Approche de référence Nous avons fait le choix d'entraîner une architecture de réseau de neurones de type ResNet sur AffectNet. Ce choix a été fait de manière arbitraire, bien qu'influencé par les excellents résultats de ce type d'architecture notamment sur des tâches liées au visage [122]. Le nombre d'exemples d'entraînement est relativement important mais nous avons tout de même choisi d'utiliser diverses techniques de régularisation telles que de l'augmentation de données (rotation, changement d'échelle, retournement horizontal) et du cutout (*i.e.* masquer des parties aléatoires de l'image durant l'entraînement) [77].

Nous distinguerons par la suite deux approches : *ResNet-disc*, qui est entraîné à classifier des émotions discrètes et *ResNet-EP*, qui est entraîné à régresser l'excitation-plaisir.

Un élément d'importance lors de l'entraînement de *ResNet-disc* a été de compenser la distribution très inégale des classes. En effet, AffectNet a été collecté automatiquement par ses auteurs et implique donc un biais important, les expressions négatives étant souvent moins présentes dans les photographies publiques que les expressions positives ou neutre. Ainsi, plus de la moitié des exemples de la base de données sont contenus dans les classes joie et neutralité. Nous avons donc choisi de pondérer la fonction de coût du réseau de neurones pour donner plus d'importance aux catégories minoritaires. Ainsi, la fonction de coût L dans le cas de la classification parmi sept classes est une entropie croisée, qui s'écrit :

$$L = \frac{1}{N}\sum_{i=1}^{N} w_{classe}(\mathbf{z_i})\ \mathcal{E}(\mathbf{z_i}, \hat{\mathbf{z}}_\mathbf{i}) \tag{4.2.1}$$

avec $\mathbf{z_i}$ le label associé à la $i^{ème}$ image, $\hat{\mathbf{z}}_\mathbf{i}$ la prédiction estimée pour la $i^{ème}$ image, $\mathcal{E}$ l'entropie croisée, N le nombre d'images dans le batch et $w_{classe}(\mathbf{z_i})$ le poids donné à la $i^{ème}$ image, dépendant du label de l'image.

Plus précisément, nous l'avons défini tel que $w_{classe}(\mathbf{z_i}) = \frac{N_{total}}{N_{z_i}\times 7}$ avec 7 le nombre de classes, N_{total} le nombre total d'images dans AffectNet et N_{z_i} le nombre d'images contenues par la classe y_i. Ce choix se justifie en décomposant L par classe :

$$L = \frac{1}{7}\sum_{k=1}^{7} \mathbb{E}(L|k) \tag{4.2.2}$$

Cela peut alors être développé sous la forme :

$$L = \frac{1}{7}\sum_{k=1}^{7}\sum_{i\in\ classe\ k} \frac{\mathcal{E}(z_i, \hat{z}_i)}{N_k} \tag{4.2.3}$$

$$L = \frac{1}{7}\sum_{k=1}^{7}\sum_{i\in\ classe\ k} \frac{\mathcal{E}(z_i, \hat{z}_i)}{N_k} \tag{4.2.4}$$

$$L = \frac{1}{N}\sum_{i=1}^{N} \frac{N}{7N_{z_i}}\ \mathcal{E}(z_i, \hat{z}_i) \tag{4.2.5}$$

$$\tag{4.2.6}$$

avec N_k le nombre d'éléments dans la classe k.

Dans notre cas, nous souhaitons donner autant d'importance à chaque classe afin d'être performant dans le cas d'une distribution uniforme des classes. Cela revient à se placer dans le cas où chaque classe

contiendrait autant d'éléments, *i.e.* que $\forall k, w_k N_k = \frac{N_{total}}{7}$. Ainsi, il faut alors fixer le poids w_k associé à la classe k tel que $w_k = \frac{N_{total}}{N_j \times 7}$.

Dans le cas de *ResNet-EP*, la fonction de coût est simplement la moyenne des erreurs au carré.

Un apprentissage multi-tâches (classes discrètes et estimation des valeurs excitation-plaisir) a également été implémenté, consistant à ajouter deux perceptrons au lieu d'un à la sortie de la dernière couche cachée du réseau de neurones, le premier classifiant les émotions discrètes et le second régressant l'excitation-plaisir. L'objectif est que la représentation cachée apprise par le réseau de neurones en soit améliorée, devenant plus générale, par le fait qu'il est alors nécessaire d'extraire des informations utiles pour les deux tâches à partir d'une même image. Les bénéfices potentiels d'une approche multi-tâches sont également rappelés dans la dernière partie du Chapitre 2 et une représentation issue de différentes tâches sera étudiée dans le Chapitre 5.

Évaluation de l'approche de référence

Évaluation du *ResNet-disc* De manière à évaluer cette première approche, nous reportons en Table 4.2.1 l'accuracy du *ResNet-disc* (pourcentage de prédictions correctes) sur l'ensemble de validation d'AffectNet en utilisant uniquement sept classes. Cela permet d'observer que malgré un nombre important d'exemples d'entraînement, les modèles avec un plus grand nombre de couches (tels que ResNet-34 et ResNet-50) semblent moins bien généraliser. L'apport de l'augmentation est également clair et s'explique par la grande variabilité des images collectées, que les 300 000 images de l'entraînement ne suffisent peut-être pas à décrire. Cela rejoint les conclusions d'approches très récentes [71], qui proposent de trouver automatiquement la meilleure stratégie d'augmentation des données et obtiennent des gains de performance important.

Architecture	Régularisation	Accuracy en %
ResNet-18	Aucune	57.4
ResNet-18	Augmentation	61.2
ResNet-18	Augmentation + Cutout	61.7
ResNet-34	Augmentation + Cutout	61.0
ResNet-50	Augmentation + Cutout	60.4
AlexNet [198]	Augmentation	57.0

TABLE 4.2.1 – Résultats de l'approche *ResNet-disc* sur l'ensemble de validation d'AffectNet avec des variations d'architecture et de technique de régularisation.

Enfin, comme illustré dans la dernière ligne de la Figure 4.2.1, des occlusions sont régulièrement présentes, masquant des parties clés du visage telles que la bouche. Le fait d'appliquer du cutout peut alors être interprété comme une simulation de diverses occlusions et pourrait expliquer l'amélioration observée. Dans le reste du chapitre, nous utiliserons comme *ResNet-disc* le modèle de ResNet-18 entraîné avec augmentation et cutout.

Nous pouvons préciser que nous avons très récemment entraîné sur la même tâche un PyramidNet-50 [117] (qui a le même nombre de paramètres que le ResNet-18) et obtenu de meilleures performances (63.5%), ce qui souligne encore l'importance du choix de l'architecture.

Évaluation du *ResNet-EP* Nous avons évalué le *ResNet-EP* avec différentes configurations, reportées en Table 4.2.2. Les conclusions à propos des bénéfices d'un apprentissage régularisé sont les mêmes que pour le *ResNet-disc*. Dans le reste du chapitre, nous utiliserons donc comme *ResNet-EP* le modèle de ResNet-18 entraîné avec augmentation et cutout.

Notons qu'une Racine de l'Erreur Moyenne au Carré ou *Root Mean Squared Error* (RMSE) de 0.34 reste un score relativement mauvais, étant donné que les valeurs d'excitation-plaisir sont comprises dans l'intervalle [-1,1]. Cela peut s'expliquer par le bruit d'annotation important, puisque l'accord moyen

entre deux annotateurs (*cf.* dernière ligne de la Table 4.2.2) a une RMSE du même niveau que notre modèle.

Architecture	**Régularisation**	**RMSE Excitation / Plaisir**
ResNet-18	Aucune	0.35 / 0.41
ResNet-18	Augmentation + Cutout	0.34 / 0.36
AlexNet [198]	Augmentation	0.40 / 0.394
Accord entre annotateur [198]	Aucune	0.36 / 0.34

TABLE 4.2.2 – Résultats de l'approche *ResNet-EP* sur l'ensemble de validation d'AffectNet sans et avec régularisation.

Évaluation de l'approche multi-tâche Enfin, nous avons évalué l'approche multi-tâches (classification d'émotion discrète et régression d'excitation-plaisir). Avec une architecture de type ResNet-18, de l'augmentation des données et du cutout, nous obtenons alors une **accuracy de 60.2%** pour la classification d'émotion discrète et une **RMSE de 0.35 / 0.39** pour la régression excitation / plaisir. Ces scores légèrement plus faibles que pour les approches mono-tâche précédentes peuvent notamment s'expliquer par l'utilisation de classifieur et régresseur linéaire, empêchant une projection plus sophistiquée. Nous préférerons donc dans les parties suivantes étudier plus en détail les relations entre les deux tâches (et donc les deux représentations) et laisserons l'utilisation de méthodes mutli-tâches [197] plus sophistiquées au rang des perspectives.

4.2.2 Quelques intuitions sur la représentation des émotions

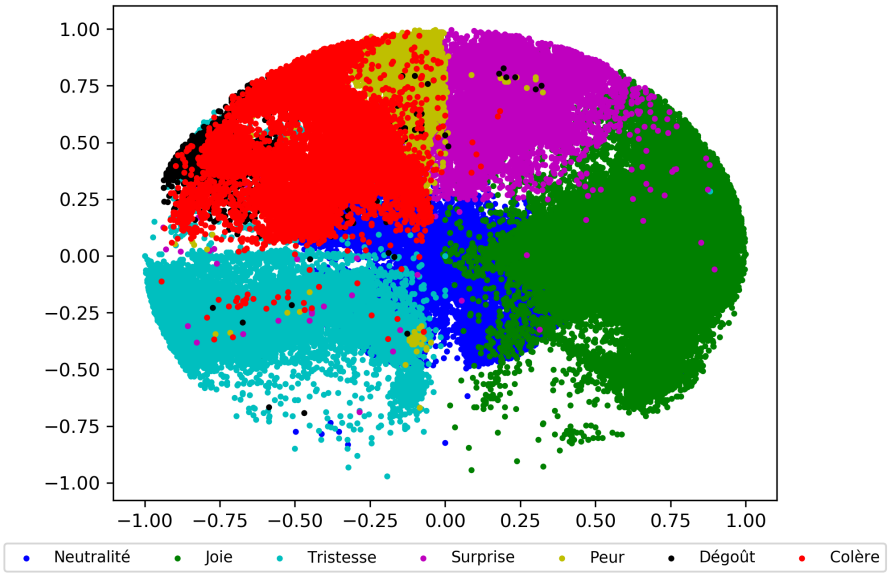


FIGURE 4.2.2 – Représentation dans l'espace excitation-plaisir des visages d'AffectNet [198]. Chaque point représente donc un visage, et est coloré en fonction de son annotation d'émotion discrète.

Nous nous intéressons maintenant à retrouver à travers un ensemble de données la relation entre émotion discrète et excitation-plaisir proposée par Russel [244]. Pour cela, nous utilisons une sous-partie d'AffectNet [198]. Tout d'abord, nous représentons chaque image par un point dans l'espace excitation-plaisir, que nous colorons suivant l'émotion discrète associée à l'image. Par exemple, un visage annoté comme *neutre* et avec des valeurs excitation-plaisir à (0,0) correspondra à un point de couleur bleue situé au centre de la Figure 4.2.2.

Cette première visualisation peut nous conforter dans l'idée qu'il existe une relation forte entre les deux types d'annotations. Néanmoins, cette observation est biaisée par le processus d'annotation même de cet ensemble de données. En effet, les images ont d'abord été annotées avec des labels d'émotions discrètes, puis les valeurs d'excitation-plaisir pour un exemple donné ont été contraintes dans un intervalle prédéfini en fonction de l'émotion discrète. Malgré ce biais, la représentation d'AffectNet s'accorde avec la théorie du circumplex des émotions de Russell [244] (*i.e.* de l'organisation des émotions suivant les concepts excitation-plaisir) et permet également d'illustrer la diversité d'expressions différentes au sein d'une même classe discrète. Enfin, il est intéressant de noter que cet espace à deux dimensions ne semble que très peu efficace pour séparer le dégoût et la colère, ce qui est également en accord avec des observations dans le domaine de la psychologie en faveur de l'utilisation d'une troisième dimension pour obtenir une meilleure séparation des classes [195].

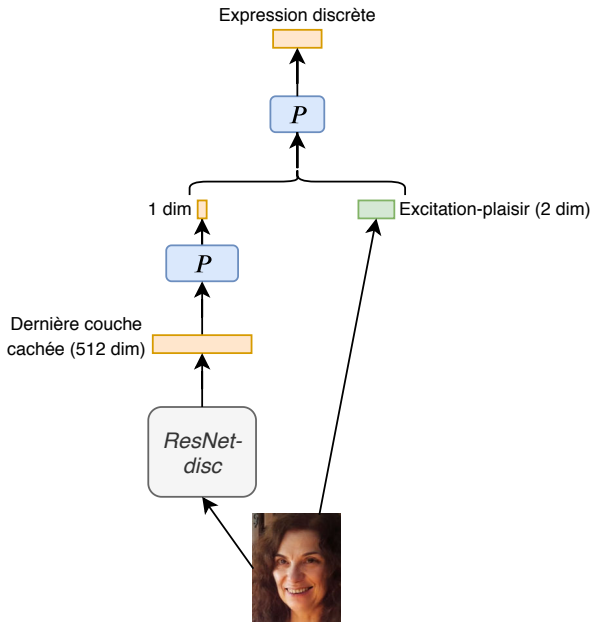


FIGURE 4.2.3 – Approche pour obtenir une troisième dimension de représentation de l'expression. P représente de simples perceptrons. La couleur bleue indique que les paramètres sont mis à jour, la couleur grise qu'ils sont figés. L'exctiation-plaisir est naturellement présente.

En repartant de l'interprétation précédente, une autre manière d'évaluer l'importance des liens entre les deux représentations des émotions est de régresser l'une à partir de l'autre. Ainsi, nous avons choisi d'entraîner sur AffectNet un simple perceptron, avec pour tâche de prédire l'émotion discrète à partir des valeurs excitation-plaisir. Cela conduit à un excellent taux de **83%** de prédictions correctes, avec encore une fois beaucoup de confusion entre le dégoût et la colère (*cf.* Table 4.2.3).

Prédiction	**Label**	Neutralité	Joie	Tristesse	Surprise	Peur	Dégoût	Colère
Neutralité		86	1	1	0	1	0	3
Joie		0	98	0	0	0	0	0
Tristesse		4	0	95	0	0	3	4
Surprise		6	1	1	94	7	0	3
Peur		1	0	1	6	88	1	5
Dégoût		1	0	1	0	2	66	27
Colère		2	0	1	0	2	30	57

TABLE 4.2.3 – Matrice de confusion obtenue pour la prédiction des émotions discrètes à partir de la vérité terrain des valeurs excitation-plaisir (en %). Notons la confusion très importante entre dégoût et colère.

Ce problème de confusion entre les deux classes nous a naturellement amené à vouloir rajouter une dimension à la représentation excitation-plaisir. Pour cela, ne disposant pas d'annotations d'une troisième dimension, nous proposons de nous servir du *ResNet-disc* décrit dans la sous-section précédente. Pour chaque image, nous avons extrait la dernière couche du *ResNet-disc* (déjà entraîné), en considérant celle-ci comme un vecteur de description de dimension 512. Nous avons alors proposé une approche illustrée en Figure 4.2.3. L'idée est d'extraire une dimension supplémentaire de la dernière couche du *ResNet-disc*, puis de concaténer les deux valeurs excitation-plaisir (véritables annotations) avec cette nouvelle dimension. Nous avons alors évalué l'impact de cet ajout en tentant de prédire à partir des trois valeurs obtenues l'expression discrète des visages. Comme l'illustre la Figure 4.2.4, le gain en accuracy apporté par cette troisième dimension est de **3 points**. En renouvelant l'expérience mais en ajoutant cette fois bien plus de dimensions, le gain décroît de manière exponentielle, atteignant **3.6 points** en utilisant 512 dimensions.

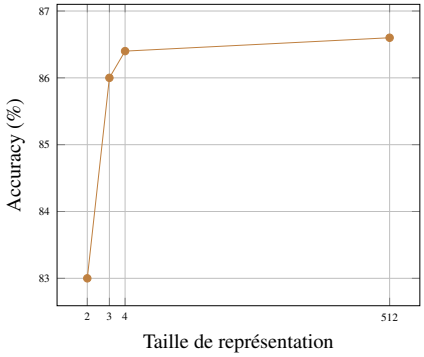


FIGURE 4.2.4 – Évolution de l'accuracy en fonction de la taille de représentation utilisée (en utilisant deux valeurs excitation-plaisir issues de la vérité terrain).

De ces première observations, nous pouvons donc retenir que l'usage d'une représentation compacte de l'expression faciale ne semble que peu dégrader la performance en termes de classification d'émotion discrète. De plus, le gain apporté par l'ajout d'une troisième dimension est très important comparé à l'ajout d'un plus grand nombre de dimensions, ce qui pourrait s'expliquer par le rôle important de cette dimension pour séparer dégoût et colère. Enfin, bien que performantes pour classifier les émotions discrètes et facilement interprétables, nous avons vu que les notions d'excitation-plaisir ne sont pas toujours optimales, notamment pour séparer des concepts tels que le dégoût et la colère. Cela nous amène à la question : pourrions-nous exhiber une représentation continue plus efficace pour différencier les émotions discrètes mais toujours interprétable ?

4.2.3 Apprentissage d'une représentation compacte et performante

Extraction de descripteurs Nous reprenons maintenant le *ResNet* de base décrit dans la première sous-section et nous utilisons sa dernière couche cachée comme une description de dimension 512 de l'expression faciale. Ce *ResNet*, dans le cas où il a été entraîné à la classification d'expressions discrètes, sera nommé *ResNet-disc*. Il est également possible de suivre une procédure identique à l'approche de base, mais en ayant pour tâche de régresser les valeurs d'excitation-plaisir. Dans ce cas, nous nommerons le modèle obtenu *ResNet-EP*.

Obtenir une représentation compacte Nous explorons deux approches. La première, illustrée par la Figure 4.2.5, consiste simplement à appliquer un perceptron **P** sur la description de dimension 512, de manière à obtenir une représentation de dimension k, avec $k << 512$. À partir de ces k descripteurs, un second perceptron P_j doit alors prédire l'expression discrète associée à l'image originale. L'indice j peut être ignoré dans ce paragraphe et représente simplement la base de données utilisée pour l'entraînement. Par la suite, nous utiliserons pour cette méthode la notation $disc_k$.

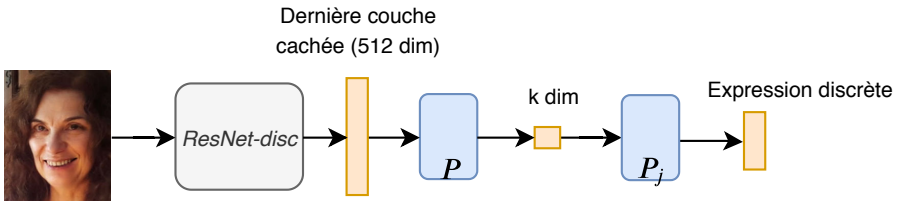


FIGURE 4.2.5 – Approche $disc_k$ pour l'apprentissage d'une représentation k-dimensionnelle de l'expression faciale, à partir d'une base de donnée j.

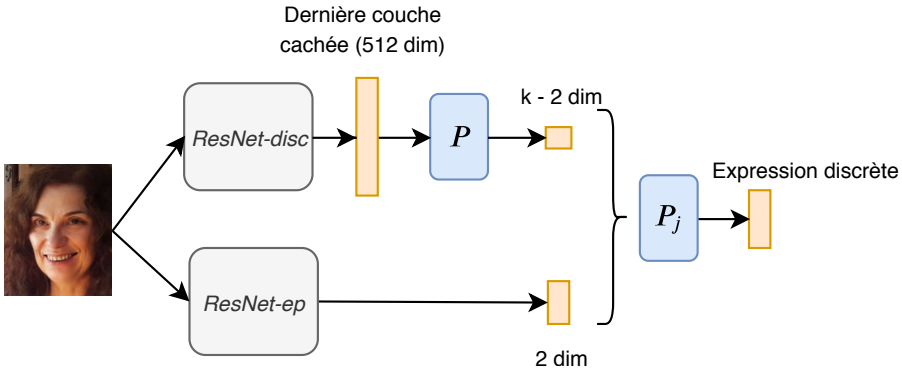


FIGURE 4.2.6 – Approche ep_k pour l'apprentissage d'une représentation k-dimensionnelle de l'expression faciale et contenant des valeurs estimées d'excitation-plaisir, à partir d'une base de données j.

La seconde approche consiste à considérer en entrée à la fois les 512 descripteurs issus du *ResNet-disc* et les valeurs excitation-plaisir estimées par le *ResNet-EP*. Le système de la Figure 4.2.6 permet alors d'entraîner un premier perceptron **P** à compresser les 512 descripteurs en un vecteur de dimension k-2. Celui-ci est alors concaténé avec les deux valeurs d'excitation-plaisir, permettant d'obtenir une représentation de dimension k. De même que pour $disc_k$, un second perceptron $\mathbf{P_j}$ permet ensuite de prédire l'expression discrète associée à l'image originale. Par la suite, nous utiliserons pour cette méthode la notation ep_k.

Apprentissage simultané sur plusieurs bases de données Comme décrit en introduction de ce chapitre, il existe diverses bases de données contenant des expressions faciales annotées de manière discrète. Chacune présentant certains avantages et une certaine inconsistance d'annotation avec les autres, il serait intéressant d'apprendre nos modèles sur l'ensemble des bases et non sur une seule. Pour cela, nous considérons que chaque base de données possèdent un indice j et qu'il existe dans les méthodes décrites précédemment un perceptron $\mathbf{P_j}$ associé à chaque base de données. Lors de l'entraînement, si l'image en entrée provient de la base de données j, seul le classifieur $\mathbf{P_j}$ sera utilisé et mis à jour. En revanche, les poids de P sont communs pour toutes les bases de données, permettant d'apprendre une représentation plus générale.

Une approche naïve consisterait alors à simplement apprendre sur le regroupement de toutes les bases. Il se trouve que les tailles des bases de données peuvent être particulièrement différentes. De plus, les bases de données regroupant un grand nombre d'éléments présentent souvent moins d'annotateurs et donc des annotations plus bruitées (*e.g.* AffectNet [198] contient un demi-million d'images, chacune annotée par moins de trois personnes, tandis que RAF [172] contient environ 15 000 images, chacune annotée par au moins 40 personnes). Ainsi, nous avons choisi de pondérer les images provenant de diverses bases de données par rapport à la taille de celles-ci, de manière à pénaliser les exemples provenant des bases de données les plus conséquentes. Cela revient à ré-écrire l'Equation 4.2.1 en introduisant une

pondération supplémentaire $w_{base}(\mathbf{x_i})$:

$$L = \frac{1}{N}\sum_{i=1}^{N} w_{base}(\mathbf{x_i}) w^i_{classe}\, E(\mathbf{y^i}, \hat{\mathbf{y}}^\mathbf{i}) \tag{4.2.7}$$

tel que :

$$w_{base}(\mathbf{x_i}) = \frac{1}{logN_{x_i \in \ base\ j}} \tag{4.2.8}$$

avec N_j le nombre d'éléments dans la $j^{ème}$ base de données. L'usage du logarithme permet d'éviter une pondération extrêmement faible des éléments appartenant aux bases les plus grandes et ainsi de s'assurer que ceux-ci aient tout de même un impact.

Méthode d'évaluation De manière à pouvoir par la suite correctement évaluer les différentes approches, nous utiliserons plusieurs métriques. En premier lieu, nous parlerons d'accuracy pour désigner le taux d'éléments correctement classifiés parmi tous les éléments. L'accuracy est la métrique usuellement utilisée pour se comparer à l'état de l'art en reconnaissance d'expression discrète. Nous faisons également le choix de calculer un Macro F1-Score, qui permet de combiner rappel et précision (F1-Score) et d'également de donner autant d'importance à chacune des classes (macro), peu importe la distribution de celles-ci. Ainsi celui-ci s'écrit :

$$F_{1macro} = \frac{1}{N_c}\sum_{i}^{N_c} F_{1i}\ ; \quad F_{1i} = 2\frac{prec_i \cdot rec_i}{prec_i + rec_i}\ ; \quad prec_i = \frac{tp_i}{tp_i + fp_i}\ ; \quad rec_i = \frac{tp_i}{tp_i + fn_i} \tag{4.2.9}$$

où i est le numéro de la classe ; $prec_i$, rec_i et F_{1i} sont respectivement la précision, le rappel et le F1-score pour les éléments de la classe i ; N_c est le nombre de classes ; et tp, fp and fn sont respectivement les taux de vrais positifs, de faux positifs et de faux négatifs.

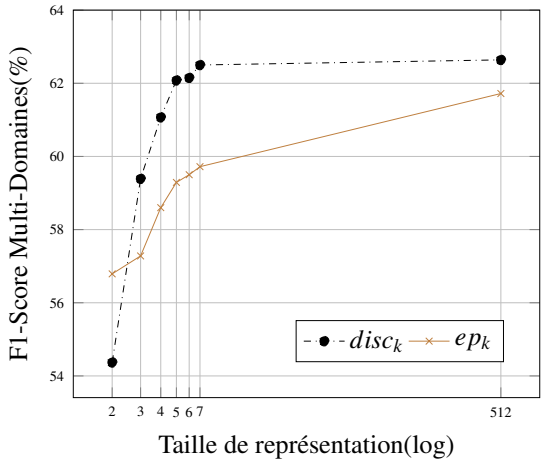
Quand cela nous est possible, nous fournissons le score moyen et l'écart-type du score du modèle, obtenus avec 10 initialisations aléatoires des poids.

Enfin, les scores reportés peuvent l'être sur l'ensemble de test d'une base de données spécifique telle que SFEW [80], RAF [172] ou AffectNet [198], mais également sur un ensemble de test regroupant les trois. Nous calculerons un F1-Score Multi-Domaines, qui est une moyenne pondérée des Macro F1-Scores sur chacune des bases. La pondération de chaque score est l'inverse de la taille de l'ensemble de test associé.

4.2.4 Évaluation de la représentation

Impact de la taille de la représentation sur les performances Nous nous intéressons ici à l'impact de la taille k de la représentation utilisée par les deux méthodes décrites précédemment ($disc_k$ et ep_k). Le but est d'étudier l'intuition développée précédemment : y a-t-il une progression importante en termes de performance lors de l'ajout d'une troisième dimension (k=3) comparé à l'ajout d'un grand nombre de dimensions ? Pour cela, nous avons évalué les méthodes $disc_k$ et ep_k pour k=2 à k=7, ainsi que pour k=512. L'allure générale des deux courbes en Figure 4.2.7 permet une première analyse de notre intuition. En effet, pour la méthode $disc_k$, en noir, le passage de k=2 à k=3 représente ici un gain de performance d'environ 4%, tandis que le passage de k=3 à k=4 ne représente plus que de 2% et enfin le passage de k=4 à k=512 n'apporte lui aussi que 2% de gain en performance. Ainsi, nous pouvons observer que le gain en performance décroît énormément en faisant croître k. De plus, l'évolution entre k=2 et k=3 est la plus importante.

En revanche, nous n'observons pas les mêmes résultats pour la méthode ep_k. Cela peut s'expliquer par le fait que les deux valeurs d'excitation-plaisir estimées qui sont imposées dans la représentation sont très bruitées et implique donc une baisse importante de performance. La valeur finale (pour k=512) est inattendue, puisque les méthodes $disc_k$ et ep_k devraient être alors quasiment équivalentes. Cela revient



Base de données	Représentation	F1 Score
Affect -Net	$disc_3$	**58.1 ± 0.5**
	$disc_2$	52.1 ± 0.4
	ep_3	55.6 ± 0.5
	ep_2	55.8 ± 0.0
SFEW	$disc_3$	**34,1 ± 1.0**
	$disc_2$	28.0 ± 0.8
	ep_3	30.2 ± 0.8
	ep_2	33.3 ± 0.1
RAF	$disc_3$	**64.4 ± 0.5**
	$disc_2$	60.6 ± 1.9
	ep_3	63.0 ± 0.9
	ep_2	61.2 ± 0.2

FIGURE 4.2.7 – Évaluation des méthodes $disc_k$ et ep_k pour différentes valeurs de k.

à considérer que l'information apportée par les valeurs d'excitation-plaisir estimées est particulièrement mauvaise et dégrade la performance. En effet, ep_7 devrait atteindre au moins des performances similaires à $disc_5$, car dans les deux cas la représentation possède 5 degrés de liberté. Cet écart important pourrait être expliqué notamment par l'évaluation conjointe sur trois bases de données dont deux inconnues de l'estimateur d'excitation-plaisir, qui ne généraliserait alors que peu d'un domaine à l'autre. De plus, ces deux bases "inconnues" (SFEW et RAF) ont été construites pour évaluer la classification d'expression discrètes et présentent donc des expressions relativement marquées, qui ne représentent qu'une faible partie de l'espace excitation-plaisir, rendant celui-ci peut-être moins pertinent.

Comparaison avec l'état de l'art Nous proposons maintenant d'évaluer l'intérêt de notre représentation $disc_3$, en la comparant avec l'état de l'art. Il est important de noter que les méthodes proposées dans la littérature sont souvent spécialisées pour une base de données en particulier. Dans notre cas, nous n'avons introduit aucune méthode particulière et avons de plus utilisé une approche multi-domaine.

La Table 4.2.4 reporte donc des résultats obtenus sur les trois bases de données utilisées dans ce chapitre. La seconde colonne nous permet d'observer que la taille de représentation utilisée par les modèles de l'état de l'art (*i.e.* la taille de la dernière couche cachée) est souvent très importante. Ainsi, bien que $disc_3$ soit dépassé par des approches telles que "Covariance Pooling" (agrégation des cartes de descripteurs en modélisant des relations d'ordre deux) ou "Deep Locality Preserving" (fonction de coût permettant de mieux séparer les éléments des différentes classes), sa performance reste compétitive en considérant que celui-ci a été appris sur plusieurs bases et avec une représentation très compacte. De plus, il est intéressant de noter que des approches proposant des modèles compactes [163] sur RAF sont dépassées par notre approche, à nombre de paramètres équivalents (environ 2 millions de paramètres dans les 2 cas). Cela nous amène à deux conclusions : premièrement, il est pertinent d'utiliser des représentations cachées de faible dimension pour prédire des expressions faciales discrètes, car trois dimensions suffisent à obtenir des résultats compétitifs. Deuxièmement, le fait d'utiliser des représentations de dimension 2000 sur des bases de données pouvant compter moins de 1000 éléments risquent d'entraîner une mauvaise généralisation et notamment une tendance à mieux exploiter les biais inhérents à chaque base de données plutôt que d'apprendre des expressions faciales.

Effets du changement de domaine Nous étudions maintenant les effets qu'a le changement de domaine (entraînement d'un modèle sur une base de données et évaluation sur une autre) sur la performance. Ainsi, nous détaillons à travers la Table 4.2.5 les résultats obtenus par chacun des classifieurs P_j non seulement sur l'ensemble de test de la base de données j mais également sur les ensembles de tests des 2 autres bases de données.

Une observation évidente est l'effet souvent catastrophique produit par le changement de domaine,

	Taille de représentation	RAF [172]	SFEW [80]	AffectNet [198]
Covariance Pooling [2]	2000	79.4	-	-
	512	-	58.1	-
DLP [172]	2000	74.2	51.0	-
Center Loss [289]	2000	72.87	-	-
Compact Model [163]	64	67.6	-	-
VGGFace[172]	2000	58.2	-	-
Transfer Learning [209]	4096	-	48.5	-
EmotiW 2015 [150]	3072	-	52.5	-
$disc_3$	3	68.9	44.7	58.2
Approche de base	512	71.7	48.7	61.7

TABLE 4.2.4 – Performance de notre approche en comparaison avec des méthodes de l'état de l'art. Nous considérons également la taille de représentation utilisée, en considérant celle-ci comme la dimensionnalité de la dernière couche cachée. Pour RAF, la performance est l'accuracy moyenne par classe, tandis que pour SFEW et AffectNet, il s'agit de l'accuracy.

		Évaluation sur		
		AffectNet	SFEW	RAF
Entraînement sur	AffectNet	**58.1** *(± 0.5)*	27.6 *(± 2.6)*	53.8 *(± 0.6)*
	SFEW	35.1 *(± 2.1)*	**34.1** *(± 1.0)*	47.3 *(± 1.2)*
	RAF	51.8 *(± 0.4)*	31.5 *(± 1.7)*	**64.4** *(± 0.6)*

TABLE 4.2.5 – Évaluation multi-domaine des trois classifieurs. Les résultats sont reportés en Macro F1-Score pour l'ensemble des bases (ce qui explique la différence avec les résultats reportés dans la Table 4.2.4.)

qui s'explique à la fois par des natures d'images différentes, mais également par l'inconsistance d'annotation entre différentes bases de données (*cf.* également la Figure 4.3.1). Cette inconsistance implique également de remettre en question les protocoles d'évaluation classiques dans le domaine des expressions faciales, consistant à entraîner et évaluer sur une seule base de données. En effet, les modèles appris, lorsqu'ils seront confrontés au monde réel, risquent de ne pas généraliser et de produire des résultats très éloignés de ceux obtenus sur une base particulière. Une piste d'amélioration récente proposée dans [312] serait d'apprendre à concilier les différentes annotations, mais au prix d'une méthode relativement coûteuse en temps de calcul.

Enfin, nous pouvons noter que les résultats obtenus sur SFEW sont systématiquement bas et avec un écart-type important. Cela s'explique par le nombre faible d'exemples et le bruit d'annotation important de cette base. Cela se confirme en observant les performances du classifieur entraîné sur RAF [172] : celui-ci généralise particulièrement bien d'une base de données à l'autre. Cette hypothèse rejoint également celle des auteurs de RAF [172], qui expliquent disposer d'une meilleure qualité d'annotation dans le sens où chaque label obtenu réside du consensus d'un grand nombre d'annotateurs, permettant de limiter le bruit dû à la subjectivité.

4.3 Analyse et génération d'expressions faciales

La méthode utilisée précédemment nous a permis de créer une représentation compacte et relativement performante sur différentes bases de données, inspirée d'une représentation excitation-plaisir-dominance proposée en psychologie. Néanmoins, rien ne garantie que les trois dimensions exhibées par la méthode $disc_3$ soient liées à ces trois dimensions psychologiques, puisque notre modèle a été entraîné sans contrainte. C'est pourquoi dans cette partie nous cherchons à explorer l'espace de représentation appris. Du fait de la faible dimensionnalité, il est alors possible d'effectuer certaines visualisations des

représentations et du processus de classification. Mais aussi d'exhiber des directions particulières dans cet espace de représentation et les visualiser dans l'espace image, en utilisant une approche générative adverse.

4.3.1 Visualisations préliminaires

Pour mieux comprendre la classification effectuée par nos modèles et le placement des frontières de classification dans l'espace créé par le réseau de neurones, nous proposons d'échantillonner un grand nombre de points de notre espace, puis d'utiliser chacun des $\mathbf{P_j}$ pour classer ces points. Ainsi, une carte dense des émotions discrètes prédites par chacun des $\mathbf{P_j}$ dans un espace à deux ou trois dimensions peut être construite. Bien sûr, certains points échantillonnés (notamment dans les valeurs extrêmes) ne correspondent pas à des expressions réellement observées dans la réalité.

Selon le nombre k de dimensions utilisées, la carte est construite différemment. Dans le cas de $disc_2$ (et ep_2), nous avons simplement utilisé des échantillons consistant en deux coordonnées, comprises entre les valeurs minimales et maximales observées dans les images réelles d'AffectNet. Pour $disc_3$, nous avons effectué dans un premier temps effectué des visualisations en trois dimensions. Mais pour permettre également une visualisation plus simple nous avons également proposé de modifier $disc_3$ en $disc_{3norme}$, en contraignant pendant l'entraînement toutes les coordonnées de la représentation à appartenir à la surface de la sphère unité (autrement dit, à réduire le rayon à 1 en coordonnées sphériques). Cela réduit les performances d'environ 2 points mais permet également une visualisation simple.

La Figure 4.3.1 montre les résultats obtenus pour chacune des méthodes de visualisation (pour $disc_{3norme}$, ep_2 et $disc_2$ respectivement d'en haut à en bas). Chaque point d'une image a pour coordonnées celle de la représentation compacte associée, *i.e.* les coordonnées sphériques pour $disc_{3norme}$ (ϕ et θ car le rayon est unitaire), les valeurs d'excitation et plaisir pour ep_2 (comprises entre -1 et 1) et deux valeurs non bornées pour $disc_2$. De plus, chacun de ces points est coloré suivant la classe prédite par le classifieur $\mathbf{P_j}$. Enfin, nous indiquons le F1-score par classe au sein de chacune des zones de couleur.

Tout d'abord, à partir des lignes de la Figure 4.3.1, nous pouvons observer que chacune des méthodes produit des représentations possédant des organisations globalement similaires d'une base de données à une autre. En effet, chacun des trois classifieurs présentent une organisation des classes discrètes très similaires, indépendamment de la base de données sur laquelle il a été entraîné. Cela peut être interprété comme une preuve de robustesse de la part de la représentation apprise, comparée aux annotations classiques.

Un autre point intéressant est que la classe neutre est souvent placée au centre de la carte et par conséquent est donc voisine de toutes les autres classes. Cela rejoint la définition de cette classe, qui pourrait être vue comme une émotion avec une intensité extrêmement faible, voire une absence d'intensité.

Néanmoins, certains changements, d'un classifieur à l'autre (*i.e.* d'une colonne à l'autre), sont visibles. Plus particulièrement, dans le cas de SFEW (deuxième colonne), la peur (violet) et le dégoût (marron) présentent des aires extrêmement réduites, ce qui n'est pas le cas pour les autres colonnes. Cela s'explique par l'annotation très particulière de cette base et par la présence faible de ces classes dans l'ensemble d'entraînement (la mauvaise performance sur ces classes pour SFEW le confirme d'ailleurs). Cela vient confirmer l'inconsistence des annotations entre les différentes bases de données d'expression faciale et souligne la nécessité de s'intéresser à des évaluations à travers différents domaines.

Nous nous intéressons maintenant aux variations entre les différentes méthodes (*i.e.* entre les différentes lignes) pour une base de données (*i.e.* colonne) particulière. Le classifieur de la méthode ep_2 (seconde ligne), qui correspond finalement à l'excitation-plaisir estimée par un modèle appris sur AffectNet, présente une organisation similaire à celle d'AffectNet représentée en début de chapitre. De plus, étant donné que la majorité d'AffectNet contient des exemples ayant une valeur d'excitation positive, le classifieur a tendance à ne pas paver entièrement l'espace, expliquant alors les aires "allongées" vers le bas des trois classes tristesse, neutralité et joie (resp. vert, bleu et orange). Au contraire, la méthode $disc_k$ échappe indirectement à cette contrainte et permet, comme évoqué précédemment, de placer le neutre au centre des autres classes.

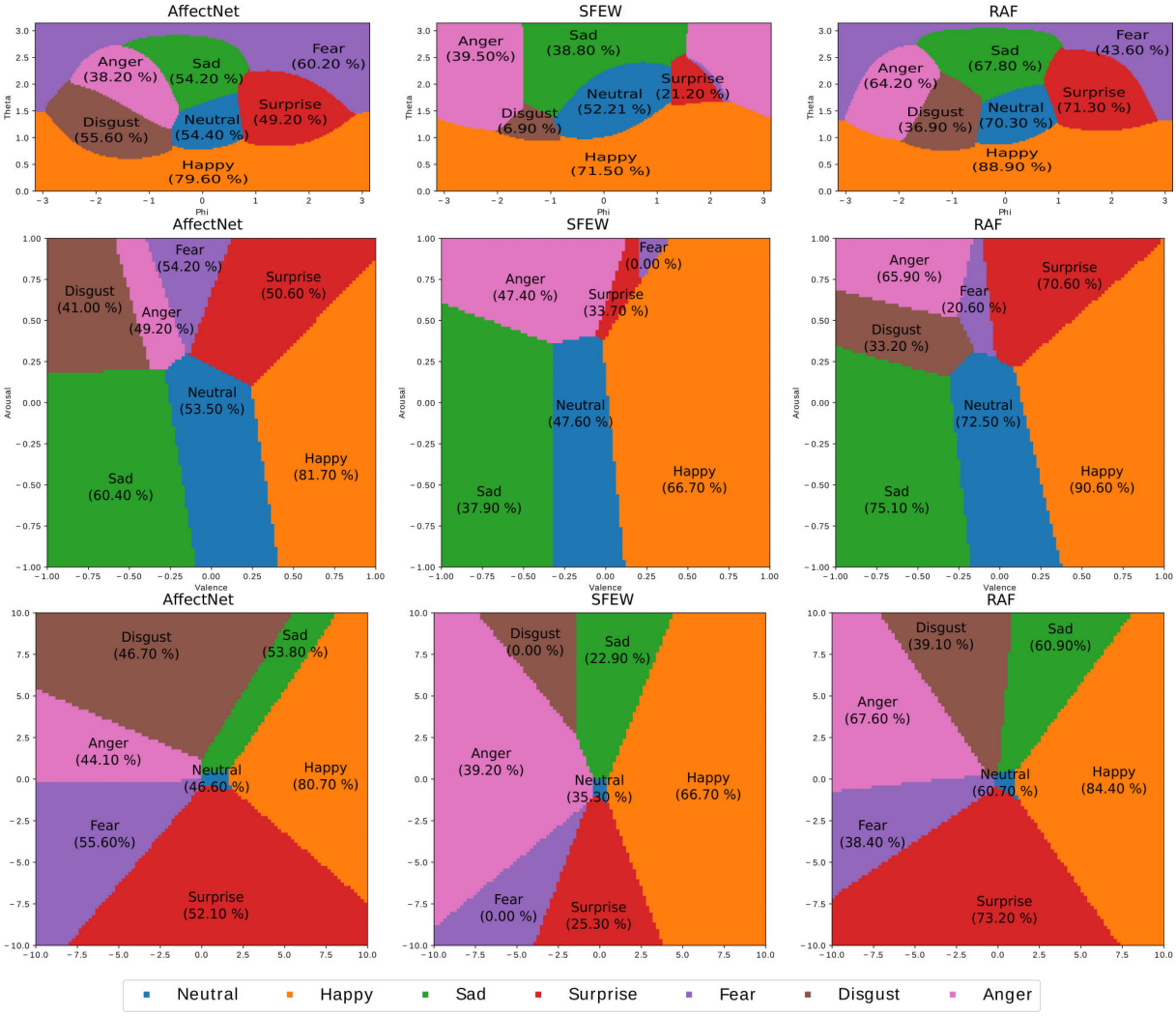


FIGURE 4.3.1 – Visualisation des représentations obtenues avec $disc_{3norme}$, ep_2 et $disc_2$ (resp. de la première à la troisième ligne). Les colonnes correspondent à l'indice du classifieur $\mathbf{P_j}$, *i.e.* la base de données utilisée pour l'entraîner.

Enfin, la similarité entre les représentations $disc_2$ et ep_2 est importante. En effet, en observant l'ordre des classes autour de la zone neutre, nous retrouvons la même séquence pour les deux méthodes : joie, surprise, peur, colère, dégoût, tristesse (dans le sens horaire pour $disc_2$ et antihoraire ep_2). Cette notion de proximité entre classes, également définie par les psychologues, souligne qu'il existe une structure implicitement proche de l'excitation-plaisir dans la représentation apprise par la méthode $disc_2$. Ainsi, un CNN, de par la nature statistique des données, peut aboutir, sans utiliser une supervision directe (*i.e.* en utilisant uniquement des émotions discrètes), à une représentation interne ayant une structure similaire aux concepts développés en psychologie.

4.3.2 Apprentissage d'un modèle de modification de l'expression

Les visualisations précédentes ont permis de développer l'intuition que la représentation $disc_3$ possédait des liens avec les dimensions psychologiques. De plus, celle-ci semble stable d'une base de données à l'autre. Nous proposons dans cette sous-partie de nous intéresser aux caractéristiques de cette représentation quand elle est projetée dans l'espace image. Pour cela, nous proposons d'utiliser $disc_3$ comme un outil d'annotation à faible coût, en extrayant cette représentation pour un large corpus de visages. Cela permet alors l'entraînement d'une méthode de génération d'expressions faciales, amenant à l'étude de l'impact des variations de la représentation $disc_3$ dans l'espace image. Plus précisément, nous étudions

comment le contrôle de la génération d'expression est rendu possible par l'identification de directions au sein de $disc_3$, permettant de retrouver certaines notions de psychologie.

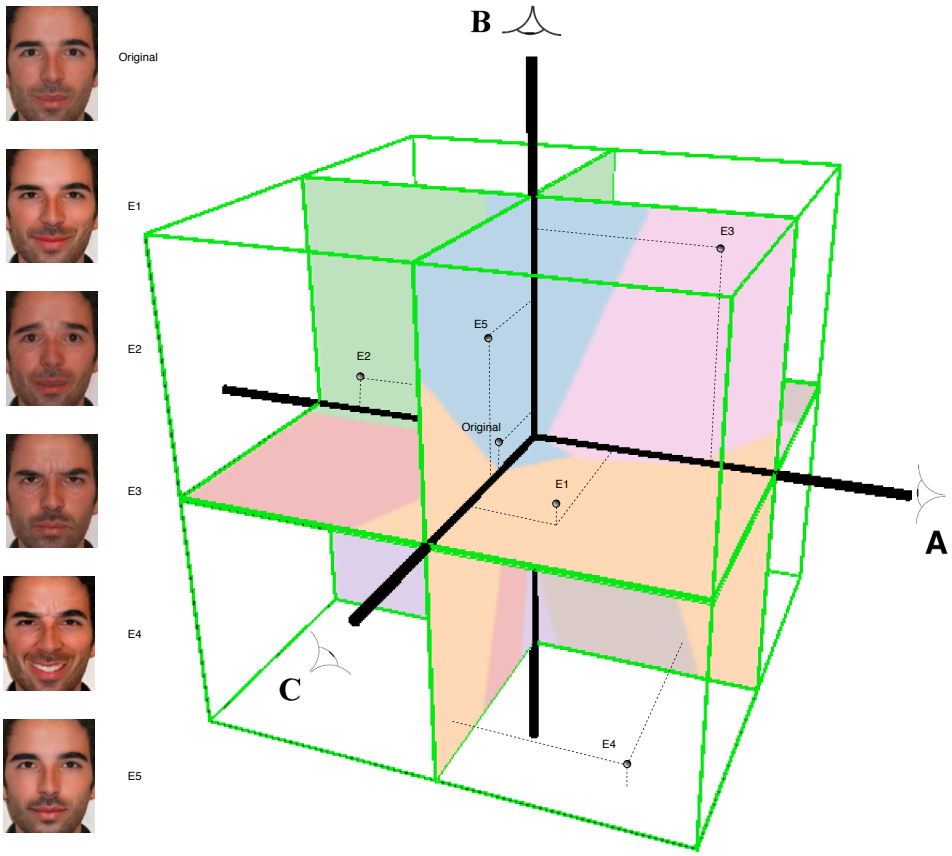


FIGURE 4.3.2 – Description de 3 plans dans l'espace $disc_3$ et des expressions associées.

Annotation continue et à faible coût avec $disc_3$ Nous reprenons la méthode $disc_3$, mais nous considérons que **P** est désormais un perceptron non-linéaire, composé d'une couche entièrement connectée et d'une activation tangente hyperbolique, permettant de contraindre les coordonnées entre -1 et 1. Le modèle ainsi appris permet d'extraire pour un visage donné une représentation tri-dimensionnelle de son expression. Ainsi, une fois l'apprentissage effectué, cela nous permet d'annoter à faible coût de manière continue un large corpus de visages. Nous disposons ainsi, comme illustré en Figure 4.3.2, d'un espace de dimension 3 permettant à la fois de positionner des expressions discrètes et d'exhiber une certaine continuité.

StarGAN Nous souhaitons maintenant être capable de modifier l'expression d'un visage donné, *i.e.* être capable de changer son positionnement dans l'espace $disc_3$. Pour résoudre ce type de modification, d'image à image, nous avons choisi de modifier l'approche existante StarGAN [66]. Cette approche générative adverse prend en entrée de son générateur un visage et une expression discrète cible et a déjà permis d'obtenir d'excellents résultats dans le domaine de la génération d'expression. Le modèle général est composé d'un discriminateur D et d'un générateur G, et est entraîné à minimiser les objectifs concurrents de ses deux sous-modèles. Notre modification consiste simplement à modifier cette ensemble de manière à pouvoir prendre en entrée des annotations continues et non une expression discrète.

Premièrement, la fonction de coût adverse, qui a pour but de rendre les images générées non distinguables des images réelles, reste inchangée :

$$L_{adv} = \mathbb{E}_x[logD(\mathbf{x})] + \mathbb{E}_{\mathbf{x},\mathbf{r}}[1 - logD((G(\mathbf{x},\mathbf{r})))] \tag{4.3.1}$$

avec **x** l'image réelle en entrée et **r** la représentation de l'expression. G et D ont respectivement pour objectif de minimiser et maximiser cette fonction de coût.

Ensuite, le discriminateur du StarGAN possède une fonction de coût de classification, qui est ici remplacée par une fonction de coût de *regression*. Elle est composée de deux termes. Le premier, L_{reg}^{real}, force D à régresser correctement l'expression continue associée à l'image d'origine. Tandis que le second terme, L_{reg}^{fake}, force *G* à générer des expressions faciales possédant une représentation continue proche de l'annotation **r** donnée en entrée. Plus formellement, notre fonction d'erreur est une moyenne des erreurs au carré :

$$L_{reg}^{real} = \mathbb{E}_{\mathbf{x},r}[D(\mathbf{x}) - \mathbf{r}]^2 \text{ and } L_{reg}^{fake} = \mathbb{E}_{\mathbf{x},\mathbf{r}}[D(G(\mathbf{x},\mathbf{r})) - \mathbf{r}]^2 \tag{4.3.2}$$

La fonction de coût de *reconstruction*, L_{rec}, permet de s'assurer que les visages générés conservent les informations non liés à l'expression (telles que l'identité, l'orientation, *etc.*). Elle est définie par :

$$L_{rec} = \mathbb{E}_{\mathbf{x},\mathbf{r_1},\mathbf{r_2}}[||\mathbf{x} - G(G(\mathbf{x},\mathbf{r_2}),\mathbf{r_1})||_1] \tag{4.3.3}$$

avec $\mathbf{r_1}$ la représentation de l'expression faciale du visage original et $\mathbf{r_2}$ celle du visage que le générateur a pour but de générer.

Enfin, nous pouvons écrire les fonctions générales de coût de D et G en utilisant des termes de pondérations λ relatifs à chaque critère et choisis par validation croisée.

$$L_D = -L_{adv} + \lambda_{reg} L_{reg}^{real} \tag{4.3.4}$$

$$L_G = L_{adv} + \lambda_{reg} L_{reg}^{fake} + \lambda_{rec} L_{rec} \tag{4.3.5}$$

Contrôler la génération des expressions faciales L'approche précédente nous assure donc de pouvoir déplacer l'expression faciale d'un visage donné dans l'espace que nous avons créé. Il est maintenant important de pouvoir contrôler ces déplacements et leurs conséquences. C'est pourquoi nous proposons, grâce à la structure de notre espace, de répondre à plusieurs scénarios. Le premier est de considérer que nous souhaitons générer autant d'expressions faciales que possibles et d'ainsi "paver l'espace", comme effectué lors des visualisations préliminaires. Mais il est également intéressant de pouvoir générer des expressions discrètes comme traditionnellement effectué par les autres approches telles que StarGAN. Enfin, puisque nous avons montré qu'il existait certaines similarités entre l'espace de représentation que nous utilisons et les dimensions psychologiques excitation-plaisir, il serait intéressant de pouvoir retrouver ces axes au sein de notre espace, de manière à ensuite pouvoir en étudier les images générées.

Paver l'espace Dans ce cas, il suffit de générer des expressions pour un grand nombre de coordonnées dans notre espace, obtenues en échantillonnant celui-ci comme dans les visualisations préliminaires.

Expressions discrètes Pour générer des expressions discrètes telles que colère, joie ou tristesse, il est nécessaire de définir une coordonnée précise dans notre espace. Pour cela, nous avons utilisé une partie des visages d'AffectNet avec leur label discret et en avons extrait également notre représentation $disc_3$. Il nous a alors été possible de calculer les barycentres de chacune des classes dans l'espace $disc_3$. De manière plus formelle, nous avons calculé les coordonnées des barycentres telles que :

$$\mathbf{r^i} = \sum_{k \in C_{discrete}^i} \frac{\mathbf{r_k}}{\#C_{discrete}^i} \tag{4.3.6}$$

où $\mathbf{r^i}$ les coordonnées du barycentre de la classe *i*, $C_{discrete}^i$ l'ensemble des éléments de la classe *i*, et $\mathbf{r_k}$ les coordonnées d'un visage d'indice *k*.

Excitation, plaisir, dominance Pour pouvoir également exhiber les axes d'excitation et de plaisir, nous avons utilisé un sous-ensemble d'AffectNet. Ainsi, pour chaque visage nous disposons de nos coordonnées en trois dimensions et des coordonnées en excitation-plaisir issues d'AffectNet. Alors si $\mathbf{v_e}$ et $\mathbf{v_p}$ sont les vecteurs directeurs de l'espace excitation-plaisir, nous pouvons exprimer leurs coordonnées dans l'espace $disc_k$ par simple régression :

$$\min_{\mathbf{v_e},\mathbf{v_p}} \sum_k ||\mathbf{ep_k} - [\mathbf{v_e};\mathbf{v_p}] \times \mathbf{r_k}||^2 \tag{4.3.7}$$

avec $\mathbf{ep_k}$ les valeurs annotées d'excitation et de plaisir du visage k.
Cette régression est validée par la Table 4.3.1, qui permet d'obtenir une erreur relativement faible.

	Approche de base	Humain	[175]	$disc_k$
RMSE Excitation	0.40	0.36	0.41	0.34
RMSE Plaisir	0.394	0.34	0.45	0.36

TABLE 4.3.1 – Racine de l'erreur moyenne au carré lors de l'estimation de l'excitation et du plaisir. L'approche de base correspond à un réseau convolutif AlexNet entraîné par les auteurs de AffectNet [198], l'approche humaine correspond à l'accord inter-annotateur sur AffectNet, [175] correspond à une approche récente entraînée sur AffectNet et $disc_k$ correspond à la régression des deux valeurs à partir de notre représentation.

Enfin, une fois les deux premiers vecteurs régressés, il nous est possible d'exhiber une troisième direction orthogonale dans l'espace, que nous calculons simplement par produit vectoriel : $i_d = i_e \otimes i_p$. Nous montrerons par la suite que cette direction présente d'importantes similarités avec la notion de dominance en psychologie. Nous pouvons donc générer une nouvelle expression suivant ces différentes directions simplement en régressant les coordonnées de l'expression souhaitée à partir de l'espace excitation-plaisir-dominance associé.

4.3.3 Évaluation de la représentation $disc_3$ pour la génération d'expressions faciales

Implémentation Nous cherchons maintenant à analyser et évaluer la représentation $disc_3$ pour la génération d'expressions faciales. Pour cela, nous utiliserons uniquement AffectNet avec 7 classes d'expressions discrètes, les annotations d'excitation-plaisir et les annotations en trois dimensions $disc_3$. De même que précédemment, les visages sont détectés, alignés et retaillés en 256 par 256 (et 3 composantes couleurs RGB).

Pour pouvoir mesurer les bénéfices de notre approche face à l'utilisation d'autres représentations, nous avons fait le choix d'évaluer trois méthodes de génération d'expression, toutes basées sur l'architecture StarGAN mais utilisant comme annotations des représentations différentes :

— *Discrete*, l'approche utilisée par les auteurs de StarGAN (sans multi-domaine), qui utilise des expressions discrètes

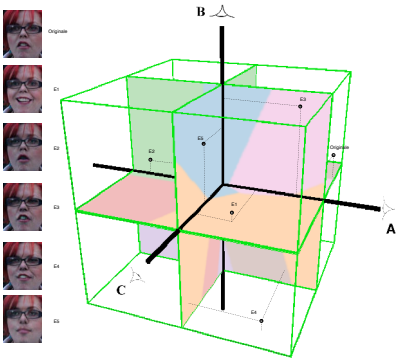
— *EP*, un starGAN modifié comme dans la sous-partie précédente mais qui utilise les valeurs d'excitation-plaisir

— $Sdisc_3$, un starGAN modifié décrit dans la sous-partie précédente et utilisant donc $disc_3$

En pratique, nous effectuons le même entraînement pour chaque modèle, avec une taille de batch de 16, un taux d'apprentissage de 1e-4 avec une décroissance exponentielle de 0.996 par epoch. Concernant les architectures de G et D, elles sont similaires à celles proposées dans StarGAN [66]. Dans le cas de EP et de $Sdisc_3$, λ_{reg} le poids multipliant la fonction de coût associée à la régression est fixé à 3 au lieu de 1 du fait de la différence d'échelle avec la fonction de coût originale. L'ensemble des autres paramètre est identique à ceux fixés dans StarGAN [66] et les modèles sont optimisés pendant 300 000 itérations.

L'évaluation de la génération est effectuée sur les visages issus de l'ensemble de validation d'AffectNet, qui n'a pas de recoupement avec l'ensemble d'entraînement.

Visualisation Pour pouvoir visualiser l'espace de représentation $disc_3$, nous générons donc les visages associés à chacune des coordonnées sur les trois axes de -1 à 1. Dans la Figure 4.3.3, nous représentons les visages contenus dans les trois plans formés par ces axes. Nous avons également ajouté par code



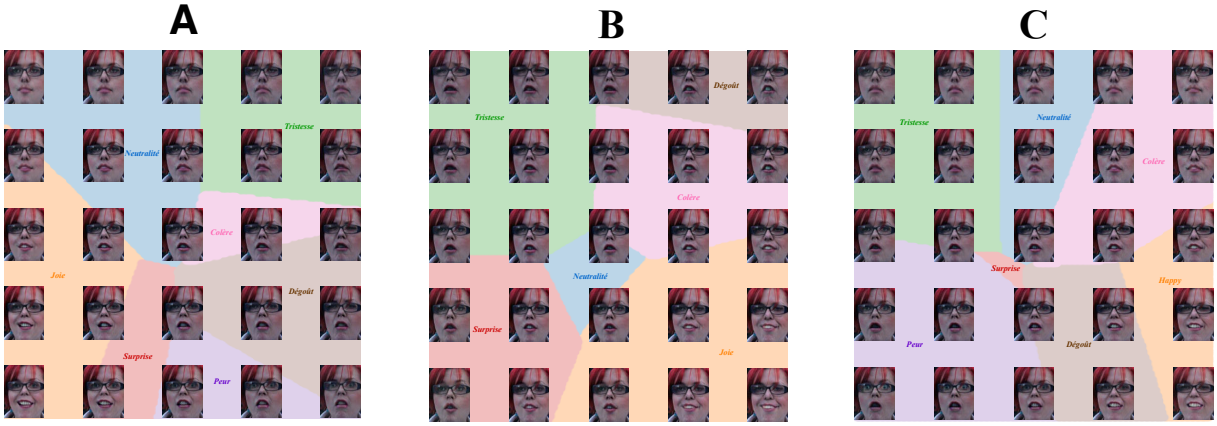


FIGURE 4.3.3 – Représentation de l'espace obtenu à partir de $disc_3$, décomposé en trois plans A, B et C.

couleur les expressions discrètes associées, soulignant les importantes variations qu'il est possible de générer au sein d'une même classe discrète. En observant les visages générés, nous pouvons également noter des directions continues et naturelles à travers les plans, conduisant par exemple d'expressions négatives à des expressions positives.

Génération d'expression discrète Nous comparons d'abord les trois méthodes en termes de génération d'expressions discrètes. Cette génération est directe pour l'approche *Discrete* et utilise les barycentres des classes discrètes dans l'espace concerné pour les autres méthodes (*EP* et $Sdisc_3$).

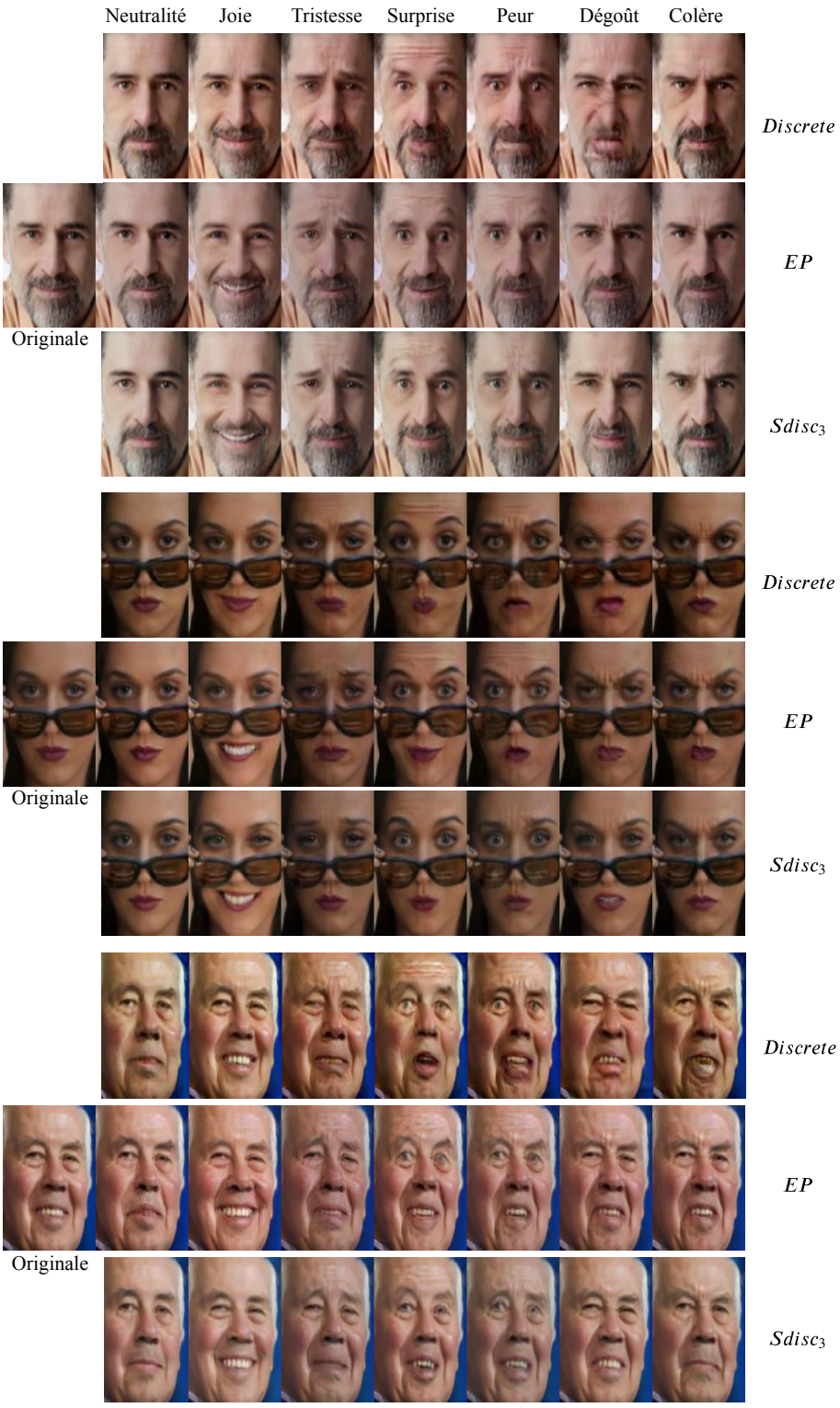


FIGURE 4.3.4 – Les 7 expressions discrètes générées par les trois approches *Discrete*, *EP* et $Sdisc_3$. Les visages sources ont été tirés aléatoirement dans l'ensemble de test.

La Figure 4.3.4 montre pour trois images données les 7 expressions discrètes qui ont été générées par chacune des méthodes. Les résultats semblent globalement similaires mais nous pouvons tout de même noter quelques différences intéressantes. Par exemple, les dents de la classe "joie" semblent plus nettes dans le cas de $Sdisc_3$. Une autre différence s'observe au niveau de la classe "Dégoût", où *Discrete* a

	RMSE sur la couleur moyenne				L_{rec}
Approche	Rouge	Vert	Bleu	RGB	
Discrete	4.5	6.3	10.2	7	0.22
EP	6.4	7.8	5.7	6.7	0.14
$Sdisc_3$	**3.7**	**3.1**	**3.2**	**3.4**	**0.12**

TABLE 4.3.2 – Évaluation de la qualité de reconstruction et de la conservation de la couleur pour les différentes approches (sur l'ensemble de test). Les scores les plus bas sont les meilleurs.

tendance à produire des artefacts, tandis que les expressions générées par *EP* et $Sdisc_3$ sont plus stables. En revanche, *EP* génère des expressions très similaires pour la colère et le dégoût, tandis que $Sdisc_3$ différencie bien les deux classes. Cela rejoint l'intuition observée en première partie suivant laquelle le dégoût et la colère possèdent des coordonnées très proches dans l'espace excitation-plaisir et qu'une troisième dimension est donc nécessaire pour correctement distinguer ces deux classes.

La classe neutre est également intéressante car symptomatique de la capacité à produire une expression avec une bouche fermée. Cela s'illustre particulièrement bien sur le troisième visage. L'intensité des expressions semblent plus importantes dans le cas de la méthode *Discrete*, ce qui peut être expliqué par le fait que les deux autres méthodes utilisent des barycentres, donc des valeurs non-extrêmes par leur nature même.

Enfin, particulièrement visible sur le troisième visage, *Discrete* semble changer la couleur moyenne des visages en modifiant l'expression. Pour vérifier cette impression de manière plus objective, nous avons calculé la couleur moyenne de l'image d'origine et la moyenne des couleurs moyennes des sept images générées. Nous avons alors pu mesurer la RMSE entre ces deux valeurs sur l'ensemble de test (5000 visages) pour chacune des approches, comme reporté en Table 4.3.2. Nous avons observé que l'erreur était plus basse pour les approches utilisant des représentations continues, et en particulier pour le canal bleu, où l'erreur de l'approche *Discrete* est très importante. L'erreur de reconstruction L_{rec} (décrite dans les méthodes) confirme également que les modifications non relatives à l'expression sont plus présentes chez *Discrete* que pour les autres approches. Nous avons évalué ici la capacité à préserver l'image d'origine, mais cela ne garantie pas que la qualité des images générées par $Sdisc_3$ est meilleure. En effet, l'évaluation de la qualité des images générées par un GAN implique l'usage d'une grande variété de métriques [41], ne faisant pas consensus puisque dépendant du problème.

Génération d'axes continus : excitation-plaisir-dominance La génération de transitions continues à partir d'une image est également intéressante et permet d'exhiber de manière visuelle les directions qui structurent les espaces de représentations. Pour pouvoir comparer les différentes approches, nous avons choisi d'utiliser les axes excitation-plaisir, qui ont l'avantage d'être facilement interprétables et souvent utilisés par la communauté psychologique [244].

Pour *EP*, cette génération des axes excitation-plaisir est directe. Pour $Sdisc_3$, nous nous référons à la régression effectuée dans l'explication de notre méthode. Enfin, pour *Discrete*, nous représentons les émotions par un vecteur one-hot (avec un seul 1 et 6 zéros) et nous créons la transition entre deux vecteurs one-hot, comme effectué pour ExprGAN [84], où les auteurs modélisent des transitions entre la classe neutre et les autres classes pour générer des intensités d'émotions variables. Ainsi, dans notre cas, nous modélisons l'axe du plaisir par une transition entre la tristesse (plaisir égal à -1 et excitation proche de 0) et la joie (plaisir égal à 1 et excitation proche de 0). Pour l'axe d'excitation, la transition est effectuée entre la neutralité (excitation et plaisir à 0) et la surprise (excitation à 1 et plaisir proche de 0). Cette deuxième transition ne modélise donc qu'une partie de l'intervalle représenté par l'axe de l'excitation.

Les Figures 4.3.6 et 4.3.5 nous permettent d'observer les générations des axes d'excitation et de plaisir par les trois méthodes (d'un bout à l'autre de l'axe). Nous pouvons en premier lieu noter que d'une approche à l'autre les expressions varient légèrement, simplement car les axes parcourus ne sont probablement pas totalement identiques, puisque *Discrete* et $Sdisc_3$ en produisent des approximations.

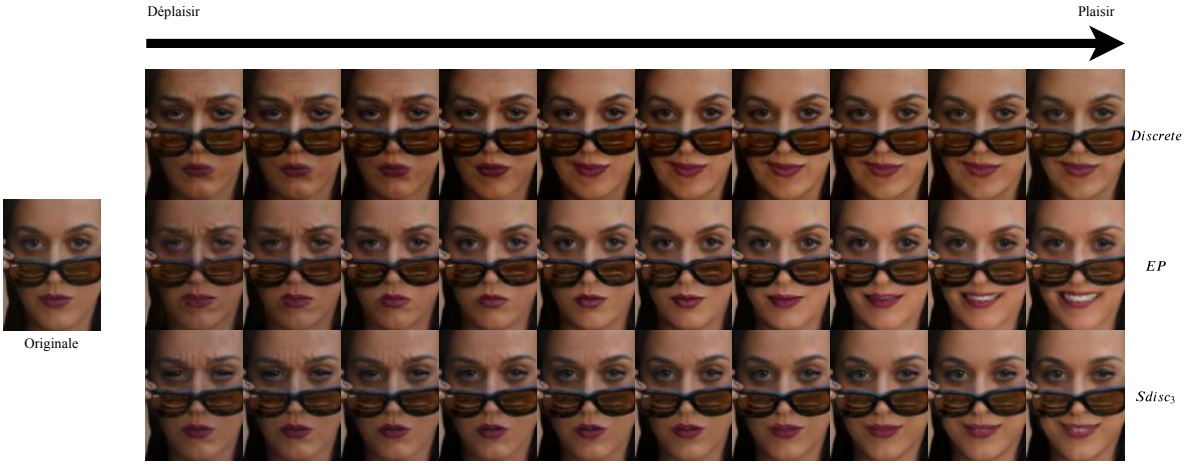


FIGURE 4.3.5 – Génération d'expressions le long de l'axe du plaisir. L'image originale est à gauche, chaque ligne représentant une approche.

FIGURE 4.3.6 – Génération d'expressions le long de l'axe de l'excitation. L'image originale est à gauche, chaque ligne représentant une approche.

Un point tout de même marquant réside dans le fait que *Discrete* produit une transition qui parait particulièrement brutale. En effet, plusieurs visages sont identiques et la transition s'effectue entre les visages en milieu de ligne. Au contraire, les deux autres approches semblent permettre d'obtenir une évolution relativement linéaire d'un bout à l'autre de l'axe.

Pour vérifier ces observations, nous avons utilisé une méthode plus objective, inspiré du protocole de l'"Inception-Score" [245]. Ainsi, nous avons récupéré le *ResNet-18-EP* de la partie précédente afin d'être capable d'estimer les valeurs d'excitation-plaisir d'un visage donné. Ensuite, nous avons utilisé celui-ci pour estimer les valeurs obtenues pour les expressions générées. Ainsi, cela nous a permis de tracer, pour chaque séquence de visages générés, une courbe de l'excitation ou du plaisir estimé en fonction de la valeur ciblée par le générateur. Une allure la plus linéaire possible correspondrait donc à une génération la plus précise, au bruit près de notre estimateur. Pour dégager une véritable tendance, nous avons construit en Figure 4.3.7 la courbe moyenne obtenue sur l'ensemble de test, pour l'excitation et pour le plaisir. Les courbes obtenues par *EP* devraient être parfaitement linéaires, ce qui n'est pas le cas du fait notamment du bruit de notre estimateur et du générateur. Néanmoins, l'allure reste linéaire et montre que cette évaluation a du sens.

Ensuite, les courbes obtenues par notre approche $Sdisc_3$ sont proches de celles obtenues par *EP* tandis que celles obtenues par *Discrete* sont plus proches d'un échelon. Cela confirme donc l'observa-

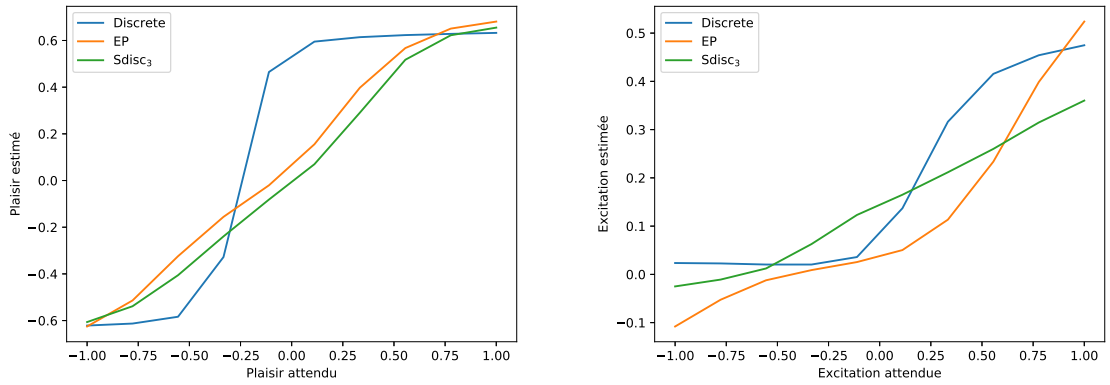


FIGURE 4.3.7 – Courbes du plaisir (à gauche) et de l'excitation (à droite) des expressions générées en fonction des valeurs cibles de plaisir (à gauche) et d'excitation (à droite) pour les trois différentes approches. Ces courbes sont moyennées sur l'ensemble de test.

tion visuelle que nous avons faite et montre la difficulté rencontrée par *Discrete* a correctement "paver" l'espace des possibles variations d'expressions.

Une troisième dimension Même si la communauté de la psychologie affective reconnaît l'excitation-plaisir comme une représentation pertinente de l'émotion, des travaux [195] ont souligné ses limitations et ont ainsi amené à la proposition de dimensions supplémentaires et en particulier la "dominance", qui peut être vue comme une expression du degré de confiance en soi.

Dans les paragraphes précédents, nous avons montré que $Sdisc_3$ permettait de retrouver à la fois des expressions discrètes mais aussi d'exhiber les directions excitation et plaisir. Nous avons d'ailleurs noté certains bénéfices apportés par l'ajout de la troisième dimension, comparé à l'usage de deux axes excitation-plaisir, notamment pour mieux distinguer dégoût et colère. Ainsi, cette troisième dimension semble apporter une information supplémentaire, qui est par construction orthogonale aux deux autres dimensions.

La génération le long de cet axe confirme cette idée, comme illustré dans la Figure 4.3.8. Ne disposant pas d'un estimateur de dominance et étant donnée la difficulté à trouver des données annotées avec cette dimension, nous avons choisi de nous comparer à des expressions générées artistiquement (par Allen Grabo [280], chercheur en psychologie). Nous observons alors le même type de transition, partant d'une absence confiance et se dirigeant vers une confiance totale. Cette première impression pourra être vérifiée en consultant plus d'exemples sur notre page [2] ou en expérimentant avec notre démonstration en ligne [3].

4.3.4 À propos de la démonstration

La démonstration en ligne est basée sur la même approche que ce qui a été décrit précédemment mais correspond à un modèle plus récent, qui a été entraîné à générer des images de meilleures résolutions (256x256x3 au lieu de 128x128x3) et avec plus de contexte autour du visage (ce qui est une difficulté supplémentaire en termes de convergence mais permet de faciliter le rendu). Ce modèle est plus robuste à différents bruits et présente de légèrement meilleurs résultats sur le corpus de test. Cela s'explique par son entraînement sur plusieurs millions de visages, issus d'AffectNet, d'Emotionet et de VGGFace2. Notons qu'il produit encore des artefacts face à certains cas spécifiques, illustrés en Figure 4.3.9, telles que des ouvertures de bouches importantes, des orientations du visage très éloignées de la position frontale et de très mauvaises résolutions ou conditions d'illumination.

2. https://github.com/vielzeuf/TheManyFacesOfEmotion_FG2019/blob/master/README.md
3. https://many-fe.noprod-b.kmt.orange.com/

FIGURE 4.3.8 – Illustration du troisième axe trouvé par produit vectoriel dans notre espace $disc_3$. La seconde ligne est un travail "à la main" qui a été effectué par Allen Grabo [280] pour illustrer ce qu'est la dimension de dominance.

FIGURE 4.3.9 – Première ligne : cas d'échecs avec présence d'artefacts détectables (flou, dents non réalistes, colorisation). Seconde ligne : cas de réussites avec robustesse face à des conditions difficiles (illumination, position du visage, changement de domaine, composition de visages).

4.4 Conclusions

4.4.1 En résumé

Ce chapitre a présenté nos travaux vers une représentation plus générique et interprétable des expressions faciales. Ainsi, nous avons inséré un goulot d'étranglement (avec une réduction de dimension bien plus forte que dans des approches utilisant ce type d'approche [122, 208, 117]) au sein d'un réseau de neurones et entraîné celui-ci à la classification d'émotion discrète. Nous avons alors montré que l'espace latent compact obtenu présente des propriétés particulières, corrélées à celles des dimensions psychologiques. En particulier, notre étude sur l'impact de la dimensionnalité d'une représentation de l'expression sur la performance de classification montre l'apport d'une troisième dimension face à un espace à deux dimensions comme l'excitation-plaisir. De plus, nous montrons que des représentations plus larges deviennent peu pertinentes en apportant un gain peu élevé et exploitant des biais propres à chaque base de données, ce qui implique une baisse de performance quand elles sont appliquées à d'autres bases. Ainsi, nous avons également souligné l'inconsistance d'annotation entre différentes bases de données d'expression faciale et montré l'importance de prendre celle-ci en compte lors de la construction d'un modèle de reconnaissance de l'expression faciale.

Le modèle compact et performant ainsi obtenu a permis d'annoter automatiquement une grande diversité de visages et d'expressions. En utilisant alors une approche générative adverse, nous avons pu

visualiser le comportement de notre représentation tri-dimensionnelle dans l'espace image. Cela s'est traduit non seulement par la capacité à générer facilement des expressions discrètes pertinentes, mais également par un contrôle fin sur la nature des expressions générées, en ayant la possibilité de retrouver des axes psychologiques tels qu'excitation-plaisir mais aussi d'en exhiber d'autres tels que la dominance. De plus, nous avons pu produire un modèle de génération plus robuste en annotant automatiquement un large nombre de visages et en pouvant ainsi modifier des images dans des conditions véritablement non contrôlées, telles qu'illustrées sur notre page et à travers notre démonstration en ligne.

4.4.2 Perspectives

Approches génératives récentes Certaines approches génératives récentes présentent plusieurs avantages en comparaison avec l'utilisation de StarGAN [66]. Il est par exemple possible de générer des contenus avec une meilleure qualité et une meilleure résolution, par exemple en se basant sur des approches d'augmentation de la taille progressives [145] (*i.e.* qui augmentent au fur et à mesure de l'entraînement la taille des entrées et le nombre de couches des réseaux générateurs et discriminateurs). Il serait intéressant d'exploiter ce type de méthodes dans le cadre de la modification de contenus, de manière à assurer une meilleure résolution et une plus grande stabilité dans la génération.

Représentation à plusieurs étages et lien avec les Action Units Pour obtenir le modèle (plus robuste) utilisé dans la démonstration en ligne, nous avons annoté un grand nombre de visages avec la représentation $disc_3$. Cette annotation pourrait être améliorée de diverses manières, notamment en utilisant plusieurs étages dans le *ResNet-disc* et en forçant une couche cachée inférieure à permettre la classification d'Action Units. Cette approche presque "multi-tâche" pourrait alors permettre de lier les avantages de contrôle et de continuité des expressions apportés par $disc_3$, mais également d'inclure la qualité de détails et la capacité d'animation (*e.g.* ouverture et fermeture "propre" de la bouche sans aucune autre modification) proposée dans des approches telles que celle de Pumarolova [227].

Extension de l'approche à d'autres tâches Le fait d'utiliser une représentation continue et d'annoter automatiquement une grande variété de visages permet d'obtenir plus de robustesse et ainsi de ne plus avoir à travailler dans des conditions non contrôlées. La même idée pourrait donc être appliquée dans des domaines connexes, tels que le rajeunissement / vieillissement ou encore la modification d'attributs (voire un apprentissage de la génération en multi-domaine, comme proposé originellement par les auteurs de StarGAN [66]). Nous étudierons également dans le Chapitre 5 certains aspects bénéfiques d'une représentation multi-tâche.

Chapitre 5

Transfert de connaissances à partir de plusieurs sources

Table des matières

5.1 Introduction

La communauté de la vision par ordinateur a implémenté et entraîné un grand nombre de modèles (notamment de réseaux de neurones) afin de résoudre une grande variété de problèmes. Cette immense "banque" de modèles entraînés peut être vue comme un rassemblement de différentes sources de connaissances extrêmement riches, compressées et adaptées à différentes tâches (*e.g*. classification d'objets, prédiction de l'émotion, identification faciale). Ces sources peuvent alors être ré-utilisées pour résoudre de nouveaux problèmes, en transférant la connaissance précédemment acquise.

Une manière classique de procéder est de choisir une seule de ses sources de connaissances, *i.e*. dans notre cas un réseau de neurones dont les paramètres ont déjà été entraînés, et d'utiliser celle-ci sur une nouvelle tâche. Plus concrètement, nous faisons l'hypothèse (sans perte de généralité, *cf*. Chapitre 2), qu'un réseau de neurones pré-entraîné peut être divisé en deux parties : un encodeur et une partie de décision finale. L'encodeur permet de transformer une entrée (*e.g*. une image) en une représentation (*i.e*. un vecteur) proche du niveau sémantique de la tâche, tandis que la partie de décision finale permet d'apporter une réponse à la tâche donnée. En reprenant l'encodeur et en l'utilisant pour extraire des représentations des entrées associées à une nouvelle tâche, il est alors possible d'apprendre une nouvelle partie de décision finale, qui va répondre à la nouvelle tâche. La connaissance contenue dans les paramètres de l'encodeur aura donc été transférée. Ceux-ci peuvent de plus être affinés sur la nouvelle tâche par la suite.

Lors du transfert de connaissances, le choix de la source de connaissances reste souvent empirique et se fait par exemple en sélectionnant un modèle entraîné sur une tâche initiale qui semble arbitrairement proche de celle à résoudre ou sur une tâche particulièrement riche et complexe (suffisamment pour englober plusieurs problèmes), telle que la classification d'objets d'ImageNet [74].

Pour automatiser le processus de sélection, des travaux récents [3, 309] ont montré qu'il est possible d'exhiber un espace des relations entre les tâches et d'ainsi identifier les tâches (et donc les sources de connaissances) qui apporteront le plus pour résoudre une tâche-cible.

Néanmoins, les modèles écartés par le processus de sélection sont tout à fait susceptibles de contenir de l'information utile pour la nouvelle tâche. Par analogie au domaine de la fusion multimodale (*cf*. Chapitre 6), chacun des modèles peut être considéré comme une modalité, *e.g*. en considérant sa dernière couche cachée comme une modalité. Or, certaines modalités peuvent obtenir de mauvaises performances lorsqu'elles sont utilisées seules mais comporter des informations très complémentaires et donc utiles lorsqu'elles sont combinées avec d'autres modalités. Par exemple, nous avons vu dans le Chapitre 3 où nous cherchions à prédire l'émotion à partir de l'expression faciale et du son, que la modalité visuelle obtenait de bien meilleures performances. Pourtant, lors de la fusion des modalités, le gain apporté par l'audio était significatif. Au contraire, le fait d'utiliser des modalités très redondantes amène souvent à un gain de performance limité.

Si nous étendons cette analogie en considérant M modalités, une méthode pertinente pour toutes les prendre en compte serait alors d'apprendre une représentation commune. Ce type de problème n'est pas trivial, comme souligné par Zamir *et al*. [309], qui montrent que simplement rassembler de manière naïve toutes les modalités (*i.e*. en transférant comme représentation la concaténation des représentations cachées de chaque modèle) ne permet pas d'obtenir de bons résultats de transfert de connaissances.

Dans ce chapitre, nous proposons tout d'abord de formuler et construire ce problème de transfert de connaissances multi-source, en l'appliquant plus particulièrement au cas spécifique de l'analyse de visages. Nous étudions dans une seconde section l'apport d'une réduction de dimensionnalité pour extraire une représentation plus générale et compacte à partir des représentations issues de tous les encodeurs. Enfin, nous mettons en avant qu'il est possible de distiller cette représentation dans un unique encodeur, qui sera validé sur une large variété de problèmes d'analyse faciale.

5.2 Construction du problème de transfert multi-source

La problématique décrite en introduction présente des similarités avec d'autres domaines de recherches. Par exemple, nous pouvons citer l'Apprentissage tout au Long de la Vie ou *Life Long Learning* (LLL) [216], qui consiste à entraîner successivement le même modèle sur différentes tâches, sans que l'apprentissage de nouvelles tâches ne dégrade la performance sur les précédentes (on parle alors d'oubli catastrophique). Nous pouvons également considérer des problèmes de transfert de connaissances classiques (nouvelle tâche avec des données de même type) et d'adaptation de domaines (tâche identique mais changement de la distribution, voire du type des données). Néanmoins, dans notre cas, nous regroupons non pas une mais plusieurs sources de connaissances (sous la forme de modèles-pré-entraînés), issues de tâches et de domaines différents et les utilisons pour résoudre une nouvelle tâche avec potentiellement un domaine différent.

Nous considérons donc que pour évaluer au mieux le fonctionnement d'une approche permettant de résoudre un tel problème, il est tout d'abord essentiel de le formaliser, mais également de disposer d'un protocole d'entraînement et d'évaluation. Cela nous a amené naturellement vers une application concrète où de nombreuses données et modèles sont disponibles : l'analyse des visages. Nous avons ainsi pu définir une banque de modèles spécifiques et différentes tâches à résoudre, liées à l'analyse des visages.

5.2.1 Formulation du problème

Définissons tout d'abord le concept de *connaissance* comme la capacité abstraite à résoudre parfaitement une tâche t sur un domaine d. Nous nous limitons dans le reste du chapitre à la famille des tâches de classification / régression à partir d'apprentissage automatique. Il est alors possible de définir une estimation de la *connaissance*, notée $\mathcal{C}_{(t,d)}$, pour un domaine d et une tâche t, *i.e.* en écrivant $\mathcal{C}_{(t,d)} = (\mathcal{E}_{(t,d)}, \mathcal{P}_{(t,d)})$, avec :

— $\mathcal{E}_{(t,d)}$ est une fonction permettant de projeter chaque élément $\mathbf{x}$ du domaine d vers une représentation $\mathbf{h}$

— $\mathcal{P}_{(t,d)}$ est une fonction permettant de projeter chaque représentation $\mathbf{h}$ vers une sortie $\mathbf{y}$

Quand nous faisons face à un nouveau problème défini par une tâche t' sur un domaine d', $\mathcal{C}_{(t,d')}$ ou $\mathcal{C}_{(t',d)}$ peuvent être utilisés pour améliorer l'apprentissage de $\mathcal{C}_{(t',d')}$. Une technique de transfert de connaissances classique consiste alors à réutiliser (ou raffiner) $\mathcal{E}_{(t,d)}$ tout en apprenant un nouveau $\mathcal{P}_{(t',d)}$. Tandis que l'adaptation au domaine cherche à apprendre un couple performant $\mathcal{E}_{(t,d+d')}$ et $\mathcal{P}_{(t,d+d')}$.

Le problème que nous cherchons à résoudre ici correspond à la capacité d'effectuer un transfert de connaissances et une adaptation à un domaine, en utilisant non une seule connaissance $\mathcal{C}_{(t,d)}$ mais un ensemble $\mathcal{S}$ de connaissances, que nous nommerons *connaissances sources* et dont les encodeurs $\mathcal{E}_{(t_i^S,d_i^S)}$ seront notés par soucis de lisibilité $\mathcal{E}_i^S$.

Notre approche va se baser principalement sur le fait de définir un opérateur $\mathcal{E}^g = \mathcal{R}((\mathcal{E}_i^S)_{i\in\mathcal{S}})$ avec $\mathcal{R}$ une fonction de rassemblement permettant d'utiliser l'ensemble des $\mathbf{h_i}$ (les représentations extraites par les $\mathcal{E}_i^S$). L'idée est en fait de rassembler toute la connaissance relative aux tâches t_i et aux domaines d_i en une seule représentation générale $\mathbf{h_g}$.

Pour obtenir une telle représentation $\mathbf{h_g}$, une technique directe serait de concaténer toutes les sorties $\mathbf{h_i}$ de $\mathcal{E}_i^S, i \in \mathcal{S}$. Mais comme souligné par Zamir *et al.* [309], une telle approche présente des difficultés à généraliser et implique donc de mauvaises performances en transfert de connaissances.

Comme évoqué en introduction, une possibilité pour obtenir $\mathcal{C}_{(t_j,d_j)}$ serait alors d'adopter une méthode de sélection [309, 3], en trouvant parmi $\mathcal{S}$ quelques connaissances $\mathcal{C}_{(t_i,d_i)}$ où t_i est très corrélée à t_j.

Néanmoins, il pourrait être intéressant d'exploiter toutes les connaissances sources et de les rassembler en une estimation de connaissance plus générale $\mathcal{C}_{(t_g,d_g)}$. C'est pourquoi nous proposons en Section 5.3 de définir $\mathcal{R}$ comme un opérateur de réduction, permettant de fournir une représentation $\mathbf{h_g}$

de dimension réduite, mais ayant la capacité d'approximer tous les $\mathbf{h_i}$ et pouvant donc potentiellement conduire à une meilleure généralisation en supprimant les biais inhérents à une trop grande complexité.

Enfin, il est important de noter que l'encodeur $\mathcal{E}^g$, associé à $\mathcal{C}_{(t_g,d_g)}$ est composé de l'ensemble des $\mathcal{E}^S_{(t_i,d_i)}$ suivi de $\mathcal{R}$. Encoder une entrée $\mathbf{x}$ vers la représentation $\mathbf{h_g}$ est donc une opération très coûteuse en calcul. Dans la section 5.4.1, nous avons fait le choix d'adopter une méthode de distillation afin de transformer $\mathcal{E}^g$ en un unique modèle moins coûteux $\mathcal{E}^g_{unique}$, qui projette directement $\mathbf{x}$ vers une représentation $\hat{\mathbf{h}}_\mathbf{g}$, permettant un transfert de connaissances plus aisé ainsi qu'une adaptation plus complète à un nouveau domaine.

5.2.2 Connaissances sources

Nous utilisons 6 modèles pré-entraînés comme sources de connaissances. Un point important est d'éviter au maximum des recoupements entre les domaines qui ont été utilisés pour entraîner ces modèles et les domaines qui seront utilisés au moment de l'évaluation. Ainsi, nous avons fait le choix d'entraîner nous même ces modèles sources, de manière à minimiser ce recoupement, bien que les méthodes qui seront décrites dans la suite du chapitre sont applicables en récupérant n'importe quel modèle pré-entraîné.

Ainsi, nous disposerons de six sources de connaissances pour l'analyse de visages :

— **Expression-AffectNet** : classification d'une expression faciale parmi 7. PyramidNet-50 [117] identique à celui décrit dans le Chapitre 4 en section 4.2 : entraîné sur 300 000 visages issus de la base de données AffectNet [198].

— **Identité-MS** : classification de l'identité. ResNet-50 [122] entraîné sur un sous-ensemble de 6.5 millions de visages de Microsoft-Celeb [116]. L'évaluation a été effectuée sur LFW [130] contenant 13233 visages.

— **AgeReg-IMDb** : régression de l'âge. ResNet-50 [122] défini et entraîné de la même manière que [13] sur un sous-ensemble de IMDb [240].

— **Genre-IMDb** classification du genre. ResNet-50 [122] pré-entraîné sur Identité-MS, puis affiné sur IMDb [240].

— **Attributs-CelebA** classification d'attributs (*e.g.* moustache, lunettes, ouverture de la bouche). PyramidNet-50 [117] entraîné sur environ 200 000 visages de CelebA [180].

— **Objet-ImageNet** classification d'objets (*e.g.* voiture, chien). ResNet-50 [122] entraîné sur ImageNet (1000 classes) [74]. Bien que le domaine ne contienne pas (ou très peu) de visages, ce type de modèle pré-entraîné est très souvent utilisé en transfert de connaissances du fait de la qualité des représentations apprises.

Connaissance	Taille de h_i	# paramètres de $\mathcal{E}^S_i$	# visages
Expression-AffectNet	376	1.7 M	0.3 M
Identité-MS	2048	25.6 M	6.5 M
AgeReg-IMDb	512	25.6 M	0.5 M
Genre-IMDb	128	25.6 M	0.5 M
Attributs-CelebA	376	1.7 M	0.2 M
Objet-ImageNet	2048	25.6 M	14 M
Total	5488	105.8 M	22 M

TABLE 5.2.1 – Résumé des six connaissances sources et des représentations h_i extraites par leurs encodeurs.

Ces 6 connaissances sources possèdent donc chacune un encodeur $\mathcal{E}_m$, qui permet d'extraire une représentation $\mathbf{h_i}$ d'un visage donné (qui correspond en pratique à la dernière couche cachée du modèle pré-entraîné associé). Ainsi, comme l'illustre la Table 5.2.1, il nous est possible d'extraire une représentation $(\mathbf{h_i})_{i\in(1..M)}$ de taille 5488 pour chaque visage donné, représentant une connaissance issue de 22

millions d'images. Le fait d'utiliser ces six encodeurs implique donc un nombre énorme de paramètres (105.8 millions) et une représentation pouvant être très redondante et donc sous-optimale. Ces observations pratiques motivent donc d'autant plus : (a) l'étude d'un opérateur de réduction $\mathcal{R}$ de manière à compacter cette représentation et (b) la réduction du nombre de paramètres nécessaires à l'encodage de la représentation, en développant par exemple un encodeur unique $\mathcal{E}^g_{unique}$.

5.2.3 Connaissances cibles

Nous définissons maintenant des connaissances cibles en analyse du visage, qui correspondent à 9 tâches sur 6 domaines différents. Ces connaissances sont liées en partie aux connaissances sources, mais présentent un domaine différent et/ou une tâche différente. La Table 5.2.2 reprend l'ensemble des connaissances et les métriques d'évaluation associées. Concernant les connaissances cibles, nous avons rassemblé :

— **AgeReg-UTK**,**AgeClassif-UTK**, **Genre-UTK**, **Ethnie-UTK** : 4 tâches peuvent être évaluées sur UTKFace [324], une base de données d'environ 20 000 visages annotés en genre (2 catégories), âge (régression ou 7 catégories) et ethnie (5 catégories).

— **AgeReg-FG** : une tâche de régression de l'âge sur 1002 visages de FG-NET [165]. Le protocole *Leave One Person Out* (LOPO) consiste en une validation croisée, mais en entraînant sur toutes les identités sauf une et en évaluant uniquement sur celle-ci. On reproduit l'opération pour chacune des identités, le score final étant la moyenne des scores obtenus.

— **Expression-SFEW** : une tâche de classification de l'expression faciale à partir de SFEW [80] (utilisé dans les Chapitres 3 et 4), contenant 1766 visages extraits de films.

— **Expression-RAF** : une tâche de classification de l'expression faciale à partir de RAF [172] (*cf.* également les Chapitres 3 et 4), contenant 15 339 avec une annotation très cohérente.

— **Douleur-UNBC** : une tâche d'estimation de la douleur de 0 à 6 (évaluée par Erreur Moyenne Absolue ou *Mean Absolute Error* (MAE)), à partir des 48391 visages de UNBC-ShoulderPain [186].

— **Identité-LFW** : une tâche de mise à paire des identités, sans entraînement sur les visages de LFW et en suivant le protocole "non restreint, labels externes". Le but est d'estimer si deux visages appartiennent à la même personne (paire positive) ou non (paire négative) à partir de la distance L2 entre les représentations neuronales de ces deux visages.

Connaissances	Métriques	Protocole d'évaluation
Expression-AffectNet*	Accuracy	7 classes, validation d'AffectNet
Expression-RAF	Accuracy	7 classes, test de RAF
Expression-SFEW	Accuracy	7 classes, validation de SFEW
Identité-MS*	Accuracy	Paire, évaluation sur LFW
Identité-LFW	Accuracy sur les paires	LFW, protocole "non restreint, labels externes"
Genre-IMDb*	Accuracy	2 classes, évaluation sur LFW
Genre-UTK	Accuracy	2 classes, test de UTKFace [324]
Attributs-CelebA*	Erreurs moyennes	40 attributs, seuil fixe à 0.5, évaluation sur le test de CelebA
AgeReg-IMDb*	MAE	Évaluation sur FG-NET [165]
AgeReg-FG	MAE	Régression entre 0 et 100, protocole LOPO
AgeReg-UTK	MAE	Régression entre 0 et 100, test de UTKFace [324]
AgeClassif-UTK	Accuracy	5 classes, test de UTKFace [324]
Ethnie-UTK	Accuracy	7 classes, test de UTKFace [324]
Douleur-UNBC	MAE	Toutes les annotations sont utilisées, 7 niveaux de douleurs [322], protocole LOPO
Objet-ImageNet*	Taux d'erreur top-5	1000 classes, évaluation sur le test d'ImageNet

TABLE 5.2.2 – Métrique et protocole utilisés pour évaluer chacune des connaissances. Les connaissances marquées par un * sont les connaissances sources.

5.2.4 Vers une connaissance générale

Le but de ce chapitre est d'être capable de construire un modèle unique à partir des connaissances sources, qui permettrait d'atteindre de meilleures performances sur les domaines et les tâches des connais-

sances cibles. En d'autres mots, nous cherchons à créer une connaissance plus générale $\mathcal{C}_{(t_g,d_g)}$, permettant de transférer l'information utile des connaissances sources vers les connaissances cibles.

Pour développer cette connaissance générale, il est alors nécessaire de disposer d'une tâche t_g et d'un domaine d_g. Notre tâche générale sera exprimée comme l'ensemble des tâches associées aux connaissances sources, tandis que le domaine d_g sera composé d'un large ensemble de visages pavant au mieux l'espace des possibles. Ainsi, la création de cette connaissance générale correspond à l'apprentissage sur d_g d'un modèle unique permettant de répondre à toutes les tâches des connaissances sources.

Ensuite, pour évaluer sa capacité à généraliser au delà des connaissances sources, nous évaluerons ce modèle sur les connaissances cibles en suivant les protocoles et les métriques de chaque base de données et tâche définies dans la littérature.

Utilisation d'un domaine non annoté d_g Cet ensemble est composé de 4.12 millions de visages, extraits de trois bases de données : l'ensemble d'entraînement de VGGFace2 [50] (3.14 millions), la base de données Emotionet [98] (0.72 million), et la base de donnée IMDB-WIKI [240] (0.26 million). Aucune annotation relative à ces différents domaines n'est conservée, car seule les images seront utilisées par la suite.

Domaine	Visages détectés	Visages non détectés
CelebA	202442	177
UTKFace	24018	98
FG-NET	1002	0
SFEW	1732	37
RAF	15330	9
ShoulderPain	48391	0
LFW	13233	0
d_g	4.12 M	0.02M

TABLE 5.2.3 – Nombre de visages détectés / non détectés sur les différents domaines.

Harmonisation Enfin, de manière à permettre une certaine homogénéité entre tous les domaines et tâches traitées, nous avons choisi d'effectuer différents pré-traitements. Nous avons utilisé un détecteur et aligneur de visages interne (basé sur une approche en cascade et un détecteur de point d'intérêt) et avons ainsi sélectionné un seul visage central par image, qui a ensuite été retaillé aux dimensions 300x300x3. La Table 5.2.3 rapporte le nombre d'images où un visage a été détecté pour chaque domaine (première colonne). Quand un visage n'est pas détecté (seconde colonne), nous considérons durant l'évaluation des modèles que (a) s'il s'agit d'un problème de classification, que la prédiction est incorrecte et (b) s'il s'agit d'une régression, que la valeur estimée par le modèle est la moyenne entre les valeurs minimales et maximales.

Vue d'ensemble En résumé, pour résoudre le problème du transfert multi-source, nous souhaitons obtenir un unique encodeur capable d'extraire une représentation générale $\mathbf{h_g}$ à partir d'une image $\mathbf{x}$, cette représentation étant suffisamment générale pour ensuite être utilisée pour résoudre de nouvelles tâches, sur de nouveaux domaines. Il existe donc deux enjeux, que nous avons choisi d'étudier séparément : la réduction, *i.e.* obtenir $\mathbf{h_g}$ à partir des $\mathbf{h_i}$, et la distillation, *i.e.* obtenir $\mathbf{h_g}$ à partir de $\mathbf{x}$.

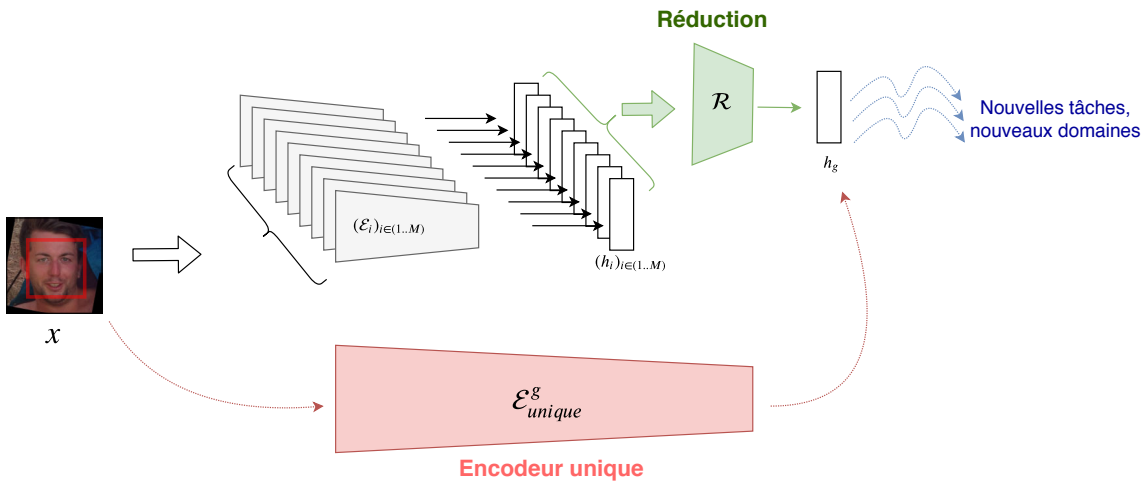


FIGURE 5.2.1 – Vue d'ensemble des enjeux de la problématique d'un transfert multi-source. L'opérateur de *réduction* $\mathcal{R}$ et l'*encodeur unique* $\mathcal{E}^g_{unique}$ ne peuvent être entraînés qu'à partir des données issues de d_g, tandis que l'ensemble des encodeurs $\mathcal{E}^S_i$ a été entraîné à partir des domaines des connaissances sources.

5.3 Réduction de dimensionnalité

Comme évoqué dans la construction du problème de transfert multi-source, nous souhaitons réduire la dimension des représentations issues des différents encodeurs $\mathcal{E}^S_i$, de manière à obtenir une représentation plus générale et reprenant notamment l'ensemble des connaissances sources. Nous recherchons donc l'opérateur de réduction $\mathcal{R}$ précédemment défini.

5.3.1 Pourquoi réduire la dimensionnalité ?

La réduction de dimensionnalité est une problématique qui a déjà été traitée en profondeur par la communauté. Comme nous avons pu l'observer tout au long du Chapitre 4, une représentation réduite d'un signal permet plus de contrôle et d'analyse, mais peut également amener à supprimer certaines redondances empêchant une bonne généralisation de la représentation. Parmi les méthodes classiques utilisées pour cela, nous pouvons citer :

PCA C'est une méthode très connue et utilisée pour ce type de problématique (avec de nombreuses variantes en découlant [138, 132, 305, 267]). L'idée principale de cette famille de méthode est de transformer un ensemble de variables corrélées et redondantes en un ensemble de nouvelles variables indépendantes les unes des autres et donc moins nombreuses. Pour cela, il est nécessaire de déterminer un ensemble d'axes W orthogonaux les uns aux autres et sur lesquels le fait de projeter les données d'origine maximise leur variance. Ces axes sont les composantes principales et permettent de construire une nouvelle représentation, dans un hyperplan de dimension égale aux nombres de composantes principales utilisées. Le choix de ces axes aboutit donc à proposer des représentations linéaires les plus efficaces en matière de compromis nombre de coefficients / erreur de reconstruction. Une opération de réduction de la dimension a donc été effectuée et a permis de supprimer des éléments redondants entre les différentes variables. Néanmoins, dans notre cas, nos données d'origine sont la concaténation des représentations $\mathbf{h_i}$, issues de M encodeurs présentant des caractéristiques différentes (*cf.* Table 5.2.1) et permettant de résoudre des tâches plus ou moins corrélées. Du fait de la complexité des représentations utilisées, il est

alors possible que la projection linéaire sur un hyperplan ne permette pas d'aboutir à une représentation véritablement pertinente.

Cas non-linéaire et variété En effet, prenons l'exemple de la Figure 5.3.1 et supposons que les données d'entrée de notre opérateur de réduction parcourt avec une forme de S un espace à trois dimensions similaire à celui de la première ligne de la figure. Alors, l'application d'une approche linéaire telle que la PCA (en première colonne de la Figure) ne permet pas de séparer finement les données, car celles-ci ont une répartition non linéaire. En revanche, en adoptant une représentation non linéaire (*i.e.* en choisissant une direction d_2 curviligne) et en considérant donc que les données sont représentées par une variété, que nous nous contenterons ici de définir comme un espace non nécessairement plan, il est alors possible d'obtenir une séparation bien plus pertinente.

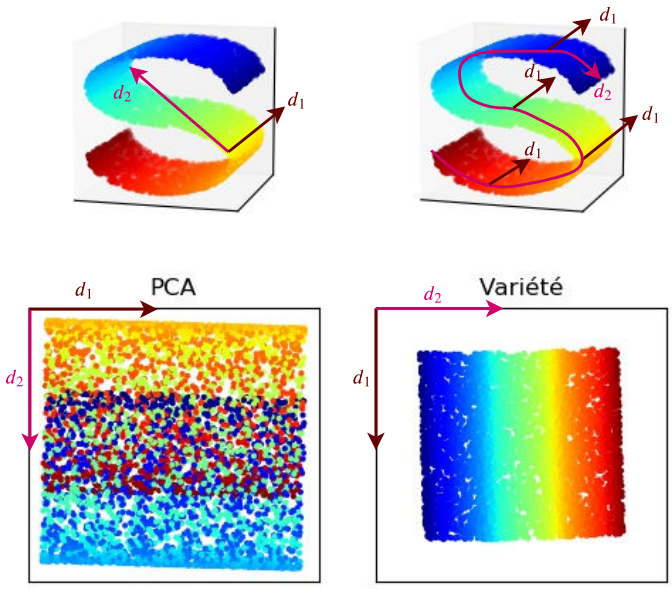


FIGURE 5.3.1 – Comparaison des projections d'un espace 3d vers deux directions (en première ligne) par deux approches (deuxième ligne) : PCA et approche non-linéaire de modélisation d'une variété. Figure réalisée à partir du code disponible sur https://scikit-learn.org/stable/modules/manifold.html

Méthodes neuronales Il existe de nombreuses méthodes d'apprentissages de variétés ou *manifold learning* [131] et de réduction de la dimensionnalité. Il est notamment possible d'effectuer ce type d'opérations à l'aide de modèles neuronaux et plus particulièrement d'auto-encodeurs [24], un auto-encodeur linéaire présentant de grandes similarités avec une PCA tandis qu'un auto-encodeur non-linéaire et à plusieurs couches permet de modéliser des axes non-linéaires par morceau et donc d'obtenir une représentation similaire à celles des approches d'apprentissage de variétés [247, 171].

5.3.2 Approche adoptée

Nous considérons maintenant le problème de l'apprentissage de l'opérateur de réduction $\mathcal{R}$, permettant d'extraire la représentation h_g que nous recherchons. Pour cela, nous disposons des M encodeurs $\mathcal{E}_i^S$ et d'un large domaine d'apprentissage non supervisé d_g.

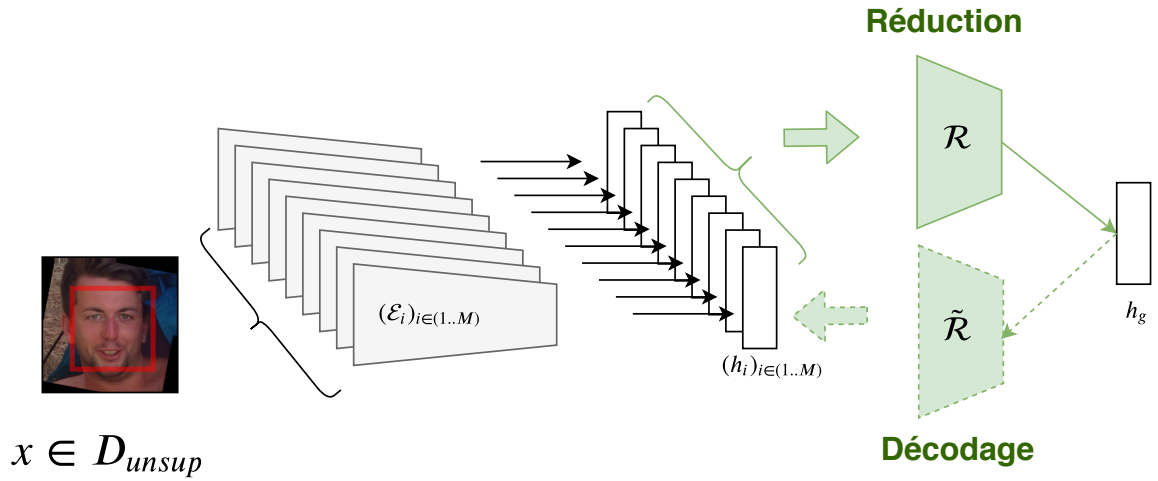


FIGURE 5.3.2 – Approche adoptée pour la réduction de dimensionnalité : l'ensemble des $\mathbf{h_i}$ est concaténé et traité par un encodeur $\mathcal{R}$ vers une représentation de plus faible dimension $\mathbf{h_g}$. Le décodeur $\tilde{\mathcal{R}}$ doit alors reconstruire la concaténation des $\mathbf{h_i}$ à partir de la représentation $\mathbf{h_g}$, formulation typique d'une tâche d'auto-encodeur.

Auto-encodeur En revenant à ce qui a été détaillé dans la sous-section précédente et comme illustré en Figure 5.3.2, nous pouvons définir $\mathcal{R}$ comme un encodeur neuronal, passant d'une représentation large $(\mathbf{h_i})_{i\in(1..M)}$ à une représentation compacte $\mathbf{h_g}$ et entraîner celui-ci comme un AE [24]. Plus formellement, nous pouvons écrire que :

$$\mathbf{h_g} = \mathcal{R}((\mathbf{h_i})_{i\in(1..M)}) \tag{5.3.1}$$

$$(\tilde{\mathbf{h}}_\mathbf{i})_{i\in(1..M)} = \tilde{\mathcal{R}}(\mathbf{h_g}) \tag{5.3.2}$$

avec $\tilde{\mathcal{R}}$ un décodeur.

$\mathcal{R}$ et $\tilde{\mathcal{R}}$ ont alors pour objectif de reconstruire $(\mathbf{h_i})_{i\in(1..M)}$, *i.e.* avec une fonction de coût $L = \Sigma_{i=1}^{M} ||\tilde{\mathbf{h}}_\mathbf{i} - \mathbf{h_i}||^2$.

Paramétrage de $\mathbf{h_g}$ Nous verrons dans la sous-section suivante qu'il existe différentes variantes de l'AE [24], tels que le VAE [156] ou le DAE [282].

De plus, des variations dans l'architecture même de $\mathcal{R}$ et $\tilde{\mathcal{R}}$ peuvent également avoir un impact et un paramètre en particulier semble important : la dimension de h_g. En effet, une dimension trop proche de celle de l'entrée impliquera une mauvaise capacité à généraliser, tandis qu'une dimension trop faible empêchera tout apprentissage. Un choix arbitraire que nous pourrons faire avant l'étude empirique correspond à donner pour dimension à $\mathbf{h_g}$ une dimension proche de la dimension moyenne des $\mathbf{h_i}$, *i.e.* dans notre cas une dimension de 1024, pour un vecteur d'entrée de dimension 5488.

L'architecture de $\mathcal{R}$ est un MLP composé de 3 couches de perceptrons : 5488×3136, 3136×1792 et 1792×1024, tandis que $\tilde{\mathcal{R}}$ présente la même architecture en inversé, allant de 1024 vers 5488.

5.3.3 Étude empirique

Pour évaluer notre opérateur de réduction, nous considérons d'abord sa capacité à reconstruire les $\mathbf{h_i}$, en comparaison avec différentes méthodes. Puis nous proposons d'extraire $\mathbf{h_g}$ de tous les visages des domaines des connaissances cibles et d'entraîner un MLP sur chacune des nouvelles tâches. Enfin, nous étudions l'impact de la dimension de $\mathbf{h_g}$ sur les performances précédentes.

Qualité de la reconstruction des connaissances sources Concernant la qualité de la reconstruction obtenue par l'AE, nous évaluons la rRMSE pour chacun des six $\mathbf{h_i}$ dans la Table 5.3.1.

Connaissance représentée par $\mathbf{h_i}$	PCA	AE [24]	VAE [156]	DAE [282]
Expression-AffectNet	0.28	**0.23**	0.26	0.26
Identité-MS	0.25	**0.22**	0.23	0.26
Objet-ImageNet	0.48	0.26	0.30	**0.25**
AgeReg-IMDb	0.23	**0.19**	0.22	**0.19**
Attributs-CelebA	0.33	**0.27**	0.31	0.29
Genre-IMDb	0.25	**0.25**	0.27	0.27
Moyenne des rRMSE	0.31	**0.24**	0.27	0.25

TABLE 5.3.1 – rRMSE obtenues sur l'ensemble de test de d_g pour chacun des h_i et en moyenne.

Notons tout d'abord qu'une simple PCA est largement dépassée par les autres approches non-linéaires, confirmant les intuitions données en début de Section 5.3. Ensuite, l'AE obtient de meilleures performances que les deux autres méthodes, qui présentent pourtant le même nombre de paramètres et des techniques de régularisation supplémentaires. Seul le $\mathbf{h_i}$ relatif à Objet-ImageNet semble mieux reconstruit par le DAE, bien que cette connaissance semble moins pertinente que les autres, puisque son domaine ne contient que très peu de visages.

Ainsi, le choix de l'AE semble ici pertinent, bien que des approches plus complexes pourraient s'avérer efficaces, telles que l'utilisation de GAN, de Hourglass [208] ou U-NET [238] (qui consiste à connecter les couches cachées de l'encodeur et du décodeur).

La représentation $\mathbf{h_g}$ permet donc d'obtenir une bonne reconstruction des informations contenues dans les différentes connaissances. Néanmoins, une rRMSE basse ne garantie pas que la représentation apprise est suffisamment générale pour s'appliquer à de nouvelles connaissances.

Qualité du transfert vers les connaissances cibles C'est pourquoi nous proposons maintenant d'extraire $\mathbf{h_g}$ pour chaque visage des domaines des connaissances cibles, puis d'entraîner un MLP pour chacune des tâches, *i.e.* d'évaluer le transfert des connaissances sources vers les connaissances cibles.

Avant cela et afin de pouvoir évaluer les bénéfices apportés par $\mathbf{h_g}$, nous avons appliqué une approche de référence, qui consiste à entraîner directement un MLP avec une architecture identique à celle de $\mathcal{R}$ sur la concaténation des $\mathbf{h_i}$, sans entraînement préalable sur d_g.

La Table 5.3.2 nous permet ainsi de comparer cette approche de référence à utiliser $\mathbf{h_g}$ comme entrée (et qui peut être extrait par la méthode de PCA ou par notre AE). La représentation obtenue par PCA obtient des performances relativement proche de l'approche de référence, bien qu'étant dépassée sur certaines connaissances. En revanche, l'utilisation de $\mathbf{h_g}$ en sortie de l'AE permet d'atteindre de bien meilleures performances sur l'ensemble des connaissances, ce qui confirme l'intérêt de l'entraînement sur d_g. De plus, cette différence de performance peut également s'expliquer par le fait que l'approche de référence a un grand nombre de paramètres à entraîner sur un domaine de taille relativement faible, ce qui n'est plus le cas de la PCA et de l'AE.

Enfin, il est intéressant de comparer cette approche de *réduction* à des approches de sélection, évoquées dans l'introduction (en section 5.1). En effet, les auteurs de Taskonomy [309] proposent par exemple une méthode pour sélectionner automatiquement la meilleure combinaison de tâches sources pour une tâche cible donnée. Nous avons choisi de reproduire cela de manière naïve, en supposant que cette méthode de sélection est parfaite et en testant donc tous les transferts possibles et en retenant soit une unique connaissance source (6 possibilités, avant-dernière colonne de la Table 5.3.2), soit une combinaison optimale de connaissances sources (63 possibilités).

Nous pouvons noter que la méthode de sélection par transfert unique obtient des performances comparables à la PCA, tandis la méthode de sélection de combinaison optimale de transfert présente des performances comparables ou inférieures à celles de l'AE.

Cela rejoint une idée évoquée en introduction, suivant laquelle il existerait un bénéfice à chercher à utiliser l'ensemble des $\mathbf{h_i}$ plutôt que de prendre le risque d'en rejeter certains en perdant alors des informations pertinentes. Mais dans notre cas particulier où les $\mathbf{h_i}$ présentent d'importantes redondances,

Connaissances	Métrique	Réduction			Sélection	
		Concat. des $\mathbf{h_i}$	$\mathbf{h_g}$ par PCA	$\mathbf{h_g}$ par AE	Unique	Combinaison
AgeReg-UTK	MAE	4.45	4.68	**4.39**	4.70	4.54
AgeClassif-UTK	Accuracy	65.50	65.00	**68.80**	63.2	67.20
Genre-UTK	Accuracy	96.92	96.7	**97.20**	96.50	97.10
AgeReg-FG	MAE	3.22	3.18	**2.85**	**2.85**	**2.85**
Expression-SFEW	Accuracy	45.70	32	52.10	52.20	**53.1**
Expression-RAF	Accuracy	84.89	81.9	**86.51**	85.48	85.74
Douleur-UNBC	MAE	0.69	0.6	0.54	0.53	**0.51**
Ethnie-UTK	Accuracy	86.65	62.5	**89.20**	83.40	86.20
Identité-LFW	Accuracy	88.10	96.8	99.10	99.65	**99.7**

TABLE 5.3.2 – Résultats obtenus par différentes approches pour transférer les connaissances sources vers les connaissances cibles.

puisque toutes issues de connaissances en analyse du visage, cette idée n'est pas applicable en concaténant simplement les $\mathbf{h_i}$, conduisant à une trop grande dimensionnalité.

Ainsi, réduire la dimension permet de préserver de l'information issue de chacune des représentations (si nécessaire), tout en évitant le problème de la trop grande dimensionnalité qu'une concaténation impose.

Lien entre taille de $\mathbf{h_g}$ et performances Comme évoqué en fin de Section 5.3 et illustré par la Figure 5.3.3, il existe une variation importante de performance en fonction de la dimension de $\mathbf{h_g}$. Ainsi, une dimension trop faible implique souvent des performances plus faibles, du fait d'une perte de l'information, tandis qu'une dimension trop élevée semble conduire à des performances relativement bonnes mais au prix d'un coup très élevé. Nous pouvons noter de plus que le choix d'une taille de représentation optimale permettant de satisfaire toutes les connaissances cibles ne semble pas exister. Par exemple la performance optimale correspondrait à une dimension de 350 de $\mathbf{h_g}$ pour Douleur-UNBC et de 1565 pour Ethnie-UTK.

Il est donc difficile de régler ce paramètre, bien que les performances optimales correspondent à une taille moyenne de représentation proche de 1024, initialement choisie. Notons l'exception du cas de Identité-LFW, qui peut s'expliquer par le protocole d'évaluation particulier, puisque l'évaluation ayant lieu pair à pair (*cf.* Table 5.2.2), aucun apprentissage n'est effectué à partir de $\mathbf{h_g}$.

Nombre de paramètres Nous avons donc pu constater dans cette section que choisir un AE comme opérateur de réduction s'avère efficace et permet d'obtenir de très bonnes performances, confirmant une bonne généralisation de la représentation $\mathbf{h_g}$ et effectuant un premier pas vers une connaissance "générale" de l'analyse de visage. Il est néanmoins important de noter que cette approche ne satisfait pas à une application dans le monde réel. En effet, le nombre de paramètres des encodeurs reliés aux connaissances sources est de 105.8 millions (détaillés en Table 5.2.1), auxquels viennent s'ajouter les 25 millions de paramètres de l'encodeur $\mathcal{R}$ permettant d'obtenir la représentation $\mathbf{h_g}$. Un tel nombre de paramètres implique un temps de calcul conséquent et limite donc les bénéfices de l'approche pour la communauté.

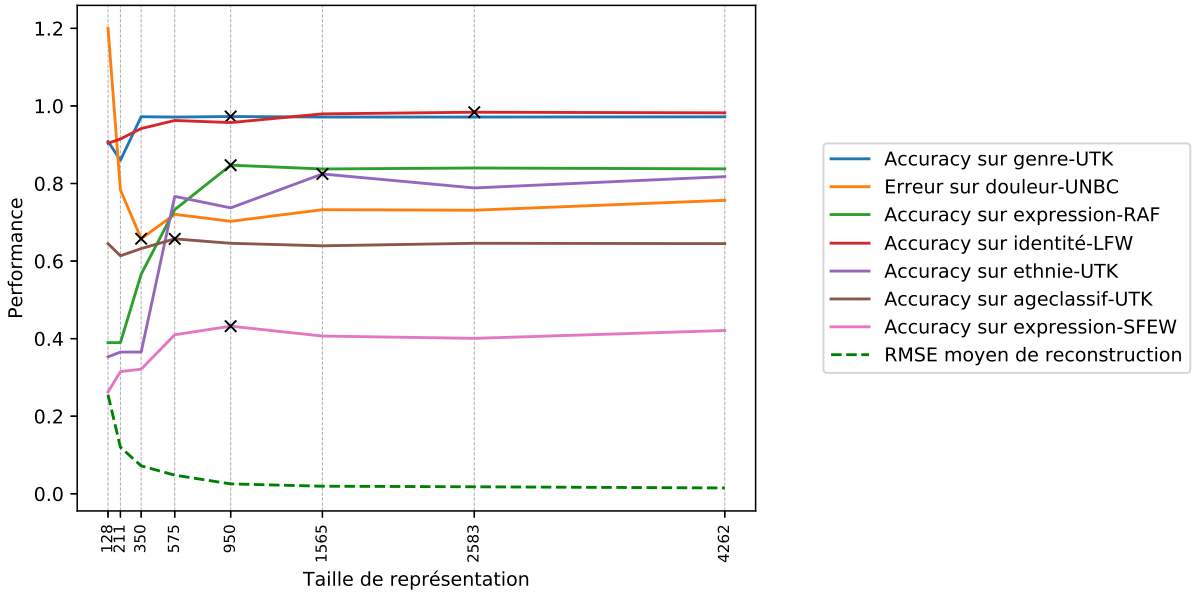


FIGURE 5.3.3 – Variations des performances sur différentes connaissances cibles et de la reconstruction des connaissances sources en fonction de la taille de la représentation $\mathbf{h_g}$. Les croix noires signalent la taille de représentation aboutissant à la meilleure performance pour chacune des connaissances cibles. Courbes réalisées à partir de 8 tailles de représentations.

5.4 Transfert des connaissances

Pour résoudre le problème du nombre de paramètres trop élevé de la section précédente, nous avons fait le choix d'utiliser une approche de distillation [123], *i.e.* nous avons entraîné un nouvel encodeur unique $\mathcal{E}^g_{unique}$, prenant une image x en entrée et cherchant à reproduire au mieux la représentation $\mathbf{h_g}$ produite par les six encodeurs $\mathcal{E}^S_i$ et l'AE. Nous revenons dans cette section sur le fonctionnement et l'intérêt d'une telle approche. Puis nous proposons différentes validations de notre méthode permettant de regrouper à la fois les connaissances sources et les connaissances cibles en un unique modèle performant.

5.4.1 Distillation pour un modèle unique et compacte

L'approche de distillation adoptée est décrite à travers la Figure 5.4.1. L'idée principale est de reprendre le modèle décrit en Section 5.3, composé des $\mathcal{E}^S_i$ et de l'AE, déjà entraîné. Il permet alors d'extraire une représentation $\mathbf{h_g}$, qui, comme nous avons pu le constater, présente de bonnes propriétés pour une reconstruction des connaissances sources tout comme pour un transfert vers les connaissances cibles.

Nous pouvons alors entraîner $\mathcal{E}^g_{unique}$ à directement extraire à partir d'une image $\mathbf{x}$ une représentation $\hat{\mathbf{h}}_\mathbf{g}$ proche de $\mathbf{h_g}$. Nous pouvons décrire ce processus de distillation en considérant que l'ensemble $\mathcal{E}^S_i$ et de l'AE est un *professeur* et que $\mathcal{E}^g_{unique}$ est un *élève*.

En pratique, nous avons choisi un ResNet-18 [122], architecture connue pour ses bonnes performances en analyse de visages et présentant un nombre de paramètres réduits. Notons que ce choix est arbitraire et que d'autres modèles de réseaux auraient pu être choisis.

La dernière couche du ResNet-18 (avant classification) étant de dimension 512, nous avons fait le choix d'ajouter un perceptron projetant cette couche vers une représentation $\hat{\mathbf{h}}_\mathbf{g}$ de taille 1024 (dimension de $\mathbf{h_g}$). Ainsi, $\mathcal{E}^g_{unique}$ est composé de l'encodeur d'un ResNet-18, suivi d'un perceptron.

Nous cherchons alors à minimiser la fonction de coût :

$$\mathcal{L}_{Distillation} = D_c(\mathbf{h_g}, \hat{\mathbf{h}}_\mathbf{g}) \tag{5.4.1}$$

avec D_c la similarité cosinus (*i.e.* $D_c(A,B) = \frac{A.B}{||A||.||B||}$).

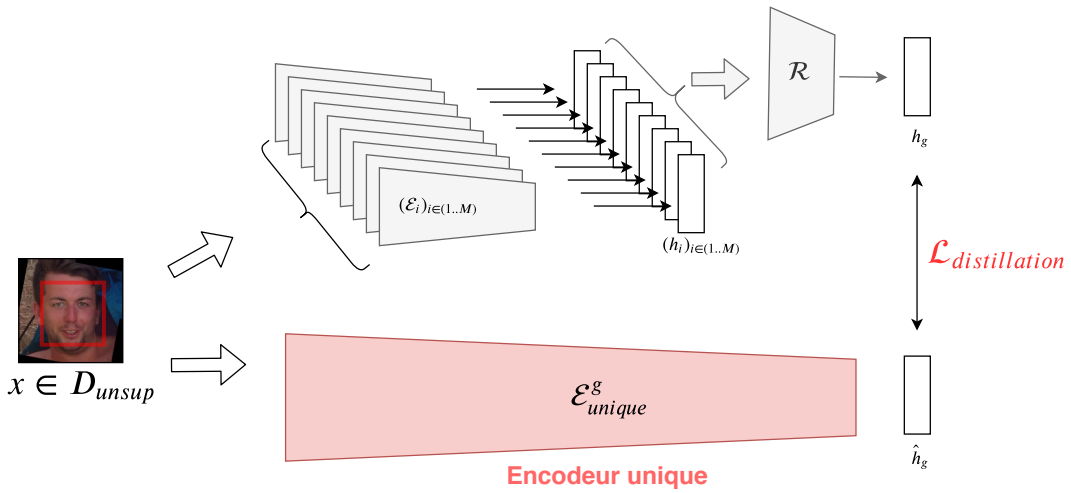


FIGURE 5.4.1 –

Enfin, une fois l'encodeur $\mathcal{E}^g_{unique}$ entraîné sur d_g, il suffit d'extraire $\hat{\mathbf{h}}_\mathbf{g}$ pour chaque image du domaine de la connaissance cible, puis d'entraîner un MLP (de la même manière que nous l'avons fait pour l'AE).

De plus, il est possible de raffiner l'ensemble du modèle, ce qui peut amener à une meilleure adaptation au domaine cible [67]. Lors du processus d'évaluation, nous parlerons d'adaptation pour désigner le fait de ré-entraîner tous les paramètres.

Pourquoi une distillation et non une compression classique de modèle ? En effet, il existe de nombreuses méthodes différentes pour "compresser" un modèle de manière à le faire fonctionner en temps réel. Par exemple, la littérature récente distingue 4 grandes catégories [64] : élagage et/ou partage de paramètres, factorisation à rang faible, filtres convolutifs compacts et distillation.

Dans notre cas, les approches de distillation, comparées aux autres catégories, permettent notamment de choisir n'importe quel modèle pour l'encodeur $\mathcal{E}^g_{unique}$. De plus, une fois cet encodeur $\mathcal{E}^g_{unique}$ obtenu, il est possible d'effectuer une adaptation de domaine classique en ré-entraînant facilement l'ensemble du modèle sur un nouveau domaine [67], ce qui peut s'avérer beaucoup plus difficile avec les six encodeurs et l'AE. Notons également qu'en disposant d'une architecture connue, il est plus facile d'appliquer certaines techniques de régularisation telles que l'augmentation de données.

Enfin, le fait que le modèle élève puisse dépasser le modèle professeur est une idée qui n'est pas nouvelle. Par exemple, les FitNets proposés par Romero *et al.* [237], sont des modèles élèves plus profonds que leur professeur et qui atteignent alors de meilleures performances. Notre modèle professeur n'est finalement que peu profond, puisque seul l'AE est entraîné sur d_g, tandis que l'encodeur $\mathcal{E}^g_{unique}$ peut disposer d'une profondeur bien plus importante, partant de l'image x.

5.4.2 Lien avec une approche multi-tâche

La présentation de cette méthode peut alors amener à un questionnement important : n'aurait-il pas été plus simple d'entraîner directement $\mathcal{E}^g_{unique}$ à prédire les $(h_i)_{i\in(1..M)}$ (voire les sorties finales des modèles pré-entraînés) sur d_g, en suivant donc une approche dite MT.

Nous verrons en pratique (*cf.* Table 5.4.2) qu'une telle approche MT, avec exactement la même architecture (y compris la même architecture de MLP que le décodeur $\tilde{\mathcal{R}}$ de $\mathcal{R}$ pour prédire les $\mathbf{h_i}$), n'aboutit pas à d'aussi bons résultats. Cela peut venir de difficultés supplémentaires à faire converger

l'ensemble, comparé à une méthode en deux étapes. Cela rejoindrait les approches progressives [145], consistant à ajouter petit à petit les différentes couches du modèles lors de l'entraînement et assurant une (meilleure) convergence qu'un entraînement direct et simultané de toutes les couches.

5.4.3 Validation expérimentale

Nous étudions dans cette section les différentes propriétés de l'encodeur $\mathcal{E}^g_{unique}$, à la fois en comparaison avec l'AE, son *professeur* mais aussi avec des méthodes de l'état de l'art spécialisées à chaque fois sur une des connaissances cibles.

Comparaison avec l'AE et connaissances sources Un premier aspect important est de savoir si la représentation $\hat{\mathbf{h}}_\mathbf{g}$ extraite par $\mathcal{E}^g_{unique}$ permet elle aussi de restituer correctement les connaissances sources. Pour cela, nous proposons d'entraîner un décodeur (identique à celui de l'AE) à prédire les h_i à partir de $\hat{\mathbf{h}}_\mathbf{g}$. Ainsi, la Table 5.4.1 rapporte les rRMSE obtenus par l'AE (pour comparaison), par la représentation extraite par $\mathcal{E}^g_{unique}$ et par l'approche MT (qui consiste, comme expliqué dans la sous-section précédente, à apprendre directement un modèle composé de la même architecture que $\mathcal{E}^g_{unique}$, mais avec un décodeur comme celui de l'AE et prédisant donc directement les h_i, en multi-tâche). Nous pouvons tout d'abord constater que le MT n'a pas réussi à converger sur toutes les tâches des connaissances sources, obtenant une reconstruction plus mauvaise que le hasard (qui a un score de 1.0) sur Identité-MS et Genre-IMDb. Dans le cas de $\mathcal{E}^g_{unique}$, les performances sont beaucoup plus uniformes d'une connaissance à l'autre et sont, pour plusieurs d'entre elles, assez proches de celles obtenues par le h_g extrait par l'AE. Le score moyen de $\mathcal{E}^g_{unique}$ reste tout de même éloigné de celui de l'AE, montrant que $\hat{\mathbf{h}}_\mathbf{g}$ a donc été apprise avec un certain bruit. Ce bruit pourrait présager une perte de performance sur les connaissances cibles, tout comme un gain de généralisation.

Conaissance	AE	MT	$\mathcal{E}^g_{unique}$
Expression-AffectNet	0.23	0.55	0.39
Identité-MS	0.22	1.12	0.35
Objet-ImageNet	0.26	0.67	0.44
Age-IMDb	0.19	0.77	0.38
Attributs-CelebA	0.27	0.97	0.35
Genre-IMDb	0.25	1.22	0.30
Moyenne	0.24	0.92	0.37

TABLE 5.4.1 – rRMSE obtenues sur l'ensemble de test de d_g pour chacune des connaissances sources et en moyenne par l'AE, l'approche MT et $\mathcal{E}^g_{unique}$

Puisqu'il est difficile d'analyser à ce stade son impact, nous proposons donc d'évaluer $\hat{\mathbf{h}}_\mathbf{g}$ sur les connaissances cibles, de manière à pouvoir comparer la capacité de généralisation de $\hat{\mathbf{h}}_\mathbf{g}$ et $\mathbf{h}_\mathbf{g}$. En consultant alors la Table 5.4.2, et plus particulièrement la colonne ($\mathcal{E}^g_{unique}$,$\hat{h}_g$) et la colonne h_g, nous pouvons observer que les résultats obtenus par les deux représentations sont globalement similaires, $\mathbf{h}_\mathbf{g}$ étant légèrement plus performante sur Attributs-CelebA, AgeReg-FG, Douleur-UNBC, Ethnie-UTK et Identité-LFW, tandis que $\hat{\mathbf{h}}_\mathbf{g}$ l'est sur AgeReg-UTK, Genre-UTK, Expression-SFEW et Expression-RAF. Ainsi, le bruit qui sépare $\hat{\mathbf{h}}_\mathbf{g}$ et $\mathbf{h}_\mathbf{g}$ possède un impact modéré en termes de transfert vers les connaissances cibles. Cela peut également s'expliquer par le fait que les représentations $\mathbf{h_i}$ que nous avons cherché à reconstruire sont elles-mêmes bruitées, puisque les modèles pré-entraînés associés aux connaissances sources ne sont pas parfaits.

Comparaison avec l'état de l'art Nous avons donc illustré que $\mathcal{E}^g_{unique}$ permet d'atteindre également de bonnes performances sur les connaissances cibles et cela avec un nombre de paramètres bien plus faible. Nous proposons maintenant d'étudier les performances obtenues en comparaison de diverses autres méthodes, chacune dédiée et à l'état de l'art pour une connaissance cible uniquement.

Pour cela, nous proposons dans la Table 5.4.2, en plus des résultats de l'état de l'art spécifique, d'étudier différentes approches. Ainsi, l'approche de référence CNN correspond à un CNN possédant la même architecture que $\mathcal{E}^g_{unique}$ (ResNet-18), mais directement entraîné sur la connaissance cible (ce qui permettra de vérifier le bénéfice apporté par la distillation sur $\mathcal{D}$).

Ensuite, la colonne MT correspond à l'approche MT décrite dans les deux sous-sections précédentes et $\mathcal{E}^g_{unique}$ à notre réseau distillé. Notons que pour chacune des deux méthodes, nous proposons deux sous-colonnes : la première correspondant à l'entraînement d'un MLP à partir de la représentation $\hat{\mathbf{h}}_\mathbf{g}$ – **MT** ou $\hat{\mathbf{h}}_\mathbf{g}$ extraite au préalable de l'image, et la seconde correspondant à une adaptation de tous les paramètres sur le domaine de la connaissance cible.

Enfin, la colonne $\mathbf{h_g}$ rappelle les scores obtenus par la représentation extraite par l'AE.

Notons que le cas d'Identité-LFW est particulier, car une adaptation de domaine n'est pas autorisé par le protocole que nous avons choisi. C'est pourquoi les modèles MT-Adaptation et $\mathcal{E}^g_{unique}$-Adaptation ont leurs scores entre parenthèse et ont été préalablement ré-entraînés sur 100 000 visages de VGGFace2 [50] en classification de l'identité.

La première observation que nous pouvons faire est que l'approche de référence CNN est dépassée par toutes les autres et sur toutes les connaissances cibles, conduisant même à une absence de convergence dans certains cas (*e.g.* AgeReg-FG avec un domaine de seulement 1002 visages). Cela valide l'intérêt d'une approche par transfert de connaissances pour l'acquisition de toutes ces connaissances cibles.

Si nous comparons maintenant les performances entre MT et $\mathcal{E}^g_{unique}$, nous pouvons confirmer que la différence de performance sur les connaissances sources préalablement observées dans la Table 5.4.1 se propage aux connaissances cibles. De plus, la même comparaison peut être faite entre $\hat{\mathbf{h}}_\mathbf{g}$ – **MT** et $\mathbf{h_g}$, ce qui confirme l'intérêt d'une bonne convergence vers les connaissances sources si l'on souhaite obtenir une bonne généralisation sur les connaissances cibles.

Un autre point important vient de l'intérêt de l'adaptation de domaine (colonnes MT-Adaptation et $\mathcal{E}^g_{unique}$-Adaptation), qui améliore toujours les performances (exception faite de AgeReg-FG, où une légère dégradation est observée, pouvant s'expliquer par la très faible taille du domaine). Cette amélioration s'explique notamment par la possibilité d'adapter plus de paramètres à la spécificité du domaine cible, mais également par l'utilisation plus aisée de technique d'augmentation des données.

Le modèle $\mathcal{E}^g_{unique}$-Adaptation finalement obtenu dépasse donc presque toujours son professeur (colonne $\mathbf{h_g}$), sauf dans le cas très particulier de AgeReg-FG. Il est alors pertinent de comparer ces résultats à ceux de méthodes état de l'art (qui utilisent pour la majorité des transferts depuis une connaissance bien choisie et des aides à la convergence spécifiquement adaptées à la connaissance cible).

Connaissance	Métrique	CNN	MT		$\mathcal{E}^g_{unique}$		$\mathbf{h_g}$	État de l'art (# paramètres)
			$\hat{\mathbf{h}}_\mathbf{g}$ – **MT**	Adaptation	$\hat{\mathbf{h}}_\mathbf{g}$	Adaptation		
Attributs-CelebA	Erreur	8.60	8.12	8.04	7.81	7.67	7.70	**7.02 [49]**(16 M)
AgeReg-UTK	MAE	6.38	4.70	4.42	4.24	**4.05**	4.39	5.39 [51](21.8 M)
AgeClassif-UTK	Accuracy	57.80	67.90	68.40	68.80	**70.40**	68.80	70.10 [73](5 M)
Genre-UTK	Accuracy	90.00	93.15	94.4	97.58	97.90	97.20	**98.23 [73]** (5 M)
AgeReg-FG	MAE	11.10	3.95	3.12	2.95	3.05	**2.85**	**2.81-3.00** [241] [13] (25 M)
Expression-SFEW	Accuracy	22.00	50.20	52.20	54.00	**57.2**	52.10	**55.40-58.14 [2]** (1.7 M/5 M)
Expression-RAF	Accuracy	69.00	82.40	87.20	87.30	**89.3**	86.51	86.77 [312] (35 M)
Douleur-UNBC	MAE	0.89	0.64	0.72	0.56	0.52	0.54	**0.51 [322]** (0.0001 M)
Ethnie-UTK	Accuracy	62.20	81.20	82.20	88.2	**91.20**	89.20	90.10 [73] (5 M)
Identité-LFW	Accuracy	(98.40)	94.27	(99.1)	98.92	(99.42)	99.10	**99.65-99.87** [248](25 M)

TABLE 5.4.2 – Résultats obtenus par différentes méthodes sur les connaissances cible. $\mathcal{E}^g_{unique}$ est composé de 2.2 millions de paramètres.

Nous pouvons vérifier que $\mathcal{E}^g_{unique}$-Adaptation atteint alors des performances proches voire supérieures à ces méthodes dédiées, adressant ainsi une large variété de connaissances cibles et validant sa bonne capacité de généralisation.

Retour aux sources Nous avons observé que le $\mathcal{E}^g_{unique}$-Adaptation permet d'atteindre des performances encore meilleures que le modèle professeur sur les connaissances cibles. Néanmoins, lors des premières comparaisons sur la reconstruction des connaissances sources (début de cette sous-section), les scores obtenus par l'AE étaient tout de même meilleurs que ceux obtenus par un décodeur appliqué à $\hat{\mathbf{h}}_\mathbf{g}$. Ainsi, une question importante est de savoir si cette différence est encore visible en termes de performance sur les connaissances sources, avec de véritables annotations et non les $\mathbf{h_i}$.

Connaissance	Source	$\mathbf{h_g}$	$\mathcal{E}^g_{unique}$-Adaptation
Expression-AffectNet	63.5	64.0	**64.4**
Attributs-CelebA	8.03	7.7	**7.67**
Genre-UTK	96.5	97.2	**97.9**
AgeReg-IMDb	**2.85**	**2.85**	3.05
Identité-LFW	**99.65**	99.1	99.42
Objet-TinyImageNet	**76.2**	71.8	56.5

TABLE 5.4.3 – Performance des modèles pré-entraînés des connaissances sources, du professeur et de l'élève sur les connaissances sources.

Nous proposons donc à travers la Table 5.4.3 une comparaison entre les scores obtenus sur les connaissances sources par les modèles des connaissances sources (colonne "Source"), la représentation $\mathbf{h_g}$ et $\mathcal{E}^g_{unique}$-Adaptation. Sur les connaissances relatives à l'expression faciale, les attributs et le genre, $\mathcal{E}^g_{unique}$-Adaptation donne de meilleurs résultats que h_g, qui donne elle-même de meilleurs résultats que les modèles sources. Ainsi, cela valide à la fois l'intérêt d'une adaptation de domaine et de la recherche d'une connaissance générale, pouvant ensuite bénéficier à la fois sur les connaissances cibles mais aussi sur les connaissances sources. Néanmoins, pour AgeReg-FG (entraînement sur IMDb), la performance est légèrement dégradée pour $\mathcal{E}^g_{unique}$-Adaptation et aucune amélioration n'est visible par h_g, ce qui pourrait éventuellement être dû à une saturation de la performance sur cette connaissance qui possède un domaine de très faible taille. Il est également intéressant de rappeler qu'un MAE inférieur à 3 correspond à un score bien meilleur que ce dont l'être humain est capable (MAE de 4.6 [118]). De même, dans le cas de l'identité l'apprentissage d'une représentation générale ne semble pas avoir conduit à une amélioration de la performance. La dégradation reste tout de même marginale et le modèle original de l'identité compte tout de même 25 millions de paramètres et une représentation de taille 2048, ce qui peut expliquer qu'il encode plus d'information relative à l'identité.

Enfin, nous avons étudié le cas particulier d'Objet-ImageNet (entraînement et évaluation sur TinyImageNet [303] du fait de la taille difficile à gérer d'ImageNet), particulier dans le sens où le domaine général d_g ne contient que des images de visages et spécialise donc indirectement la représentation $\mathbf{h_g}$ à ce type de contenu. De manière assez logique, une dégradation comparée modèle source est observée. De plus, $\mathbf{h_g}$ garde tout de même de bonnes performances, ce qui n'est pas le cas de $\mathcal{E}^g_{unique}$-Adaptation avec 56.5% d'accuracy, à peine plus performant qu'un CNN entraîné directement sur TinyImageNet et qui donnera 52% d'accuracy. Cette différence pourrait s'expliquer par l'impact plus important du contenu de d_g sur $\mathcal{E}^g_{unique}$, qui considère directement l'image $\mathbf{x}$ en entrée et se spécialise donc pour ce type d'entrée au contraire des $\mathcal{E}^S_i$ qui sont figés et conservent donc leur comportement d'origine et permettent à l'AE de garder certains éléments pertinents relatifs à ImageNet.

Paramètres Nous avons déjà expliqué en fin de Section 5.3 et avec la Table 5.2.1 que l'approche utilisant l'ensemble des $(\mathcal{E}^S_i)_{i=1..6}$ combiné à un AE présentait le défaut d'un grand nombre de paramètres (130 millions), rendant complexe une conservation du modèle en mémoire, tout comme une application en temps réel. Le modèle final que nous proposons, $\mathcal{E}^g_{unique}$-Adaptation, permet d'atteindre de meilleures performances mais présente également l'avantage d'utiliser 2.2 millions de paramètres, permettant l'usage d'un unique GPU. Il est également intéressant d'observer que les méthodes de l'état de l'art pour les différentes connaissances cibles utilisent souvent un bien plus large nombre de paramètres, impliquant donc des contraintes plus lourdes en termes d'application et possiblement une moins bonne

généralisation.

Temps d'entraînement La Figure 5.4.2 illustre également le temps de calcul nécessaire pour l'entraînement de 4 méthodes. CNN, l'approche de référence consistant à entraîner un nouveau CNN pour chacune des connaissances cibles, reste la méthode la moins coûteuse, mais s'avère également beaucoup moins efficace.

La méthode nommée "Sélection" correspond à la méthode de sélection de la meilleure combinaison de connaissances sources décrite dans la sous-section 5.3.3 et s'avère extrêmement coûteuse dès lors qu'il existe plusieurs connaissances cibles à acquérir.

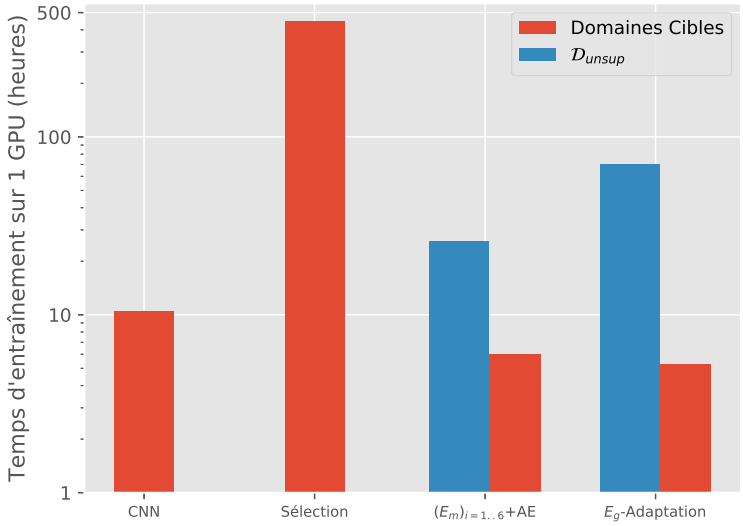


FIGURE 5.4.2 – Temps d'entraînement en heures (échelle logarithmique) avec un GPU P-100 pour différentes méthodes présentées dans cette section. Nous distinguons la phase d'entraînement non supervisée sur d_g en bleu et la phase d'entraînement pour chacune des connaissances cibles en rouge.

Ensuite, nous reportons les temps d'entraînement de la combinaison $(\mathcal{E}_i^S)_{i=1..6}$ et AE et de $\mathcal{E}^g_{unique}$-Adaptation. Le temps passé sur les connaissances cibles est alors plus faible que dans le cas même du CNN, le pré-entraînement permettant de consacrer la convergence à une simple adaptation au domaine cible. En revanche, un nombre d'heures important est consacré à l'apprentissage non supervisé sur d_g (en bleu). Notons tout de même que l'entraînement sur d_g de $\mathcal{E}^g_{unique}$-Adaptation comptabilise à la fois le processus de distillation mais également l'entraînement préalable de l'AE. Les méthodes de l'état de l'art ne comptabilisent probablement pas autant d'heures d'entraînement pour un modèle mais demande beaucoup de temps en termes de préparation, puisque à chaque connaissance cible une méthode très spécifique est associée. Ainsi, nous pouvons considérer que le fait d'utiliser $\mathcal{E}^g_{unique}$-Adaptation permet de gagner du temps dans la résolution d'une grande diversité de tâches d'analyse de visage, sur différents domaines.

Enfin, en comparaison avec une recherche exhaustive des solutions de transfert, $\mathcal{E}^g_{unique}$-Adaptation est non seulement au moins aussi performant, mais présente également un temps d'entraînement total 4 fois moins important. Dans le cas où le nombre de connaissances sources et le nombre de connaissances cibles à acquérir augmenteraient, un entraînement sur d_g se révélerait alors de plus en plus rentable. Par exemple, en doublant le nombre de connaissances cibles à acquérir, nous doublerions l'écart de temps de calcul entre la méthode de "sélection" et $\mathcal{E}^g_{unique}$-Adaptation.

5.5 Conclusions

5.5.1 En résumé

À travers ce chapitre, nous avons construit un problème de transfert de connaissances multisource et multicible, dans le contexte de l'analyse faciale. Pour cela, nous avons tout d'abord défini la notion de connaissance comme un couple (domaine, tâche). À partir de cette notion, nous avons pu associer à une connaissance donnée un modèle entraîné (sur le domaine et par la tâche). Nous avons alors proposé comme problématique le fait d'utiliser différentes sources de connaissances, *i.e.* différents modèles pré-entraînés, afin d'apprendre une connaissance générale, *i.e.* un unique modèle, permettant à la fois de restituer les connaissances sources mais également d'acquérir de nouvelles connaissances cibles. Pour cela, nous avons alors identifié deux étapes essentielles : le rassemblement de l'ensemble des connaissances sources et la création d'un modèle unique.

Pour la première étape, nous avons entraîné de manière non supervisée, sur un large corpus (noté d_g) de visages, un opérateur de réduction de la dimensionnalité, ayant pour tâche de rassembler les représentations $\mathbf{h_i}$, issues des six modèles associés aux six connaissances sources, en une unique représentation $\mathbf{h_g}$, six fois plus compacte. Nous avons alors illustré les bénéfices de cette réduction de dimensionnalité à travers les différentes propriétés de la représentation créée, celle-ci permettant de retrouver à la fois les connaissances sources, mais généralisant également bien sur les connaissances cibles. En effet, en utilisant $\mathbf{h_g}$ comme une description des visages issus du domaine d'une connaissance cible et en entraînant un MLP à résoudre la tâche associée, nous avons obtenu de bonnes performances sur une large variété de connaissances cibles liées aux visages.

Pour compenser le nombre de paramètres très importants représenté par les six modèles sources donnant les $\mathbf{h_i}$ et par l'AE les projetant en h_g, nous avons appris un unique encodeur $\mathcal{E}^g_{unique}$, prenant en entrée une image et extrayant une représentation $\hat{\mathbf{h}}_\mathbf{g}$. Pour cela, nous avons appliqué une approche de distillation, consistant à entraîner $\mathcal{E}^g_{unique}$ à extraire une représentation $\hat{\mathbf{h}}_\mathbf{g}$ la plus proche de h_g. Le modèle ainsi obtenu peut alors être utilisé comme un extracteur de description $\hat{\mathbf{h}}_\mathbf{g}$, obtenant des performances sur les connaissances sources similaires à celles obtenues avec $\mathbf{h_g}$. Mais il est également possible de ré-entraîner l'ensemble des paramètres de $\mathcal{E}^g_{unique}$ (*i.e.* seulement 2.2 millions de paramètres, comparés au 130 millions utilisés pour extraire $\mathbf{h_g}$) pour ainsi mieux adapter le modèle au nouveau domaine de la connaissance cible, ce qui permet d'améliorer encore les résultats et d'être compétitif avec des résultats de l'état dédié spécifiquement à chacune des connaissances cibles. Ainsi, le modèle unique que nous avons obtenu permet d'obtenir des performances au niveau de l'état de l'art sur une large diversité de tâches d'analyse faciale et peut être vu comme un premier pas vers une connaissance plus générale, bien que restreint à un domaine spécifique.

Notons qu'effectuer les deux étapes, réduction et distillation, s'est avéré également plus efficace que de directement apprendre un modèle unique dit multi-tâche, *i.e.* entraîné à prédire directement les $\mathbf{h_i}$ sur d_g. Cela pourrait simplement s'expliquer par la difficulté à faire converger ce modèle multi-tâche, qui mélange les deux étapes en un entraînement et n'arrive possiblement donc pas à satisfaire les deux. Notre approche en deux étapes peut alors être interprétée comme une technique progressive, déjà utilisée pour faciliter la convergence d'AE et de GAN [145].

5.5.2 Perspectives

Opérateur de réduction, architecture L'opérateur de réduction utilisé est un simple auto-encodeur. Bien qu'ayant expérimenté avec des variantes de l'auto-encodeur (en Section 5.3) et n'ayant pas obtenu de meilleurs résultats, il serait pertinent d'évaluer des méthodes plus sophistiquées, telles qu'utiliser une fonction de perte adverse, utiliser une approche "démêlante" (disentangling) [272] ou encore un réseau de type hourglass [208] ou U-net [238].

De même, l'architecture choisie pour $\mathcal{E}^g_{unique}$ est purement arbitraire et pourrait être améliorée, notamment car le ResNet-18 extrait une représentation de taille 512 que nous avons étendue linéairement

vers les 1024 de h_g, ce qui rend probablement la tâche de distillation moins facile que si nous avions utilisé un réseau aboutissant sur une couche cachée de dimension 1024.

Lien avec l'apprentissage auto-supervisé Dans la section 2.1.6, nous avons vu qu'il existe des méthodes d'entraînement dite auto-supervisées. Celles-ci consistent à formuler une tâche non coûteuse en annotation, qui permet d'apprendre un modèle qui obtiendra de bonnes performances en transfert de connaissances. Il existe par exemple un grand nombre de tâches différentes relatives aux images [318, 319, 214, 215, 106, 231], consistant à perturber celles-ci (par exemple avec une rotation) et à chercher alors à reconstruire les images d'origine.

Dans notre cas, nous avons utilisé des encodeurs $\mathcal{E}_i^S$ pour extraire différentes vues d'une même image. Ces vues pourraient être considérées comme des versions "modifiées" de l'image, au même titre que les tâches des méthodes auto-supervisées. Il serait alors intéressant de formuler comme tâche pour l'auto-encodeur non de chercher à reconstruire les $\mathbf{h_i}$ mais directement l'image d'origine, à partir de ces $\mathbf{h_i}$.

Extension des connaissances En fin de section 5.4.1, nous avons considéré le cas où nous augmenterions le nombre de connaissances sources et cibles et avons évoqué le fait que cela n'augmenterait que relativement peu le temps de calcul sur les connaissances cibles. En revanche, en disposant d'un large panel de connaissances sources, possiblement entraîné sur des domaines véritablement différents, il serait probablement nécessaire de rajouter encore plusieurs étapes, en hiérarchisant par exemple les apprentissages et reconstruction des différentes tâches sources par similarité. Cela impliquerait alors de construire au préalable un espace des tâches (comme cela a été fait par exemple dans [309]), puis de ré-appliquer une méthode similaire de manière hiérarchique, en rassemblant petit à petit les $\mathbf{h_g}$ obtenus en un unique $\mathbf{h_G}$ final.

Vers une fusion multimodale ? Nous avons évoqué en introduction la possibilité de formuler le problème abordé dans ce chapitre comme un problème multimodal, chaque représentation $\mathbf{h_i}$ apparaissant comme une modalité avec un haut niveau de sémantique, que nous fusionnons alors par simple concaténation. Nous verrons dans le Chapitre 6 que des architectures de fusion à différents niveaux peuvent grandement améliorer les performances et qu'à chaque problème multimodal, une architecture de fusion spécifique peut être plus adaptée. Il serait donc intéressant d'utiliser différentes représentations issues des $\mathcal{E}_i^S$ et d'utiliser les techniques de fusion, que nous allons voir par la suite, pour les combiner et en extraire une représentation $\mathbf{h_g}$ d'autant plus pertinente.

Chapitre 6

Recherche d'architectures de fusion multimodale

Table des matières

6.1 Introduction

Résoudre un problème dit multimodal consiste à tirer parti de différentes modalités ou vues d'un même signal, pour estimer celui-ci au mieux. Par exemple dans le Chapitre 3, nous avons formulé un problème multimodal concret, visant à déterminer l'émotion ressentie par une personne à partir de deux modalités, le son et l'image. Le fait de disposer de ces deux modalités permet d'obtenir plus d'information sur le signal caché qu'est l'émotion et d'ainsi obtenir de meilleurs résultats qu'en exploitant une seule modalité. Nous avons également retrouvé le concept de multimodalité dans le Chapitre 5, où nous avons cherché à obtenir une représentation compacte et générale d'un visage en se servant des représentations de différents réseaux de neurones pré-entraînés, pouvant être considérées comme autant de vues différentes du même visage.

Vers la fin du Chapitre 2, en Section 2.3, nous avons pu observer qu'il existe une grande diversité de stratégies pour fusionner deux modalités (*e.g.* image et son). Un des choix cruciaux est la manière de fusionner les représentations unimodales. Deux choix classiques sont illustrés en Figure 6.1.1 : une fusion tardive consiste à fusionner les représentations unimodales lorsqu'elles sont proches du niveau sémantique, tandis qu'une fusion précoce va exploiter des représentations beaucoup plus proches des entrées. Enfin, une fusion hybride consisterait à utiliser à la fois des représentations proches des entrées et des représentations proches du niveau sémantique.

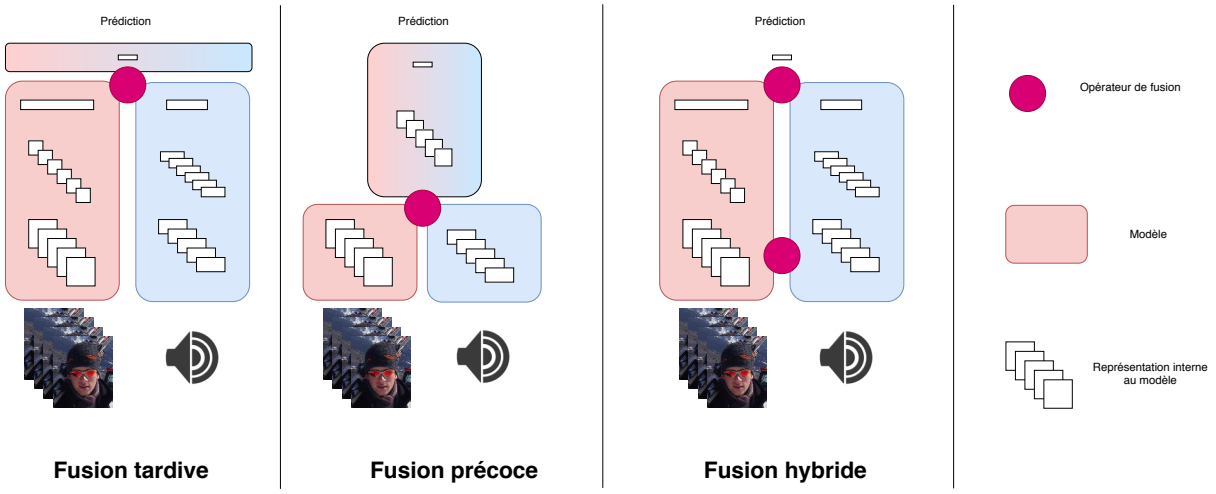


FIGURE 6.1.1 – Exemple de trois stratégies de fusion multimodale : fusion tardive, fusion précoce et fusion hybride. Figure issue du Chapitre 2

Il se trouve que cette diversité de méthode de fusion pourrait provenir du fait qu'une stratégie de fusion multimodale qui fonctionne bien sur une base de données et une tâche précise peut ne pas donner les mêmes résultats lorsqu'elle est confrontée à un nouveau problème.

Une illustration de cette dépendance au problème traité apparaît notamment dans le Chapitre 3, en section 3.3 (Table 3.3.6), lorsque différentes méthodes de fusion, qui obtiennent des résultats à l'état de l'art sur d'autres problèmes, sont testées dans le cas particulier de la reconnaissance d'émotion sur AFEW [81]. Elles n'atteignent alors pas les performances d'une méthode basique, consistant par exemple à utiliser la moyenne des prédictions obtenues par chaque modalité.

Dans ce Chapitre, nous souhaitons analyser plus en détail cette hypothèse que la stratégie de fusion à adopter dépend du problème traité. C'est pourquoi nous nous intéressons dans la section 6.2 à la simulation de "plusieurs" problèmes multimodaux, en jouant sur deux aspects : la qualité de chacune des modalités et l'information qu'elles partagent. Et en étudiant différentes architectures et techniques de fusion multimodale sur les problèmes générés, nous analysons alors plus finement cette dépendance.

Ensuite, ce problème d'adaptation de la stratégie est traité à travers deux solutions. Dans la section 6.3, les bénéfices d'une approche utilisant un réseau central de fusion à plusieurs niveaux sont

d'abord mis en avant, puis nous faisons le choix de reformuler le problème de la fusion multimodale comme un problème de recherche d'architectures à travers la Section 6.4.

6.2 Génération d'un problème multimodal et expériences préliminaires

Cette partie propose une méthode pour générer un problème multimodal paramétrable (*cf.* Section 6.2.1), de manière à ensuite pouvoir facilement étudier différentes méthodes de fusion face à une large variété de problèmes générés (*cf.* Section 6.2.2).

6.2.1 Génération d'un problème multimodal "jouet"

Nous cherchons dans cette partie à simuler un problème multimodal, ou plus simplement bimodal. Nous désirons donc disposer de deux modalités $\mathbf{x}$ et $\mathbf{y}$, qui sont deux vues d'un même signal s. Pour cela, plusieurs méthodes existantes [11, 206, 56, 170] proposent d'utiliser comme signaux $\mathbf{s}$ des images de chiffres manuscrits (de 0 à 9) de la base de données MNIST [167].

Une première approche [11, 56] consiste à proposer pour chaque modalité une moitié (gauche et droite) de l'image, ce qui permet d'assurer une certaine corrélation entre les deux vues. Une seconde approche [206, 170] quant à elle, suit le même principe mais en divisant l'image en quatre parties égales au lieu de deux. De plus, certaines des quatre modalités ainsi créées peuvent être bruitées au moment du test pour étudier la robustesse des modèles utilisés. Enfin, certaines approches [18] proposent de créer des données totalement synthétiques en générant des exemples où une seule des deux modalités contient de l'information (à partir d'un modèle génératif et d'une loi de Bernoulli).

Nous avons fait le choix de ré-utiliser la base de donnée MNIST comme point de départ. Notre but est de pouvoir contrôler le recouvrement de l'information entre $\mathbf{x}$ et $\mathbf{y}$, et d'également pouvoir modéliser des modalités plus ou moins bruités.

Nous avons alors cherché à exhiber les composantes indépendantes dans notre base de données. Pour cela, nous avons effectué une PCA sur les images aplaties en vecteurs de taille 784 de MNIST, ce qui nous a permis d'obtenir un ensemble de vecteurs singuliers $\mathbf{v_i}$, associés à chacune des valeurs singulières i. Ces vecteurs sont ordonnés par énergie, formant une matrice $\mathbf{V}$:

$$\begin{pmatrix} \mathbf{v_1} & \mathbf{v_2} & \dots & \mathbf{v_N} \end{pmatrix} \tag{6.2.1}$$

Il est alors possible de projeter une image donnée $\mathbf{s}$ dans l'espace des vecteurs singuliers par le produit matriciel $\mathbf{M}\ \mathbf{s}$, avec $M = VDV^T$ où D est une matrice diagonale avec des 0 et des 1. Afin de contrôler l'énergie totale contenue par les modalités et le recouvrement entre celles-ci, nous avons fait le choix de définir deux matrices $\mathbf{M_x}$ et $\mathbf{M_y}$, issues de $\mathbf{M}$, en remplaçant les vecteurs singuliers que nous ne désirons pas par des zéros.

Définissons e comme le pourcentage d'énergie totale (équitablement répartie entre les deux modalités) et r le recouvrement entre les deux modalités.

Pour obtenir une énergie proche de $\frac{e}{2}$ dans chaque modalité et un recouvrement proche de r en termes de composantes, nous attribuons les vecteurs singuliers entre $\mathbf{M_x}$ et $\mathbf{M_y}$ suivant les règles suivantes :

1. Si $r = 0$ (pas de recouvrement), soit N_e tel que $\sum_i^{N_e} E(\mathbf{v_i}) < \frac{e}{2}$ (avec $E(\mathbf{v_i})$ l'énergie associée à $\mathbf{v_i}$). Alors parmi les N_e premiers vecteurs, les indices pairs (*i.e.* $\mathbf{v_{2i}}$) sont conservés dans $\mathbf{M_x}$ et les indices impairs (*i.e.* $\mathbf{v_{2i+1}}$) dans $\mathbf{M_y}$.
2. Si $1 > r > 0$, nous définissons $N_r = rN_e$. Les N_r premiers $\mathbf{v_i}$ sont attribués à la fois à $\mathbf{M_x}$ et à $\mathbf{M_y}$, tandis que les $N_e - N_r$ restants sont attribués avec la règle précédente (pairs pour $\mathbf{M_x}$ et impairs pour $\mathbf{M_y}$). Notons qu'augmenter le recouvrement va augmenter l'énergie attribuée par modalité dans la mesure où les $\mathbf{v_i}$ partagés correspondent aux plus hautes valeurs d'énergie.

Il est important de remarquer que l'énergie et le recouvrement sont approximatifs, dans les sens où les valeurs $E(\mathbf{v_i})$ ne décroissant pas linéairement, ce qui implique que $E(\mathbf{v_{2i}})$ peut être fortement supérieure à $E(\mathbf{v_{2i+1}})$.

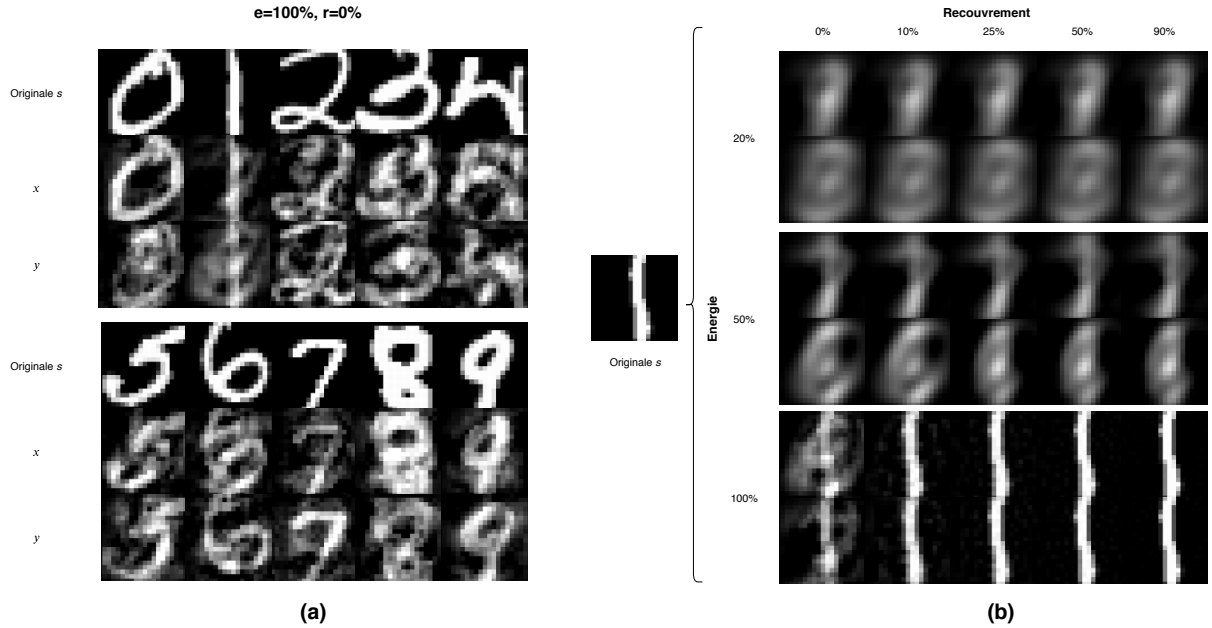


FIGURE 6.2.1 – (a) Exemple de génération des deux modalités **x** et **y** pour chacune des 10 classes de MNIST Multimodal avec 100% d'énergie totale et aucun recouvrement entre les deux modalités. (b) Exemple de génération des modalités **x** et **y** en faisant varier les valeurs d'énergie et de recouvrement.

Une fois les matrices $\mathbf{M_x}$ et $\mathbf{M_y}$ créées, l'obtention des modalités **x** et **y** est directe. **s** est projeté dans l'espace de représentation, puis **x** et **y** sont rétro-projetés à partir des matrices précédemment définies :

$$\mathbf{x} = \mathbf{M_x}\mathbf{M}^T\mathbf{s} \tag{6.2.2}$$

$$\mathbf{y} = \mathbf{M_y}^T\mathbf{M}^T\mathbf{s} \tag{6.2.3}$$

La base de données ainsi créée, que nous appellerons par la suite MNIST Multimodal (e, r), contient 70 000 paires d'images **x** et **y** de résolution 28x28 (55 000 pour l'entraînement, 5 000 pour la validation et 10 000 pour le test), associées à un label de chiffre entre 0 et 9. Un grand nombre de variations de MNIST Multimodal (e, r) peut alors être généré, en utilisant différentes valeurs d'énergie e et de recouvrement entre modalités r. Cela permet donc de simuler une variété de différents problèmes bimodaux. La Figure 6.2.1 en illustre plusieurs configurations : (a) un cas particulier pour $e = 50\%$ et $r = 0\%$ et (b) les modalités **x** et **y** issues d'une même image **s** pour diverses valeurs de e et r. Nous pouvons noter que l'effet d'un recouvrement important (*e.g.* 50%) des modalités dans le cas d'une énergie à 100% entraîne des **x** et **y** visuellement très similaires à l'image d'origine s, bien que des différences fines subsistent (invisibles pour le cas ($e = 50\%$ et $r = 90\%$)).

6.2.2 Différentes approches appliquées à MNIST Multimodal

Approche de base Une fois les différentes variations de base de données générées, nous avons choisi l'architecture LeNet-5 [167], classiquement utilisée sur MNIST, pour traiter chacune des modalités. Ce réseau de petite taille comprend deux couches convolutives (avec max-pooling), suivies de deux couches entièrement connectées. Pour assurer une meilleure convergence, nous avons modifié l'architecture originale en ajoutant une BN entre chaque couche et du masquage des connections ou *dropout* [260] (dropout) en sortie de la première couche entièrement connectée. Le réseau ainsi construit est entraîné sur les images originales de MNIST pendant 100 epochs avec un taux d'apprentissage de 0.001 et une taille de batch de 128. Il atteint une performance moyenne sur 50 expériences de 66 erreurs sur les 10 000 images de l'ensemble de test (soit 99.34% d'accuracy).

Réseau unimodal Les images **x** et **y** générées sont chacune de dimension 28x28. Ainsi, nous considérerons qu'un réseau unimodal peut être appliqué sur chacune de ces modalités, en reprenant le réseau LeNet-5 précédent.

Fusion multimodale Pour combiner les deux réseaux multimodaux, nous allons pouvoir modéliser différentes stratégies. Ainsi, nous proposons de paramétrer les différentes méthodes par deux variations possibles :

— l'opération de fusion en elle-même, qui peut être par exemple une soustraction, une somme ou un produit des deux représentations unimodales (*cf.* Chapitre 2 pour des cas d'usage de ces opérateurs).

— la profondeur à laquelle la fusion est effectuée (illustrée en partie dans la Figure 6.2.2, allant de 0 lorsque les prédictions unimodales sont fusionnées à 4 lorsque **x** et **y** sont directement fusionnés).

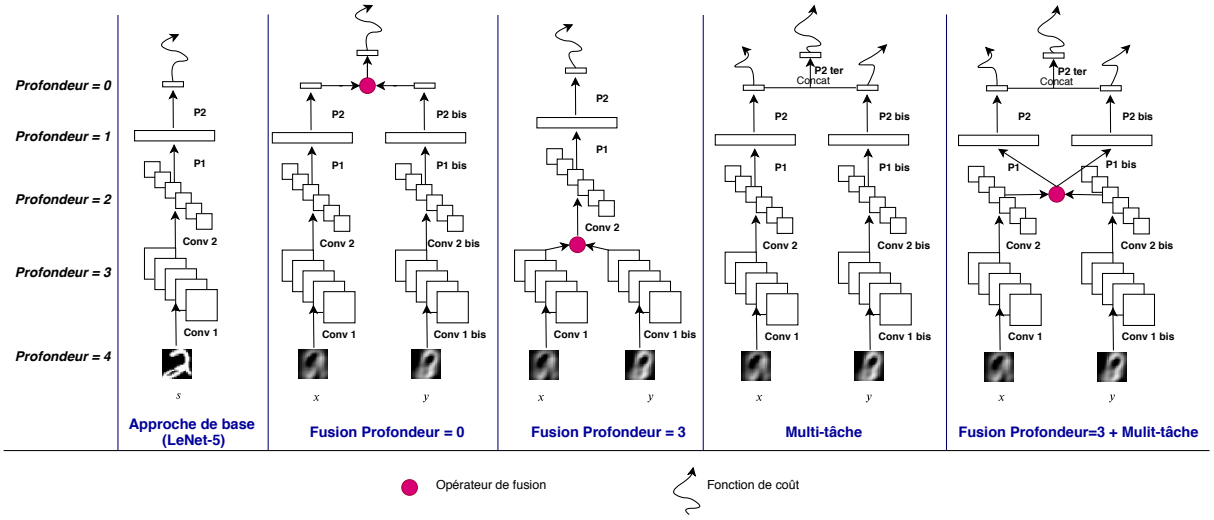


FIGURE 6.2.2 – De gauche à droite : l'approche de base, consistant en un réseau LeNet5 traitant le MNIST original, la fusion effectuée à une profondeur = 0, *i.e.* au niveau des prédictions, la fusion effectuée à une profondeur = 3, *i.e.* au niveau de la sortie de la première convolution, la méthode dite "Multi-tâche", qui consiste à rajouter deux fonctions de coût unimodales. L'opérateur de fusion peut être une somme ou un produit.

Nous préciserons par la suite la méthode de fusion utilisée en donnant systématiquement son opérateur (*e.g.* somme) et sa profondeur (*e.g.* 2).

Multi-tâche Une approche que nous avons choisi de nommer "Multi-tâche" consiste à optimiser les deux réseaux unimodaux en parallèle, chacun avec sa fonction de coût. En plus de cela, une fonction de coût générale est ajoutée pour optimiser une couche entière connectée qui prend en entrée la concaténation des prédictions unimodales et sort une prédiction finale, utilisée pour évaluer le réseau. Cela revient donc presque à une fusion de profondeur 0 mais avec un opérateur de fusion particulier (couche entièrement connectée) et l'ajout de deux fonctions de coût (multi-tâche).

Fusion + Multi-tâche L'approche Fusion et Multi-tâche consiste à combiner les deux méthodes précédentes en utilisant à la fois la fonction de coût multi-tâche mais aussi en effectuant une fusion à une profondeur donnée (*e.g.* profondeur = 3 dans la Figure 6.2.2). Cette opération de fusion permet d'obtenir une représentation multimodale, qui est dupliquée et ré-injectée en entrée des parties finales des deux réseaux unimodaux (dans la Figure 6.2.2, les couches P1 et P1-bis prennent donc en entrée la même représentation issue de la fusion qui précède). Ainsi, dans les premières couches (avant la fusion), nous

sommes dans le cas multimodal, puis dans les couches après fusion (*e.g.* P1, P1-bis et P2, P2-bis dans le cas de la Figure 6.2.2), nous sommes dans une approche de type ensembliste.

6.2.3 Expériences préliminaires

Nous nous intéressons maintenant à l'impact des variations de MNIST Multimodal sur les résultats des différentes méthodes. Tout d'abord, il est essentiel de préciser que toutes les évaluations reportées ici sont systématiquement une évaluation moyenne sur 50 entraînements avec des initialisations différentes et avec un écart-type de l'ordre de 2 erreurs sur 10 000 (ou 0.02% en accuracy). Dans une première sous-section, nous cherchons plus spécifiquement à identifier l'impact des deux paramètres (énergie et recouvrement). Puis nous profitons du cadre "contrôlé" de MNIST Multimodal pour mieux étudier les différentes méthodes présentées dans la sous-section précédente.

6.2.3.1 Variations de MNIST Multimodal

Effet de l'énergie Une première observation, cohérente avec l'effet désiré, est que plus une modalité contient d'énergie, plus le réseau qui la traite va être performant. Ainsi les deux réseaux unimodaux appris séparément sur les modalités x et y donnent des performances assez proches (dans les premières colonnes de la Table 6.2.1). Cette notion d'amélioration de la performance lorsque l'énergie augmente est également présente dans la Figure 6.2.3, où la méthode de fusion est impactée (quel que soit le recouvrement entre modalité) par l'énergie : diviser l'énergie par deux implique une importante baisse de performance (courbe orange versus courbe bleue).

Cet effet de l'énergie peut être simplement vu comme un niveau d'information contenu au sein de chacune des modalités, une énergie à 100% correspondant à l'intégralité de l'information présente (*i.e.* environ 50% dans chacune des deux modalités), tandis qu'une énergie plus basse va correspondre à une perte d'information.

Énergie	Unimodal **x**	Unimodal **y**	Multi-tâche	Fusion soustraction profondeur=2
0.2	30 %	25 %	44 %	46.2 %
0.5	61.1 %	72.5 %	88.2 %	94.1 %
1.0	97.7 %	97.5 %	98.6 %	98.8 %

TABLE 6.2.1 – Variations de l'accuracy (pourcentage de réponses correctes) sur l'ensemble de test pour les différentes méthodes à différents niveaux d'énergie et sans recouvrement d'information.

Effet du recouvrement entre modalité Le recouvrement entre modalités semble également avoir un impact sur la performance, bien que plus léger. En effet, en observant la Figure 6.2.3, la performance augmente avec le recouvrement, jusqu'à un recouvrement de 50% (tout du moins pour une énergie à 100%). Lorsque les modalités sont pratiquement identiques (90% de recouvrement), la méthode est proche d'un simple problème d'ensemble et la performance de la fusion devient légèrement plus faible. Cette baisse de performance pourrait s'expliquer par le fait que les dernières valeurs singulières trouvée par la PCA correspondent à des éléments n'aidant pas à résoudre le problème, le fait de les supprimer ayant alors un effet débruitant.

6.2.3.2 Choix d'architectures de fusion

Profondeur optimale de la fusion Une stratégie de fusion est souvent dépendante du choix de la profondeur à laquelle l'opération de fusion entre réseaux unimodaux est effectuée. La Figure 6.2.4 nous permet de modéliser les profondeurs donnant les meilleures performances (avec un écart significatif) pour différentes combinaisons d'énergie et de recouvrement. Ainsi, nous pouvons retrouver des "lignes de niveaux", dans le sens où la profondeur de fusion va diminuer lorsque l'énergie et le recouvrement augmentent conjointement. En termes de stratégies de fusion, cela revient à dire que plus nous disposons

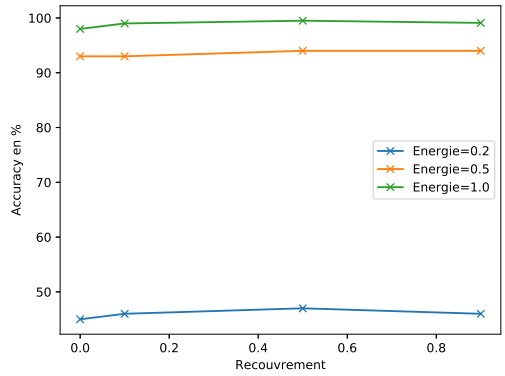


FIGURE 6.2.3 – Représentation des résultats obtenus pour une méthode de fusion (opérateur somme) de profondeur=2 en fonction du recouvrement entre modalités (en abscisses).

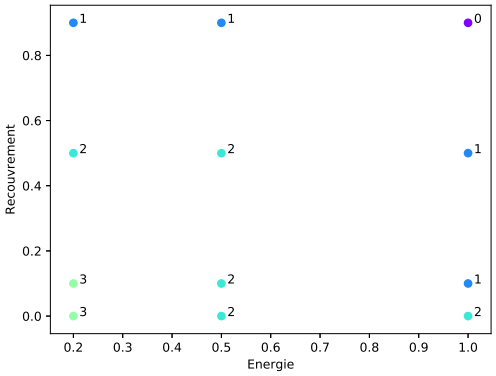


FIGURE 6.2.4 – Profondeur de fusion donnant le meilleur score suivant les valeurs d'énergie (abscisses) et de recouvrement (ordonnées) fournies. Évaluation pour la méthode fusion somme.

d'information en entrée, plus la fusion doit être tardive et vice versa. Ces observations illustrent donc que malgré des changements relativement mineurs en termes de nature, des variations inter et intra modalités peuvent impliquer des stratégies de fusion totalement différentes.

Opérateur optimal de la fusion Mis à part la profondeur, il est également intéressant d'observer l'impact d'un opérateur de fusion. Nous avons implémenté les méthodes avec les opérateurs de soustraction, de somme et de produit terme à terme. Les résultats obtenus en avant-dernier groupe de lignes de la Table 6.2.2 permettent de comparer les meilleurs résultats obtenus par ceux-ci. Les différences sont relativement minimes et nous pouvons noter que la profondeur optimale est toujours située au même niveau, ce qui laisse penser que l'impact de l'opérateur sur la profondeur à laquelle effectuer la fusion n'est que limité.

Fusion, Multi-tâche et Fusion+Multi-tâche Nous nous intéressons maintenant à l'ensemble des méthodes et les comparons dans la Table 6.2.2.

Le premier groupement de lignes correspond à des approches de base, tandis que les autres traitent MNIST Multimodal pour des valeurs d'énergie de 100% et de recouvrement de 50%. Une première observation réside dans le gain relativement important entre la Fusion et la Fusion+Multi-Tâche. Cette différence se retrouve également sur un changement de la profondeur optimale de fusion, qui pousse à une fusion très précoce (ce qui n'est pas surprenant dans le cadre d'entrées présentant beaucoup de similarités). La méthode Multi-Tâche seule donne également d'excellents résultats pour des modalités particulièrement propres et non bruitées, dépassant les méthodes de fusion dans ce cas. Notons en revanche que dans le cas de modalités incomplètes (par exemple 50% d'énergie), la méthode Multi-Tâche est largement dépassée par une méthode de fusion (resp. 88.2% versus 94.1%, *cf.* Table 6.2.1).

Ensuite, le score de l'approche de base est également intéressant, lorsqu'il est comparé aux résultats obtenus par les méthodes de fusion. En effet, celui-ci est inférieur à ceux obtenus par la méthode de Fusion+Multi-Tâche. Cela pourrait être expliqué facilement par le nombre de paramètres qui a été presque doublé ou par le peu de bruit occasionné par les paramètres élevés d'énergie et de recouvrement. Pourtant, utiliser un ensemble de deux LeNet-5 avec des initialisations différentes (appliqué aux images originales) nous a seulement permis d'atteindre un score de 99.36 %, tandis qu'un ensemble de 3 LeNet-5 plafonne à 99.40%. Cela peut être vu comme un effet débruitant de la suppression des dernières composantes de la PCA, conduisant à une amélioration de la capacité de généralisation, car en diminuant l'énergie, il pourrait être plus difficile d'exploiter des biais d'apprentissage.

Méthode		Accuracy	Profondeur
Approche de base (1 LeNet-5)		99.34	
Ensemble (2 LeNet-5)		99.36	
Ensemble (3 LeNet-5)		99.40	
Unimodal **x**		97.51	
Unimodal **y**		97.72	
Multi-Tâche		99.38	
Fusion	Soustraction	99.36	2
	Somme	99.32	0 et 2
	Produit	99.29	2
Fusion + Multi-Tâche	Soustraction	99.37	3
	Somme	99.42	4
	Produit	99.37	2

TABLE 6.2.2 – Résultats obtenus par les différentes approches pour des valeurs d'énergie de 100% et de recouvrement de 50% (excepté pour les trois premières lignes, appliquées sur les images originales). La profondeur, lorsqu'elle est précisée, correspond à la profondeur optimale.

Vers une architecture de fusion multimodale indépendante du problème Les résultats obtenus montrent dans un contexte contrôlé que (a) la stratégie de fusion multimodale dépend énormément du problème à aborder, (b) les approches multi-tâche et multimodales ne sont pas incompatibles (*e.g.* Fusion+ Multi-tâche) et (c) la fusion multimodale permet un gain particulièrement important dans le cas de modalités bruitées (signal d'origine incomplet).

Dans cette partie, nous avons recherché la profondeur de fusion optimale en testant toutes les possibilités. Comme évoqué dans le Chapitre 2, il pourrait être pertinent de chercher à apprendre à quelle(s) profondeur(s) une fusion serait la plus bénéfique, ce qui serait un premier pas vers une automatisation du choix d'une stratégie de fusion.

6.3 CentralNet : une fusion multi-niveau centralisée

A partir de l'intuition développée dans la partie précédente, il serait intéressant d'être capable d'adapter la stratégie de fusion (*i.e.* dans notre cas, nous nous restreignons à identifier les niveaux auxquelles la fusion est effectuée) à chaque problème. Pour cela, nous proposons dans cette partie de créer un modèle qui centralise les différentes représentations unimodales en des représentations multimodales.

Pour définir un tel modèle, considérons tout d'abord des travaux spécifique au domaine de la vision. Escorcia *et al*. [95] défendent l'idée que les attributs visuels (permettant de répondre à une tâche donnée) sont répandus dans l'ensemble des représentations extraites par un réseau de neurones. Ainsi, chacune des représentations cachées peut apporter potentiellement une information utile pour une tâche donnée. Or les attributs visuels nécessaires à la résolution de différentes tâches sont par nature différents : par exemple, la couleur des cheveux ne servira absolument pas à déterminer l'émotion, tandis qu'elle permettra d'aider à l'estimation de l'âge. Une manière de justifier alors l'intuition de modèles dit résiduels [122] serait de considérer qu'en conservant l'information d'une couche à l'autre, la représentation finale obtenue permettrait de considérer l'ensemble des attributs visuels utiles à la tâche, et non d'ordonner de manière sémantique les différents niveaux du réseau. Si on définit une couche neuronale classique comme une fonction p transformant une entrée $\mathbf{h_i}$ en une sortie $\mathbf{h_{i+1}}$, alors son équivalent résiduel consiste à considérer $\tilde{p} = p + id$ avec id la fonction identité, permettant de forcer p à apporter une information dite "résiduelle".

Si nous revenons à un problème dit multimodal, nous pouvons tout à fait transposer le raisonnement précédent, en considérant que nous souhaitons conserver au cours du processus de fusion l'information unimodale utile pour notre tâche, contenue dans les représentations des réseaux unimodaux. Ainsi, nous pourrions facilement définir un réseau multimodal composé de couches ayant une entrée $\mathbf{h_i}$, fusion de

deux représentations unimodales $\mathbf{x_i}$ et $\mathbf{y_i}$ et ayant pour sortie une représentation $\mathbf{h_{i+1}} + \mathbf{x_{i+1}} + \mathbf{y_{i+1}}$. Il est néanmoins nécessaire de prendre en compte l'aspect multimodal : le fait de disposer de deux modalités potentiellement très différentes peut entraîner des incompatibilités à combiner certaines représentations d'une modalité avec celles de l'autre modalité. Il devient alors intéressant de définir un système de "portes", de manière à éviter de transmettre l'information unimodale lorsqu'elle est néfaste au réseau multimodal.

Nous proposons de définir un tel modèle de fusion multimodale, que nous nommerons CentralNet. Nous nous limiterons dans la description de ce modèle à un problème de classification bimodale pour simplifier les notations, mais il est également possible d'appliquer cette méthode avec plus de deux modalités et à des problèmes de régression par exemple.

6.3.1 Construction du modèle

Formulation du problème Nous considérons donc que la fusion multimodale consiste ici à combiner les informations issues de différentes modalités et à différents niveaux. De manière plus formelle, nous définissons deux modalités $\mathbf{x}$ et $\mathbf{y}$, leurs représentations très bas niveau $\mathbf{x_0}$ et $\mathbf{y_0}$, et leurs réseaux unimodaux associés f et g, possédant $N+1$ couches, respectivement $(f_i)_{0...N}$ et $(g_i)_{0...N}$. f et g permettent alors d'extraire deux représentations unimodales finales, respectivement définies par $\mathbf{x_N} = (f_N \circ ... \circ f_{i+1} \circ f_i \circ ... \circ f_0)(\mathbf{x_0})$ et $\mathbf{y_N} = (g_N \circ ... \circ g_{i+1} \circ g_i \circ ... \circ g_0)(\mathbf{y_0})$, chacune permettant d'obtenir une décision unimodale, $\hat{\mathbf{z}}_\mathbf{x}$ et $\hat{\mathbf{z}}_\mathbf{y}$. Le but d'un procédé de fusion est alors d'obtenir une meilleure décision $\hat{\mathbf{z}}_\mathbf{c} = \hat{\mathbf{z}}_{\mathbf{x},\mathbf{y}}$, en tirant parti des deux modalités.

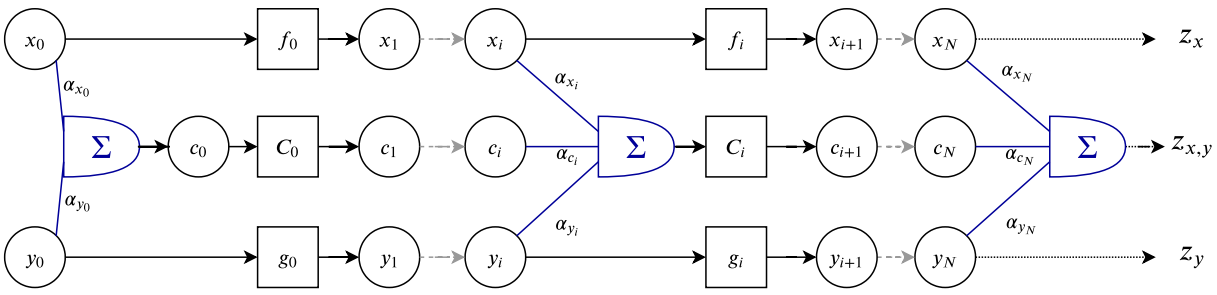


FIGURE 6.3.1 – Schéma représentant l'approche CentralNet : les réseaux f et g unimodaux voient leurs représentations x_i et y_i traitées par un réseau central C. À chaque étage i du réseau central, une pondération α est attribuée à chacune des représentations, puis la somme pondérée est traitée par le bloc suivant C_i du réseau central.

Un réseau central Nous disposons donc de (N+1) représentations unimodales pour chacune des modalités. Nous supposons qu'à chaque niveau i les représentations $\mathbf{x_i}$ et $\mathbf{y_i}$ sont de même dimensions, si nécessaire en complétant par des zéros.

Nous définissons alors un réseau central C, que nous nommerons CentralNet, illustré en Figure 6.3.1, et qui à chaque niveau prend comme entrée une somme pondérée de $\mathbf{x_i}$ et $\mathbf{y_i}$, ainsi que sa propre représentation c_i issue de la couche précédente. Notons l'intérêt de cette somme pondérée, permettant de grandement réduire le nombre de paramètres utilisés et de permettre certaines interprétations, par exemple en annulant la contribution d'une modalité dont la représentation n'apporte rien à cet étage i. La sortie de la nouvelle couche peut alors s'écrire :

$$\mathbf{c_{i+1}} = C_i(\alpha_{c_i}\mathbf{c_i} + \alpha_{y_i}\mathbf{y_i} + \alpha_{x_i}\mathbf{x_i}) \tag{6.3.1}$$

avec α des poids scalaires appris pour chacun des termes de la somme et C_i une couche neuronale similaire à g_i ou f_i.

Concernant l'entrée du réseau central, il n'existe alors pas de représentation centrale $\mathbf{c_{-1}}$ et nous proposons donc simplement d'écrire $\mathbf{c_0} = \alpha_{y_0}\mathbf{y_0} + \alpha_{x_0}\mathbf{x_0}$. Le CentralNet permet d'obtenir une dernière représentation $\mathbf{c_{N+1}} = \alpha_{c_N}\mathbf{c_N} + \alpha_{y_N}\mathbf{y_N} + \alpha_{x_N}\mathbf{x_N}$, qui correspond alors à la décision $\hat{\mathbf{z}}_\mathbf{c}$.

Optimisation L'ensemble des poids des réseaux unimodaux f et g, ceux du CentralNet C ainsi que les paramètres scalaires α sont optimisés par descente de gradient stochastique avec une approche Adam [155]. La fonction de coût totale $\mathcal{L}_{totale}$ est alors définie telle que :

$$\mathcal{L}_{totale} = \lambda_2 \mathcal{L}(\hat{\mathbf{z}}_\mathbf{c}, \mathbf{z}) + \lambda_1 \mathcal{L}(\hat{\mathbf{z}}_\mathbf{x}, \mathbf{z}) + \lambda_1 \mathcal{L}(\hat{\mathbf{z}}_\mathbf{y}, \mathbf{z}) \tag{6.3.2}$$

avec $\mathcal{L}$ une fonction de coût appropriée pour le problème et $\mathbf{z}$ le label (ou la vérité terrain) associé à l'échantillon $(\mathbf{x}, \mathbf{y})$.

Dans un premier temps (que nous considérons comme déjà accompli dans le reste de la section), les modèles f et g ont été pré-entraîné, *i.e.* $\lambda_1 = 1$ et $\lambda_2 = 0$. Puis, pour entraîner le CentralNet, nous considérons que $\lambda_1 = \lambda_2 = 1$.

Ainsi, la fonction de coût totale permet d'optimiser à la fois la fusion des modalités mais également de conserver une bonne performance des réseaux unimodaux. Cet objectif multi-tâches est proche dans l'esprit de Neverova *et al.* [206], qui proposent de pré-entraîner les réseaux unimodaux avant d'autoriser de la fusion entre ceux-ci. De plus, les expérimentations sur MNIST Multimodal de la partie précédente (*cf.* Fusion+Multi-tâche versus Fusion dans la Table 6.2.2) laissent également penser qu'il en découle un réel bénéfice sur la performance.

Implémentation Les poids scalaires α sont initialisés avec les mêmes valeurs et une somme unitaire. Ainsi, avant entraînement, la somme pondérée de l'équation 6.3.1 correspond à une simple moyenne. Durant l'entraînement, les poids scalaires ne sont pas contraints, les représentations sommées ayant été préalablement normalisées par une couche de BN.

Ensuite, CentralNet peut notamment s'insérer sur des architectures unimodales existantes et déjà performantes. Un des problèmes majeurs des approches multimodales à plusieurs niveaux combinant directement toutes les modalités à tous les niveaux résident dans le nombre de paramètres trop importants [115], impliquant des difficultés de convergence et une tendance à mal généraliser (et donc le besoin d'utiliser une régularisation très spécifique). Le fait d'utiliser de simples portes modélisées par des scalaires réduit grandement le nombre de paramètres supplémentaires dédiés à la fusion. Et même si la somme pondérée est effectivement une opération linéaire, les blocs neuronaux C_i la suivant sont des opérateurs non-linéaires permettant donc d'apprendre des représentations multimodales complexes.

Interprétation Enfin, les valeurs finales des différents α peuvent amener à certaines interprétations, en analysant quels niveaux semblent les plus utilisées pour la fusion. Par exemple, obtenir en fin d'entraînement des α_{x_i} et α_{y_i} avec des valeurs proches de 0 dès que $i > 0$ correspond à une stratégie de fusion précoce, tandis que des poids α_{c_i} proche de zéros jusqu'à la dernière couche peut être vu comme une stratégie de fusion tardive.

6.3.2 Validation expérimentale

Nous proposons maintenant de valider cette approche sur différentes bases de données, en la comparant avec d'autres stratégies de fusion multimodales mais aussi en essayant d'interpréter les stratégies adoptées.

Bases de données Nous avons utilisé sept bases de données avec différents types de modalités et attachées à différentes tâches, de manière à pouvoir étudier finement les changements de stratégie de fusion multimodale d'un problème à l'autre.

— **MNIST Multimodal :** est décrite dans la section 6.2. Cela nous permet de pré-valider le modèle et le comparer à d'autres stratégies dans des conditions contrôlées.

— **AV-MNIST** a été créé en combinant une modalité image issue de MNIST Multimodal (avec 50% de l'énergie) et une modalité son, consistant en un spectrogramme bruité. La modalité son correspond à une prononciation des différents chiffres, issue de la base de données *Tidigits* augmentée

en la combinant aléatoirement avec les sons urbains d'*ESC-50* [223]. Le but est toujours de prédire le chiffre associé à la paire audiovisuelle et la métrique utilisée est l'accuracy sur un ensemble de test de 10 000 éléments.

— **Montalbano version 2, RGB+D+audio (Montalbano) [94]** contient 14 000 vidéos de gestes italiens. Les vidéos sont enregistrées avec une Kinect, permettant d'obtenir plusieurs modalités. Nous avons utilisé les mêmes modalités que Neverova *et al.* [206] : audio, capture dynamique du squelette, et description RGB+D de la main gauche et de la main droite. Le but est de prédire le type de geste parmi 21 classes. Nous nous évaluons sur l'ensemble de test avec une métrique d'accuracy (diffère du protocole usuel se basant sur un index de Jaccard mais celui-ci impliquerait d'entraîner un classifieur préalable pour identifier les périodes où des gestes sont effectués, la définition de ce classifieur pouvant d'ailleurs impacter significativement les performances observées).

— **mmIMDb [18]** contient 25 959 films, décrits par leur affiche (image RGB) et leur synopsis (texte), extraits de Internet Movie Database (IMDb) [1]. Le but est de prédire à partir de cette paire image et texte le genre du film, parmi 23 possibilités non exclusives. Ce problème de classification multi-label sera évalué suivant le protocole officiel [18, 149], qui correspond à utiliser un F1-Score sur l'ensemble de test. La métrique utilisée est l'accuracy.

— **AFEW [81]** contient 1 156 vidéos courtes extraites de films et de télé-réalité, contenant au moins une personne exprimant une émotion. Le but est de prédire l'émotion parmi 7 classes (*cf.* Chapitre 3), en se servant à la fois du visage et du son.

— **Audioset, sous-ensemble des animaux (Animaux) [103]** est un sous-ensemble d'Audioset [103] et contient 19 842 vidéos de 10 secondes de différents animaux (nommés "animaux domestiques" dans l'ontologie d'Audioset). Le but est de prédire le type d'animal, mais également certains éléments de contexte (*e.g.* sonnerie de téléphone) parmi 211 classes non exclusives. La métrique utilisée est un F1-Score Micro.

— **NTU-RGB+D [250]** contient 56 880 vidéos RGB+D+Squelettes, capturées à partir de 40 sujets différents avec différents points de vue et 60 classes d'activités différentes (pouvant impliquer une ou deux personnes, *e.g.* manger, boire, tomber, serrer la main, frapper). Nous appliquons le protocole usuel de mesure de l'accuracy à travers les sujets, en utilisant uniquement deux modalités : RGB et Squelette.

Une première validation expérimentale sur MNIST Multimodal Si nous revenons à la base de données et aux expérimentations préliminaires, nous avions observé que la stratégie optimale de fusion semblait varier suivant les valeurs d'énergie par modalité et de recouvrement entre modalités choisies. Ainsi, nous reportons en Figure 6.3.2 les résultats obtenus par CentralNet pour différentes configurations de MNIST Multimodal et les comparons à différentes références, notamment en considérant les méthodes de Fusion et de Fusion+Ensemble avec une profondeur optimale (*i.e.* nous testons toutes les profondeurs et reportons le meilleur résultat). La partie gauche de la Figure 6.3.2 montre que sur cette base de données simple, Centralnet permet d'atteindre des performances similaires voire supérieures à toutes les autres méthodes proposées, en s'adaptant donc à différents niveaux de bruits. La partie droite de la Figure 6.3.2 permet également de valider la méthode, excepté lorsque les modalités ont un recouvrement nul, la méthode de Fusion+Multi-tâche semblant alors apporter un réel gain.

Ces observations constituent une première validation de l'objectif du CentralNet : malgré des changements de configurations importants et donc une modification du problème multimodal à résoudre, nous n'observons pas de chute de performance. De plus, une fusion à plusieurs niveaux semble même plus bénéfique que simplement choisir un niveau de fusion optimal.

Les techniques de fusion présentées ici restent simples et pour valider la qualité du CentralNet, nous proposons de continuer à étudier ces premières intuitions plus en détail en se comparant à des approches à l'état de l'art sur divers problèmes multimodaux.

1. https://www.imdb.com/

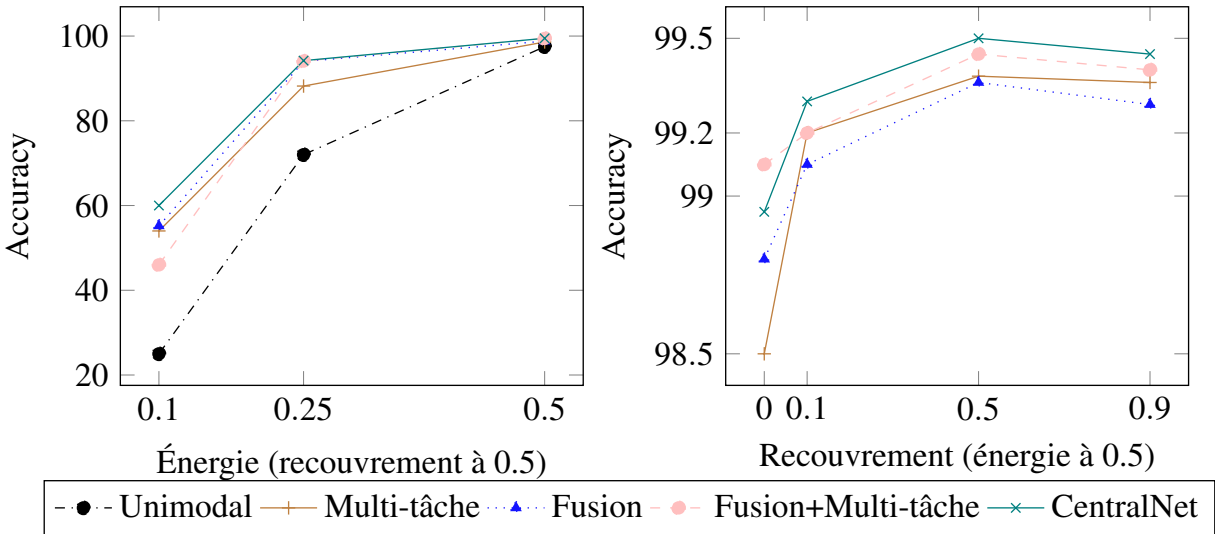


FIGURE 6.3.2 – Évaluation sur MNIST Multimodal. À gauche : représentation de l'accuracy comme une fonction de l'énergie (à recouvrement constant de 0.5) pour différentes méthodes. À droite : représentation de l'accuracy comme une fonction du recouvrement entre modalités (pour une énergie constante à 100%). "Unimodal", le meilleur réseau unimodal, Multi-tâche, la méthode décrite en première partie, Fusion et Fusion+Multi-tâche les méthodes décrites en première partie avec une profondeur optimale et une somme comme opérateur de fusion.

Comparaison avec d'autres méthodes de l'état de l'art Nous nous proposons de nous comparer aux méthodes suivantes afin d'évaluer notre approche. Suivant les problèmes considérés, celles-ci permettent d'atteindre des résultats à l'état de l'art.

— **Meilleure modalité :** choisir la modalité qui obtient les meilleurs résultats sur l'ensemble de validation.

— **Fusion précoce :** fusion au niveau de la première couche.

— **Fusion tardive :** fusion par concaténation des prédictions et classifieur linéaire.

— **ModDrop :** notre implémentation de la méthode proposée par Neverova *et al.* [206], décrite dans le Chapitre 3.

— **GMU :** notre implémentation de la méthode proposée par Arevalo *et al.* [18], proche d'un mécanisme d'attention multimodale.

	AV-MNIST	Montalbano [94]	mmIMDb [18]	AFEW [81]	Animaux [103]	NTU-RGB+D [250]
	Accuracy	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy
Meilleure moda.	74.52	88.0	60.2	52.5	55.2	85.2
Précoce	72.3	97.8	63.0	40.1	50.1	85.4
Tardive	<u>87.8</u>	97.5	<u>63.5</u>	<u>55.3</u>	58.3	<u>88.6</u>
ModDrop [206]	<u>87.8</u>	<u>98.2</u>	62.4	53.2	57.9	88.4
GMU [18]	83.4	98.0	63.0	55.1	<u>58.7</u>	85.8
CentralNet	**87.9**	**98.3**	**63.9**	**57.2**	**59.1**	**89.4**

TABLE 6.3.1 – Résultat obtenus par les différentes méthodes sur toutes les bases de données. Les résultats sont des moyennes sur 10 entraînements. Les meilleurs résultats sont en gras, les deuxièmes meilleurs sont soulignés.

Le détail des architectures unimodales choisies ainsi que le nombre de couches du CentralNet pour chaque base de données sont reportés en Annexe A, de manière à faciliter la lecture. La Table 6.3.1

nous permet d'observer que les méthodes de fusion choisies dépendent bien de la base de données choisie. Par exemple, la GMU donne de bons résultats sur mmIMDb, AFEW et Animaux, mais atteint des performances beaucoup plus basses sur les autres problèmes. De même, la fusion tardive semble permettre d'obtenir souvent de meilleurs résultats que sa variante précoce, comme déjà observé dans la littérature [255].

Nous pouvons également noter qu'une technique de fusion multimodale peut ne pas toujours être bénéfique, puisque lorsque la stratégie semble mal adaptée au problème, la performance obtenue peut être inférieure à celle de la meilleure modalité seule (*e.g.* fusion précoce sur Animaux).

Enfin, CentralNet permet d'obtenir des résultats à l'état de l'art sur l'ensemble des problèmes et en utilisant les mêmes modalités que les autres méthodes. Cela vient confirmer nos premières observations sur MNIST Multimodal. Néanmoins, la justification de cet écart de performance reste très empirique et il serait intéressant d'isoler quels facteurs permettent au CentralNet d'atteindre ces performances.

6.3.3 Analyse

Nous proposons donc plusieurs interprétations, tout d'abord en analysant l'impact de certains facteurs sur la convergence du CentralNet. Puis en essayant de comprendre si les poids scalaires du CentralNet permettent d'identifier différentes stratégies.

Fonction de coût Nous avons choisi, lors de la description du CentralNet, d'ajouter à la fonction de coût basée sur le réseau central de fusion, les fonctions de coût unimodales, avec pour intuition d'ainsi empêcher la dégradation des performances initiales des réseaux unimodaux. Les courbes grises de la Figure 6.3.3 permettent d'observer l'impact de cette approche "multi-tâche". En, effet, la convergence semble plus difficile et plus bruitée dans le cas d'un CentralNet optimisé sans "multi-tâche" que dans le cas contraire. Nous pouvons également noter que cette différence s'amenuise après un nombre important d'itérations.

Un autre élément d'importance que souligne la Figure 6.3.3 réside dans les variations des α au cours de l'entraînement et à travers les différentes couches. Ainsi, dans les deux dernières couches du réseau, le poids central α_c (qui pondère la représentation centrale de la couche précédente) présente une valeur bien plus importante que les poids accordés aux représentations unimodales. En revanche, pour les deux premières couches, seuls les poids accordés au texte sont très faibles, tandis que les valeurs des poids accordés aux représentations visuelle et centrale sont du même ordre en fin de convergence. Une explication plausible est que, bien que le texte seul apporte des performances excellentes sur cette base de données, sa représentation est très compacte et une fois qu'elle a été absorbée par la couche centrale après la première somme pondérée, extraire des informations supplémentaires visuelles, tâche plus difficile mais complémentaire, devient alors plus pertinent. Enfin, nous pouvons observer une tendance générale à la stabilisation des α en fin de convergence du réseau.

Stratégie et valeurs des α La stabilisation des α en fin d'entraînement permet également d'aboutir à des interprétations au sujet de la stratégie de fusion optimale à adopter pour un problème donné.

Ainsi, à partir de la Figure 6.3.4, nous pouvons dans certains cas identifier des stratégies, *i.e.* des architectures de fusion particulières.

En effet, si nous reprenons l'exemple de mmIMDb, nous retrouvons les valeurs des poids visuel, textuel et central pour chacune des couches. Ainsi, dans les premières couches, les réseaux unimodaux apportent une contribution importante, tandis que dans les couches 2 et 3, le réseau utilise principalement la représentation centrale. Ainsi, cela peut être vu comme une stratégie de fusion ni précoce ni tardive (principalement au niveau de la couche 2).

Pour les bases AV-MNIST, AFEW, Animaux, et NTU-RGB+D, il est difficile d'identifier une stratégie de fusion tranchée, mais nous pouvons tout de même retrouver la même tendance à donner de plus en plus de poids à la représentation centrale en allant vers les couches supérieures. De plus, dans le cas d'Animaux, la représentation centrale ne possède aucun poids dans les premières couches, ce qui peut

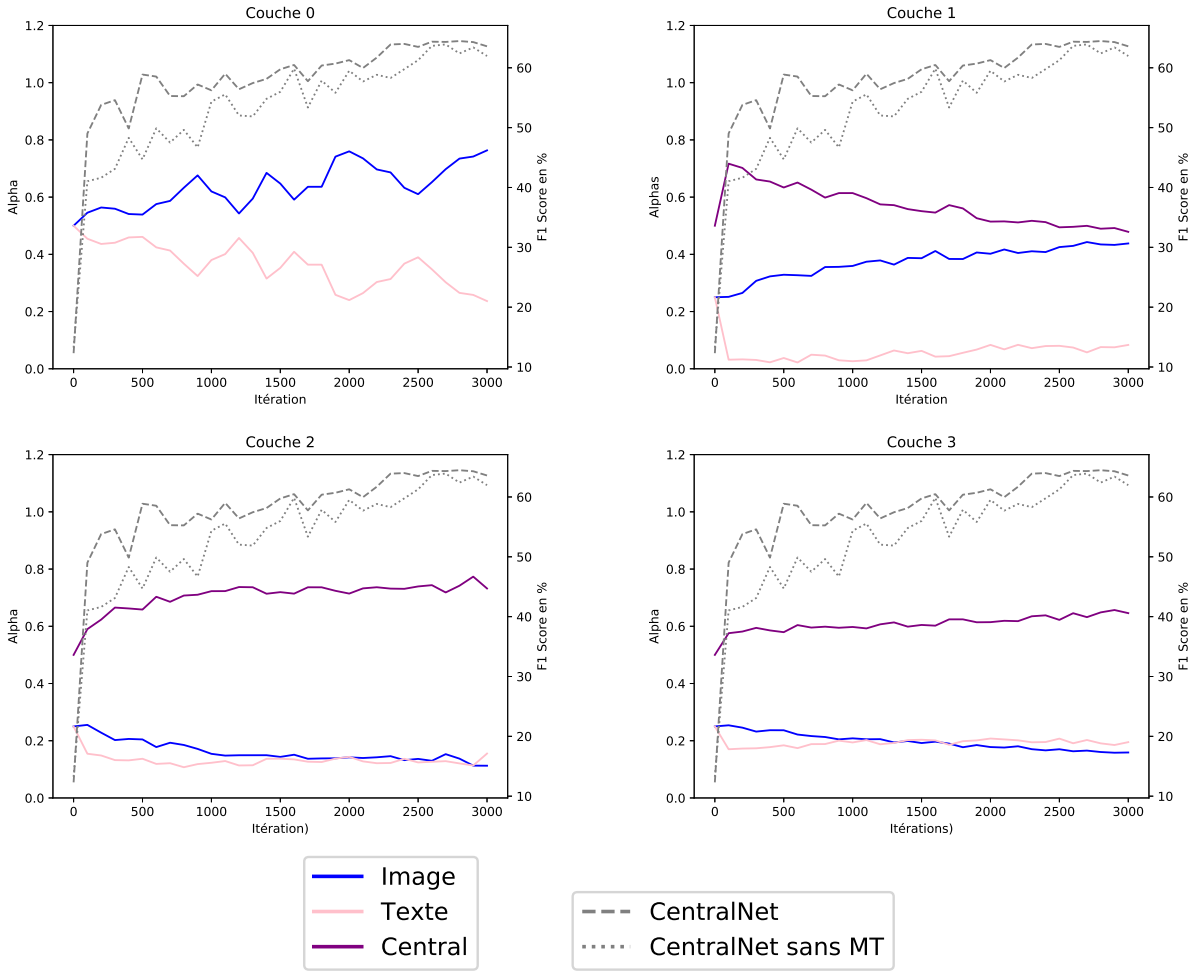


FIGURE 6.3.3 – Illustration de l'évolution des proportions des α (Image, Texte et Central, renormalisés pour une somme unitaire) lors de la convergence sur mmIMDb pour les 4 couches du CentralNet. Notons également les deux courbes des F1 Scores obtenus sur l'ensemble de test respectivement pour un entraînement de CentralNet et un entraînement de CentralNet privé de la fonction de coût "multi-tâche".

s'expliquer car les entrées sont un spectrogramme et une image et leur fusion très précoce n'a que peu de sens.

Enfin, dans le cas particulier de Montalbano, qui comporte 4 modalités, le poids très important donné à la représentation centrale dans la couche 1 correspond à une fusion très précoce, même si les représentations unimodales de plus haut niveau sont ensuite ré-utilisées et que le dernier poids central correspond à une valeur nulle et donc à une fusion relativement tardive.

Ainsi, les α adoptent des stratégies de fusion souvent progressives et ressemblant à celles trouvées manuellement dans la littérature [255]. Néanmoins, il y a un gain réel à fusionner les représentations à plusieurs niveaux de profondeur, comme nous l'avons vu en Table 6.3.1.

Discussion Le CentralNet permet donc d'obtenir des performances à l'état de l'art sur des problèmes de fusion différents et adopte des stratégies qui sont en partie interprétables. Notre approche présente tout de même deux limitations importantes :

— nous sommons les représentations et devons donc adapter au préalable leurs dimensions, ce qui peut amener à la création d'une représentation centrale non pertinente.

— nous supposons qu'il existe N couches dans les deux réseaux unimodaux et que les représentations $\mathbf{x_i}$ et $\mathbf{y_i}$ correspondent à des niveaux compatibles. Ce n'est pas toujours le cas et il pourrait par exemple être plus pertinent de fusionner $\mathbf{x_i}$ avec $\mathbf{y_{i+1}}$.

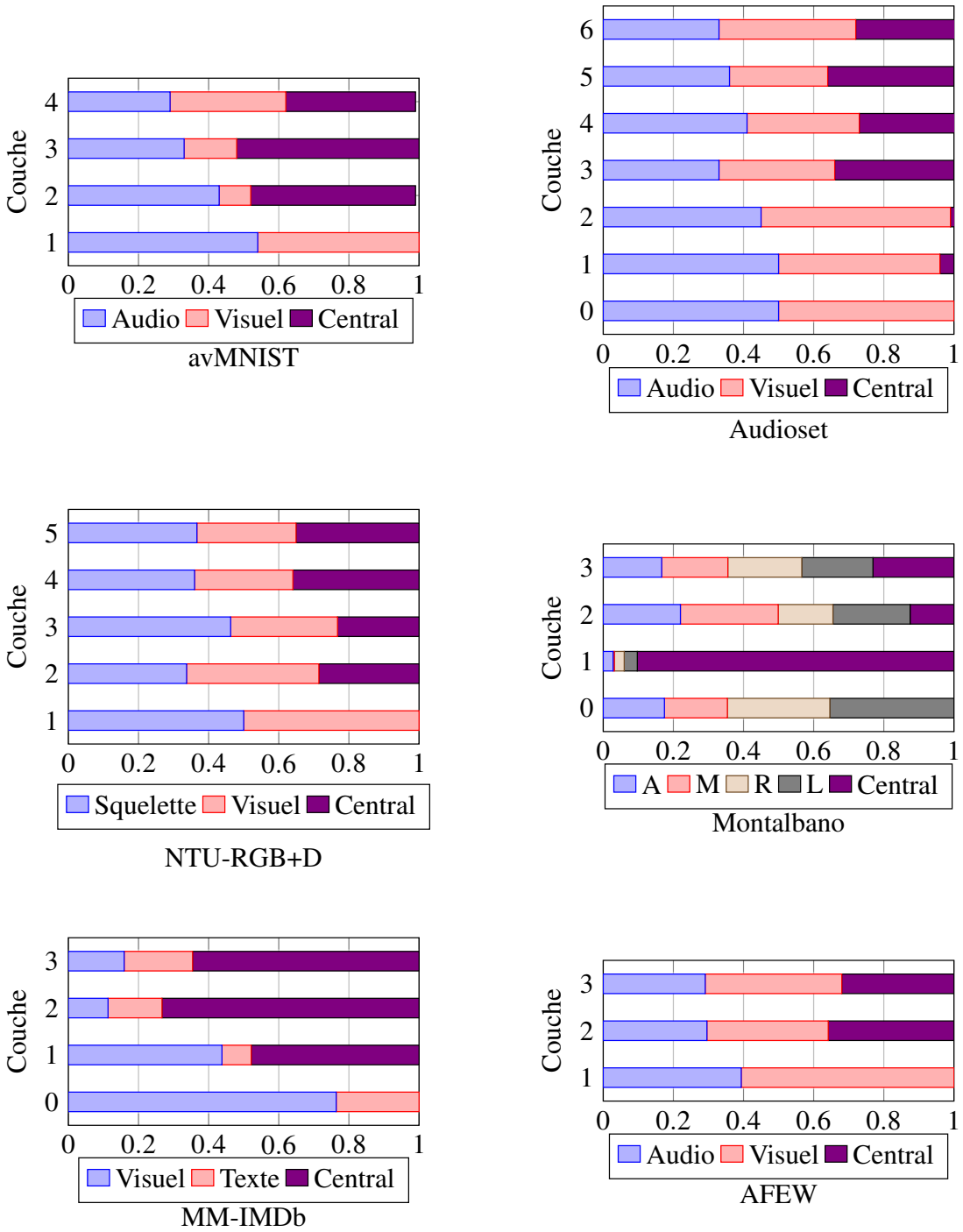


FIGURE 6.3.4 – Valeurs finales des α_i pour les différentes couches i, sur 6 problèmes différents.

6.4 Recherche automatique d'architectures de fusion multimodale

En introduction de la section 6.3, nous avons présenté l'analogie entre CentralNet et une approche résiduelle. Il se trouve que les techniques de type ResNet [122] ont par exemple été étendues en DenseNet [133], consistant à connecter à chaque couche donnée n toutes les couches $i < n$ précédentes. Une telle approche, transposée au domaine multimodal, permettrait alors de résoudre le problème posé par CentralNet, qui ne garantit pas de fusionner des représentations unimodales compatibles. Mais cela impliquerait d'adapter les tailles de représentations utilisées et donc de construire un grand nombre de connexions supplémentaires, ce qui serait particulièrement lourd.

Pour pallier le problème posé par la complexité d'approches telles que le DenseNet [133], il est pertinent d'explorer l'espace des architectures possibles et de trouver par exemple une sous-configuration de ce DenseNet qui soit tout aussi performante et bien moins lourde à utiliser. Il est alors même possible de rechercher des architectures plus variées, avec des activations différentes et successions de blocs plus élaborées. Il existe une grande variété de méthodes pour rechercher des architectures optimales pour un problème donné [92], présentant toutes un point commun : il est nécessaire de définir un espace de recherche des architectures constructibles, *i.e.* il faut pouvoir définir une architecture comme une configuration d'éléments de base.

Nous proposons dans cette partie de construire un tel espace de recherche pour une architecture de fusion multimodale, qui consiste finalement à une reformulation plus générale des architectures exprimables par CentralNet. La recherche d'architectures en elle-même est alors effectuée en appliquant l'approche Recherche Progressive et Efficace d'Architectures Neuronales ou *Efficient Progressive Neural Architecture Search* (EPNAS) de Perez *et al.* [219] et l'efficacité des architectures obtenues est validée sur différentes bases de données.

6.4.1 Reformulation du problème

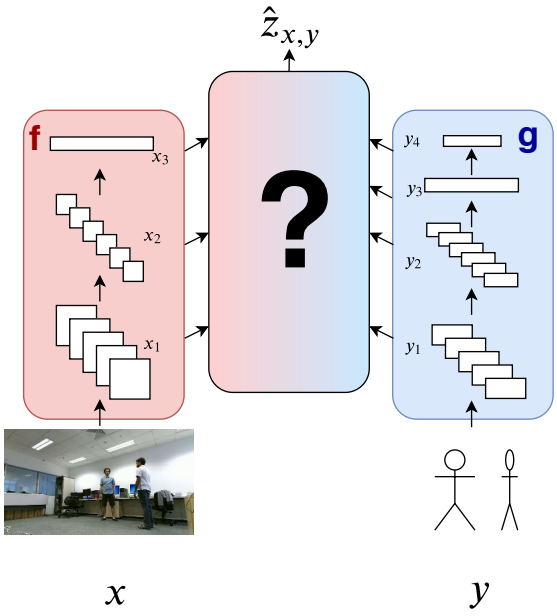


FIGURE 6.4.1 – Formulation du problème comme un problème de combinaisons

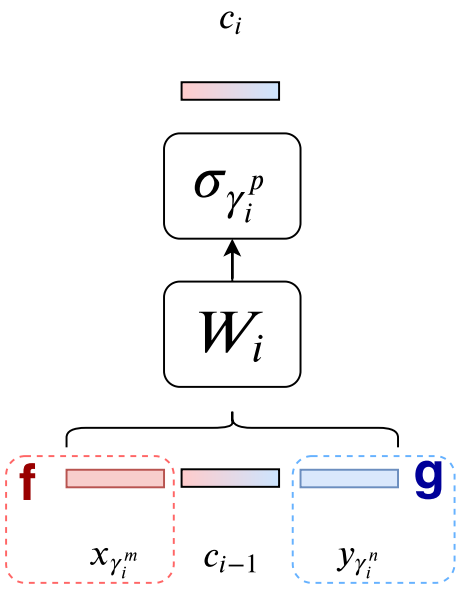


FIGURE 6.4.2 – L'opérateur de fusion paramétrable.

Reformulation Reprenons les réseaux unimodaux f et g et considérons maintenant qu'ils possèdent respectivement un nombre de couches N et M. Il existe donc pour les modalités $\mathbf{x}$ et $\mathbf{y}$ respectivement N et M représentations unimodales. Notre but est finalement de savoir comment tirer au mieux parti de

ces représentations pour construire la meilleure décision $\mathbf{z}_{\mathbf{x},\mathbf{y}}$ possible. Une illustration plus visuelle en Figure 6.4.1 nous ramène à rechercher un modèle central permettant de combiner les représentations de manière optimale. Pour effectuer cette recherche, il faut tout d'abord revoir la formulation du problème et les opérations qu'il est possible d'effectuer, *i.e.* définir un espace de recherche approprié.

Espace de recherche Pour définir l'espace de recherche, nous proposons de revenir à une opération de fusion de base, inspirée de ce qui a été fait pour le CentralNet, mais en la rendant plus générale. L'opération de fusion possède trois entrées : $\mathbf{x}_{\gamma^m}$, $\mathbf{y}_{\gamma^n}$ des représentations unimodales respectivement sélectionnées parmi les $(\mathbf{x_i})_{1...N}$ et les $(\mathbf{y_i})_{1...M}$ et c, la représentation centrale issue de l'opération de fusion précédente. Ces trois entrées sont alors concaténées (permettant d'éviter le problème rencontré par CentralNet des représentations de tailles différentes), puis traitées par un perceptron W, suivie d'une non-linéarité σ_{γ^p} choisie parmi P fonctions d'activations. La somme pondérée utilisée par le CentralNet n'est alors qu'un sous-cas particulièrement simple, où $\mathbf{W}$ possède de nombreuses valeurs identiques.

Il est finalement possible de construire un réseau à partir d'une succession de ces opérations, qui peuvent être plus formellement écrites, pour une couche i :

$$\mathbf{c}_l = \sigma_{\gamma_i^p}\left(\mathbf{W}_i \begin{bmatrix} \mathbf{x}_{\gamma_i^m} \\ \mathbf{y}_{\gamma_i^n} \\ \mathbf{c}_{i-1} \end{bmatrix}\right)$$

Par la suite, nous noterons une configuration d'architectures de fusion comme une séquence de triplets $(\gamma_i^m, \gamma_i^n, \gamma_i^p)$, avec i l'indice de la représentation centrale (*i.e.* le numéro du triplet), γ_i^m l'indice de la représentation venant de $\mathbf{x}$, γ_i^n de $\mathbf{y}$ et γ_i^p la non-linéarité utilisée.

Il est intéressant de noter que l'espace des architectures que nous pouvons construire contient notamment des architectures similaires au CentralNet. Le nombre total de configurations possibles dépend alors du nombre N de représentations issues de $\mathbf{x}$, du nombre M de représentations issues de $\mathbf{y}$, du nombre d'activations possibles P et du nombre L d'opérations de fusion (ou couche du réseau central). Il s'écrit simplement $(N \times M \times P)^L$ et, pour $N = M = 16$, $P = 2$, $L = 5$ avoisine les $3.51 * 10^{13}$ configurations. Ce nombre énorme a pour conséquence de proscrire toute approche directe consistant à seulement explorer toutes les configurations possibles.

6.4.2 Recherche d'architecture

Recherche progressive C'est pourquoi nous proposons d'adapter un algorithme de recherche progressive d'architectures [178, 220], de manière à pouvoir réduire l'exploration (et donc le temps de calcul) nécessaire. Cette approche consiste à explorer l'espace de recherche progressivement, en le divisant en différents niveaux de complexité. Elle conduit à d'aussi bons résultats finaux que des approches plus directes [330, 331] (et donc beaucoup plus coûteuses). Et cela s'explique par le fait que le problème de la recherche d'architectures est facile à diviser en des étapes de complexité croissante (les différents blocs du réseau peuvent être vus comme des "micro-espaces" de recherche [178, 220, 331]). Ainsi, l'idée est d'échantillonner les architectures de manière séquentielle, au sens de la complexité, *i.e.* de la plus simple vers la plus complexe. De la même manière, dans notre cas, nous commençons notre recherche en échantillonnant uniquement un seul opérateur de fusion, puis en construisant des modèles plus complexes en ajoutant de plus en plus d'opérateurs, jusqu'à atteindre une profondeur L du réseau central. Il est important de noter que l'espace que nous avons construit précédemment est suffisamment contraint et présente de bonnes propriétés pour ce type de recherche.

Algorithme L'idée principale de l'algorithme est d'apprendre un modèle permettant de prédire l'accuracy d'un échantillon d'architecture donné. Ce modèle, dit "fonction auxiliaire", est entraîné au fur et à mesure de la recherche progressive de l'espace et permet finalement de réduire le nombre d'architectures de fusion à entraîner et évaluer en prédisant la performance de configurations non observées.

Algorithm 1 Algorithme MFAS

1: **procedure** MFAS
2: π : fonction auxiliaire
3: $\mathbf{f}, \mathbf{g}$: réseaux unimodaux
4: L : nombre maximal de couches de fusion
5: E_{search} : nombre d'itérations de recherche
6: E_{train} : nombre d'epochs d'entraînement
7: K : nombre d'échantillons d'architectures de fusion
8: S_{train}, S_{val} : ensemble d'entraînement et de validation
9: T_{max}, T_{min} : valeur maximale et minimale de la température
10:
11: $T \leftarrow \mathrm{T_{max}}$ *// Fixer la température*
12: $\mathcal{B}, \mathcal{A} \leftarrow \{\}$ *// Ensemble des architectures et de leur performance associée.*
13: **for** $e = 1 \cdots \mathrm{E_{search}}$ **do**
14: $\mathcal{F}_1 \leftarrow \Gamma_1$ *// Ensembles des architectures de fusion pour* $(L = 1)$
15: $\mathcal{M}_1 \leftarrow \texttt{buildFusionNets}(\mathcal{S}_1, \mathbf{f}, \mathbf{g})$
16: $\mathcal{C}_1 \leftarrow \texttt{train}(\mathcal{M}_1, \mathrm{S_{train}}, E_{train})$
17: $\mathcal{A}_1 \leftarrow \texttt{evaluate}(\mathcal{C}_1, \mathrm{S_{val}})$
18: $\mathcal{B}, \mathcal{A} \leftarrow \mathcal{B} \cup \mathcal{S}_1, \mathcal{A} \cup \mathcal{A}_1$
19: $\pi \leftarrow \texttt{updateSurrog}(\mathcal{S}_1, \mathcal{A}_1)$
20: **for** $l = 2 \cdots L$ **do**
21: $\mathcal{F}'_l \leftarrow \texttt{addLayer}(\mathcal{F}_{l-1}, \Gamma_l)$
22: $\hat{\mathcal{A}}'_l \leftarrow \texttt{predAccuracies}(\mathcal{F}'_l, \pi)$
23: $\mathcal{P}_l \leftarrow \texttt{computeProbs}(\hat{\mathcal{A}}'_l, T)$
24: $\mathcal{F}_l \leftarrow \texttt{sampleKArchi}(\mathcal{F}'_l, \mathcal{P}_l, K)$
25: $\mathcal{M}_l \leftarrow \texttt{buildFusionNets}(\mathcal{S}_l, \mathbf{f}, \mathbf{g})$
26: $\mathcal{C}_l \leftarrow \texttt{train}(\mathcal{M}_l, \mathrm{S_{train}}, E_{train})$
27: $\mathcal{A}_l \leftarrow \texttt{evaluate}(\mathcal{C}_l, \mathrm{S_{val}})$
28: $\mathcal{B}, \mathcal{A} \leftarrow \mathcal{B} \cup \mathcal{F}_l, \mathcal{A} \cup \mathcal{A}_l$
29: $\pi \leftarrow \texttt{updateSurrog}(\mathcal{S}_l, \mathcal{A}_l)$
30: $T \leftarrow \texttt{updateTemp}(T, T_{max}, T_{min})$
31: **end for**
32: **end for**
return $\texttt{topK}(\mathcal{B}, \mathcal{A}, K)$
33: *// Return best K from all sampled archs.*
34: **end procedure**

Nous proposons d'utiliser comme "fonction auxiliaire" un réseau récurrent, puisque les échantillons d'architectures de fusion ont des longueurs variables ($[\gamma_i]_{i \in \{1,\cdots,L\}}$). Par la suite, nous l'appellerons π. Les paramètres de π sont mis à jour à l'itération i par descente de gradient stochastique sur un sous-ensemble de Γ_i avec des valeurs observées d'accuracy $\mathcal{A}_i$.

Notre procédure, nommée MFAS, est décrite dans l'algorithme 1. Celui-ci se base en partie sur les approches progressives précédentes [178]. Ainsi, l'algorithme commence avec un modèle de fusion le plus simple possible, *i.e.* à $L = 1$. Puis, des niveaux de complexité supplémentaires sont ajoutés les uns après les autres en échantillonnant K architectures avec une probabilité qui est une fonction des prédictions de π (lignes 23 and 24).

Nous avons également implémenté des itérations de recherche ($\mathrm{E_{search}}$) et un échantillonnage basé sur une température ($\mathrm{T_{max}}, \mathrm{T_{min}}$) de la même manière que dans l'approche EPNAS [220]. Cela permet de faire en sorte que la fonction auxiliaire ne guide pas la recherche à partir de prédictions biaisées à cause d'observations trop partielles, notamment lors des premières itérations. L'usage de la température permet en effet d'avoir un degré de confiance progressif en la fonction auxiliaire : en début d'entraînement,

Méthodes	AV-MNIST	mmIMDb [18]	NTU-RGB+D [250]
CentralNet	87.86 %	0.6223	89.36 %
MFAS	**88.38** %	**0.6250**	**90.04** %

TABLE 6.4.1 – Comparaison des performances obtenues par nos deux méthodes sur trois bases de données multimodales.

Method	Modalités	Accuracy sur le validation
5 meilleures architectures trouvées aléatoirement		
$[(3,3,2),(5,3,2)]$	image + spect.	0.9174
$[(1,1,2),(4,3,1),(5,2,1)]$	image + spect.	0.9190
$[(5,3,1),(4,1,2)]$	image + spect.	0.9196
$[(5,2,1),(5,3,1)]$	image + spect.	0.9224
$[(5,3,1)]$	image + spect.	0.9222
Moyenne (écart-type)	0.9203 (0.0021)	
5 meilleures architectures trouvées par MFAS		
$[(3,3,2),(5,2,1),(1,3,1),(1,1,2)]$	image + spect.	0.9258
$[(5,2,1),(5,2,2),(5,1,1)]$	image + spect.	0.9260
$[(5,3,1),(4,2,1),(5,3,1)]$	image + spect.	**0.9270**
$[(5,3,1),(4,2,1),(3,3,2)]$	image + spect.	0.9266
$[(4,3,1),(5,3,1),(4,3,1),(5,3,1)]$	image + spect.	0.9268
Moyenne (écart-type)	0.9264 (0.0004)	

TABLE 6.4.2 – Évaluation de notre méthode de recherche (en bas) et d'une méthode par exploration aléatoire (en haut) sur AV-MNIST. Les configurations sont décrites par des séquences de triplets dans l'espace de recherche avec $M = 3, N = 5, P = 2$.

on favorise une exploration, puis en réduisant la température, on donne de plus en plus de poids aux prédictions de π (ligne 30).

Enfin, pour gagner en temps de calcul, les échantillons d'architectures sont entraînés pendant très peu d'epochs (comme dans ENAS [221]) et les poids appris sont si possible partagés entre les différentes architectures. Traiter des réseaux de fusion multimodale demande énormément de ressources de calcul et ces techniques de réduction sont donc essentielles dans notre cas.

6.4.3 Validation expérimentale

Détails d'implémentation Nous avons fait le choix d'utiliser $P = 3$ activations : sigmoïde, ReLU et LeakyReLU. Les modèles unimodaux sont les mêmes que ceux utilisés pour CentralNet (décrits en Annexe A) et impliquent que N et M changent suivant les bases de données. Enfin le nombre maximal L de couches de fusion est fixé à 4 pour toutes les bases de données.

Bases de données Comme nous le verrons par la suite, les temps de calcul pour MFAS sont conséquents. C'est pourquoi nous avons choisi de limiter notre étude à trois bases de données sur lesquelles nous avions déjà évalué CentralNet (et qui sont donc décrites dans la partie précédente) : AV-MNIST (chiffres en image + son), mmIMDb (genre d'un film à partir de son affiche et de sa synopsis), et NTU-RGB+D (actions à partir d'une vidéo + squelette).

La Table 6.4.1 permet d'attester que la méthode MFAS, tout comme CentralNet, atteint d'excellentes performances sur plusieurs problèmes différents. De plus, elle dépasse systématiquement CentralNet d'une marge statistiquement significative.

Comparaison avec une recherche par exploration uniquement Un premier élément que nous souhaitons vérifier est l'efficacité de la méthode de recherche comparée à un échantillonnage aléatoire. La Table 6.4.2 reporte les résultats obtenus pour une exploration aléatoire (partie haute) et pour MFAS (partie basse). Les deux approches utilisent le même nombre d'échantillons (180 configurations). Nous montrons la configuration et l'accuracy des 5 meilleures architectures trouvées par chacune des méthodes. Dans le cas de la méthode d'exploration aléatoire, nous constatons un écart-type entre les différents scores beaucoup plus important que dans le cas de MFAS. De plus, la méthode MFAS permet d'obtenir

Méthode	Modaltiés	Accuracy (%)
Approches unimodales		
LSTM [250]	squelette	60.69
part-LSTM [250]	squelette	62.93
Spatio-temp. attention [259]	squelette	73.40
Inflated ResNet-50 + Glimpse Cloud [26]	vidéo	86.6
Approches multimodales		
Shahroudy *et al.* [251]	vidéo + squelette	74.86
Shahroudy *et al.* [251]	vidéo + squelette	74.86
Bilinear Learning [127]	vidéo + squelette	83.30
Bilinear Learning [127]	vidéo + squelette + profondeur	85.40
2D/3D Multitask [187]	vidéo + squelette	85.50
Coopération + fusion tardive produit [284]	vidéo+squelette	86.4
Réseaux unimodaux f et g		
f : Inflated ResNet-50 [26]	vidéo	83.91
g : Co-occurrence [169]	squelette	85.24
Nos approches		
Fusion tardive [252]	vidéo + squelette	88.60
CentralNet	vidéo + squelette	89.36
MFAS	vidéo + squelette	90.04

TABLE 6.4.3 – Résultats des approches MFAS et CentralNet et d'approches à l'état de l'art (fin 2018) sur NTU-RGB+D.

en moyenne des résultats plus compétitifs, soulignant donc l'intérêt de la recherche progressive d'une configuration optimale.

Comparaison avec l'état de l'art (fin 2018) sur NTU-RGB+D Pour le cas de la base de données NTU-RGB+D qui est très utilisée par la communauté, nous avons également étudier la position de nos deux solutions comparée à l'état de l'art.

Ainsi la Table 6.4.3 souligne tout d'abord la compétitivité des deux réseaux unimodaux f et g sur lesquels nous nous sommes basés. Ensuite, nous pouvons noter que les méthodes multiniveaux que nous proposons permettent d'atteindre des performances proche ou même dépassant l'état de l'art.

6.4.4 Architectures trouvées

La Figure 6.4.3 illustre l'architecture de fusion trouvée par MFAS pour deux des bases de données.

Dans le même esprit que pour CentralNet, nous pouvons retrouver des différences importantes entre les architectures obtenues. En revanche, il est beaucoup plus difficile de trouver une interprétation aux architectures échantillonnées, bien qu'il soit possible de faire quelques liens entre par exemple la complexité du modèle et celle du problème à résoudre. En effet, NTU-RGB+D est un problème beaucoup plus complexe et avec beaucoup plus de dimensionnalité que mmIMDb, ce qui implique alors d'exhiber des relations beaucoup plus complexes entre les représentations unimodales.

6.4.5 Convergence

La Figure 6.4.4 montre le comportement de notre procédure de recherche en traçant l'erreur sur l'ensemble de validation de NTU-RGB+D obtenue par les architectures échantillonnées.

Nous pouvons observer que les architectures échantillonnées présentent des performances de plus en plus stables au fur et à mesure que la recherche est effectuée. Cette stabilisation s'explique par deux éléments. Les poids partagés entre les différentes itérations sont de plus en plus efficaces au cours de la recherche. Et le fait d'utiliser une température décroissante permet de donner de plus en plus de confiance à la fonction auxiliaire, qui sélectionne alors des modèles de plus en plus pertinents et semblables. Enfin, nous pouvons noter que, lors des dernières itérations de l'algorithme, l'erreur moyenne (étoiles) sur l'ensemble de validation est beaucoup plus faible qu'au départ.

Un autre effet propre à notre espace de recherche est le fait que même lors des premières itérations de l'algorithme, il est possible d'aboutir à certaines configurations fournissant de bonnes performances sur l'ensemble de validation. Comme les architectures échantillonnées ne sont entraînées que pour quelques

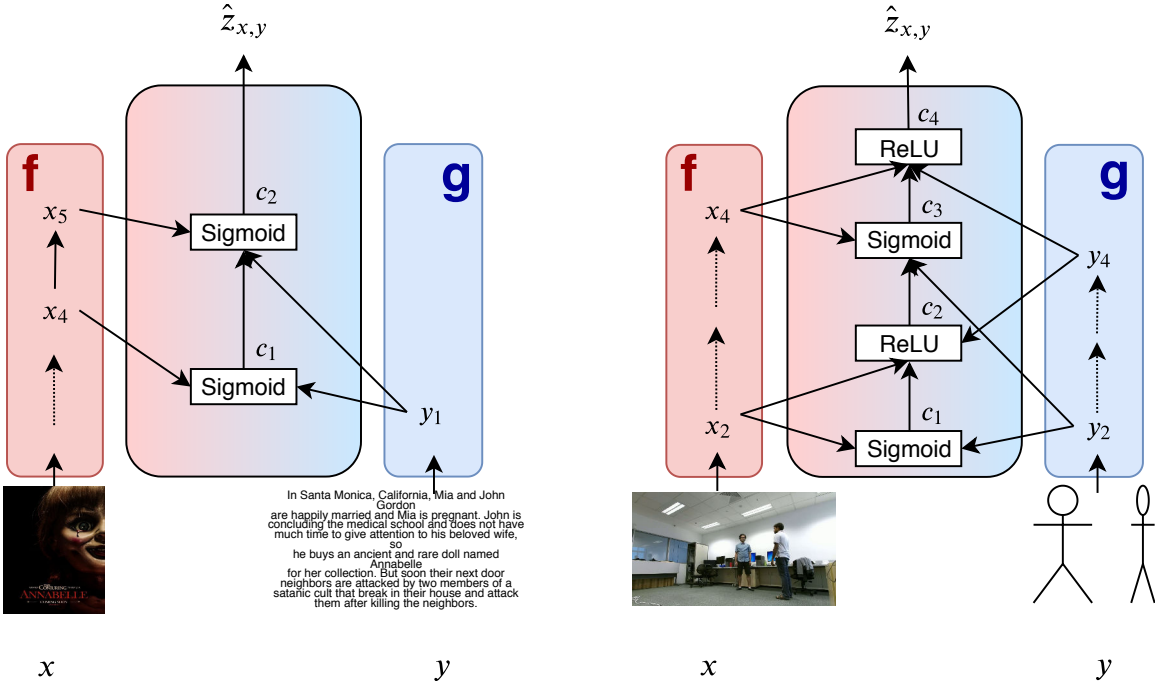


FIGURE 6.4.3 – Architectures de fusion obtenues pour mmIMDb (gauche) et NTU-RGB+D (droite).

epochs, cet effet n'est pas nécessairement un indicateur clair de la qualité d'une architecture échantillonnée. En effet, il est possible d'échantillonner une opération de fusion très simple mais de l'appliquer à des représentations unimodales très profondes et d'ainsi obtenir de meilleures performances que d'autres architectures plus complexes, qui donneront pourtant dans les prochaines itérations de meilleurs résultats du fait de la mise à jour de leur poids.

C'est pour cette raison que notre approche d'échantillonnage par température offre la possibilité de s'échapper de ces fausses configurations optimales, qui peuvent être vues comme de faux optima locaux. Ainsi, cela souligne encore une fois l'importance de reposer sur la fonction auxiliaire qu'une fois l'exploration bien avancée, afin d'éviter de rester bloqué sur les premiers résultats biaisés. Nous avons choisi de manière empirique (par comparaison avec une décroissance linéaire) d'utiliser une décroissance exponentielle pour la température d'échantillonnage (illustrée dans la partie gauche de la Figure 6.4.4).

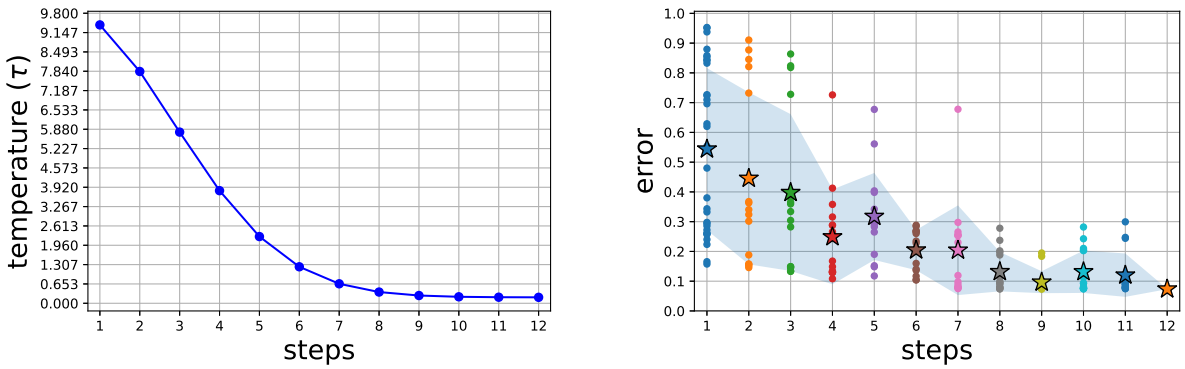


FIGURE 6.4.4 – Descente de température (gauche) et son effet sur les écarts d'erreur dans les modèles échantillonés (droite). Un point correspond à l'erreur obtenue par une architecture échantillonnée, une étoile correspond à la moyenne des erreurs des architectures échantillonnées et l'aire bleue permet de visualiser l'intervalle de confiance.

6.4.6 Temps de calcul

La Table 6.4.4 résume les temps de calcul et le matériel utilisé lors des différentes expériences. Une approche mutli-GPU a été nécessaire pour effectuer des entraînements sur NTU-RGB+D en conservant des tailles de batch ne perturbant pas la convergence. Les temps de calcul plus importants sur NTU-RGB+D s'expliquent donc notamment par la dimensionnalité importante des entrées, qui implique alors des modèles plus complexes et donc un espace de recherche plus large.

Bien que ces temps de calcul soient importants, il faut garder à l'esprit qu'ils sont effectués qu'au moment de l'entraînement, donc une seule fois. Trouver manuellement une méthode permettant d'atteindre ce type de performance impliquerait un coût bien plus important qu'une semaine de calcul.

Enfin, bien que les résultats obtenus soient particulièrement bons, les architectures trouvées présentent le défaut d'être beaucoup moins interprétables que celles proposées par CentralNet. Les deux méthodes peuvent donc être pertinentes, suivant que le besoin tende plus vers un besoin d'interprétation de la stratégie ou vers un besoin de performance maximale.

Base de données	GPUs (P100)	$E_{search} \times L$ (itération)	Temps de recherche (heures)	Temps moyen par itération (heures)
AV-MNIST	1	$3 * 4 = 12$	3.42	0.285
mmIMDb	1	$5 * 3 = 15$	9.24	0.616
NTU-RGB+D	4	$3 * 4 = 12$	150.91	12.57

TABLE 6.4.4 – Temps de calcul

6.5 Conclusions

6.5.1 En résumé

Ce chapitre a permis d'apporter trois contributions au domaine de la fusion multimodale. Tout d'abord, nous avons proposé une méthode pour simuler une base de données multimodale en jouant sur deux facteurs : le bruit propre à chaque modalité et le recouvrement de l'information entre les modalités. Cette base de données nous a alors permis d'évaluer différentes stratégies de fusion et de confirmer que la stratégie optimale dépend du problème à traiter.

Cela nous a naturellement amenés à nous concentrer sur une méthodologie pour adopter automatiquement la meilleure stratégie pour un problème donné. Nous avons donc construit dans un premier temps CentralNet, composé d'un modèle de fusion multiniveau, centralisant les représentations des réseaux unimodaux. Pour permettre de choisir quelles représentations utiliser, nous avons utilisé des "portes" modélisées par des scalaires, appris en même temps que l'ensemble du réseau. Cette approche a obtenu des résultats à l'état de l'art sur sept bases de données multimodales différentes et a également permis d'interpréter différentes tendances en termes de stratégie.

Enfin, nous avons étendu le principe de CentralNet en proposant de formuler la construction de ce réseau central de fusion multimodale comme un problème de recherche d'architectures. Pour cela, il nous a fallu définir un espace de recherche des architectures multimodales qui soit pertinent et qui permette d'appliquer des techniques de recherche progressives, *i.e.* qui soit décomposables en blocs très simples. Nous avons donc arbitrairement formaliser l'opération de fusion, de manière à définir une architecture multimodale comme une composition de ce type d'opérations. Les résultats obtenus dépassent également ceux de CentralNet et sont donc à l'état de l'art sur les trois bases ayant servi pour l'évaluation. Bien que très performante, cette dernière méthode présente le désavantage d'être plus coûteuse en temps de calcul (pour l'entraînement) et moins interprétable que CentralNet. Il y a donc également un compromis à trouver entre ces deux méthodes, chacune pouvant être utile suivant les besoins d'un utilisateur.

6.5.2 Perspectives

Extension de l'espace de recherche Dans MFAS, nous avons volontairement énormément simplifié le bloc de fusion à la base de toute architecture. Les architectures explorées sont donc tout de même limitées à certains comportements spécifiques. Or certaines méthodes particulièrement efficaces en fusion multimodale impliquent d'utiliser des techniques telles que des mécanismes d'attention multimodale [125, 182], consistant à repondérer les représentations unimodales en fonction d'un contexte multimodal. Il serait donc intéressant d'autoriser ce type de comportements, ainsi que d'autres opérateurs, tels que des produits bilinéaires.

Tirer le meilleur parti des deux approches CentralNet et MFAS présentent tous deux des avantages et des inconvénients. En s'inspirant de méthodes de recherches d'architectures récentes [179], il serait possible de trouver un meilleur compromis entre performance et interprétation, en effectuant une recherche d'architectures multimodales mais en se basant sur des "portes" (*i.e.* des poids sur chaque branche du réseau). De plus, il serait alors possible d'initialiser la recherche d'architecture à partir d'une première configuration trouvée par CentralNet.

Problème multimodal et temporel Les problèmes multimodaux, dont certains que nous avons abordés dans ce Chapitre, sont parfois également temporels. Pour NTU-RGB+D par exemple, nous avons évacué ce problème en considérant les représentations moyennes temporelles. Mais la notion de d'ordre créé par la temporalité possède dans certains cas un impact important (*e.g.* pour reconnaître une séquence d'actions) et le prendre en compte permettrait sûrement d'atteindre de bien meilleures performances.

Une méthode usuelle pour traiter des entrées séquentielles est le réseau récurrent (évoqué dans le Chapitre 2 et illustrée par la Figure 2.1.5), qui peut tout à fait être appliqué dans le cas d'entrées multimodales. De nombreux aspects entrent alors en jeu, comme le niveau de la fusion temporelle comparé à celui de la fusion multimodale, certains préconisant une fusion multimodale avant la fusion temporelle [308], d'autres après [196]. Et finalement nous retombons sur le même problème de recherche d'architectures que celui que nous venons d'aborder pour la fusion multimodale. Il serait donc pertinent d'adapter nos deux méthodes à ce type de problématique, en intégrant par exemple la fusion temporelle comme une opération supplémentaire à prendre en compte.

Chapitre 7

Conclusion générale

Table des matières

Nous proposons de conclure ce manuscrit à travers un résumé des contributions apportées face aux problématiques soulevées en introduction. Des limitations de ces contributions seront ensuite discutées et ouvertes à perspectives en Section 7.2.

7.1 Contributions

Les travaux qui ont été présentés dans ce manuscrit se sont inscrits dans le contexte des réseaux de neurones profonds et de leur application à des données dites multimodales. Lors de cette thèse et comme souligné en Section 1.2 de l'introduction générale, nous avons été confrontés à différentes problématiques inhérentes aux problèmes multimodaux. Ainsi, nous avons cherché à adresser celles-ci à travers diverses contributions, que nous résumons en reprenant les problématiques abordées en Section 1.2 de l'introduction générale.

7.1.1 Fusion entre modalités

Étant données la grande variété de problèmes multimodaux et les nombreuses méthodes de fusion multimodale qui y sont associées, face à un nouveau problème multimodal il est particulièrement difficile de sélectionner une stratégie de fusion efficace. Nous avons donc proposé différentes contributions, notamment pour vérifier que la stratégie optimale de fusion dépend du problème à traiter mais aussi pour être capable d'adapter la stratégie de fusion à chaque problème.

Contributions Nous avons validé cette observation dans les Chapitres 3 et 6, où différentes méthodes de fusion de l'état de l'art ont été évaluées et ont obtenu des résultats très variables suivant les problèmes abordés. Ainsi, nous avons proposé dans le Chapitre 3 de nous limiter à une fusion particulièrement simple et tardive, de manière notamment limiter les effets de la faible taille de l'ensemble d'entraînement. Dans le Chapitre 6, nous avons ensuite proposé deux nouvelles solutions de fusion multimodale (CentralNet et MFAS), permettant de s'adapter au problème multimodal et d'ainsi dépasser l'état de l'art sur une variété de problèmes. CentralNet (*cf.* Section 6.3) consiste en la définition d'un réseau de neurones central qui fusionne les représentations issues de réseaux unimodaux pré-entraînés en apprenant à les combiner via une pondération adéquate. Cette fusion par pondération permet alors de modéliser différentes architectures et d'ainsi trouver une configuration plus adaptée pour le problème donné. MFAS a permis d'étendre cette idée en définissant un plus grand nombre de configurations et en utilisant une méthode de recherche d'architectures progressive pour sélectionner les meilleures architectures pour le problème donné.

Enfin, nous avons abordé dans le Chapitre 5 un problème de transfert de connaissances multi-sources, pouvant être vu comme un problème multimodal très particulier. En effet, les représentations extraites par les différents modèles pré-entraînés peuvent être perçues comme autant de modalités décrivant le signal d'origine (*i.e.* le visage qui a été traité par les modèles). Rassembler ces représentations et en tirer une information sémantique correspond alors à la définition d'un problème multimodal telle que donnée en Chapitre 2, mais dans un cas où le nombre de modalités est important et où l'on possède peu d'exemples annotés. C'est pourquoi nous avons proposé un apprentissage non supervisé sous la forme d'un auto-encodeur, rejoignant des méthodes de fusion multimodales cherchant à réduire la dimensionnalité en exploitant les corrélations entre modalités.

Limitations Les méthodes de recherche d'architectures de fusion proposées correspondent à des temps d'entraînement pouvant devenir très importants (*cf.* MFAS), pour des gains de performances pouvant être perçus comme relativement marginaux. Notons néanmoins que le temps nécessaire à la conception manuelle d'une bonne stratégie de fusion multimodale peut être conséquent et devrait être pris en compte dans ce contexte. Comme évoqué dans les perspectives du Chapitre 6, l'aspect temporel a été également très peu traité par nos approches de fusion multimodale. En effet, lorsque nous étions confrontés à des représentations séquentielles (*e.g.* NTU-RGB+D), nous nous sommes contentés d'utiliser une moyenne

temporelle des représentations extraites. Bien que se révélant efficace, cette opération est destructrice et pourrait amener pour certains problèmes multimodaux une perte importante de performance. Ainsi, temporalité et multimodalité pourrait être explorée conjointement, en ré-utilisant des techniques similaires à celles du Chapitre 6. Il est également important de noter, que bien que les évaluations des méthodes de fusion ont été effectuées sur de nombreux ensembles de données et se sont révélées très performantes, nous nous sommes tout de même restreints à une définition spécifique d'un problème multimodal. En effet, les problèmes que avons présenté dans ce manuscrit supposent une perception simultanée des différentes modalités. Dans le cadre d'une interaction, il est plus probable que la perception des modalités soit séquentielle, une modalité pouvant piloter le traitement d'une autre. Par exemple, le *Visual Question Answering* consiste à répondre à une question (modalité textuelle) à partir d'une modalité visuelle. Appliquer des méthodes tels que CentralNet et MFAS serait alors plus difficile et impliquerait une adaptation importante, en prenant en compte les techniques très différentes utilisées. Nous détaillerons dans les perspectives générales quelques propositions d'adaptation et piste de réflexion pour relier ces différents problèmes multimodaux.

7.1.2 Grande dimensionnalité du signal d'entrée et faible quantité de données

L'utilisation de contenus multimodaux implique souvent des entrées avec une dimensionnalité élevée et des bases de données contenant peu de données annotées (particulièrement dans le cas de l'émotion multimodale). Nous avons donc proposé différentes méthodes pour éviter les effets négatifs dus à cette haute dimensionnalité, tels que la difficulté à converger ou le sur-apprentissage.

Contributions Pour pallier le faible nombre de données et à leur importante dimensionnalité, nous avons exploré des techniques de transfert de connaissances et de réduction de la dimensionnalité, au sens large, dans l'ensemble des chapitres. Avec les Chapitres 3 et 6, nous avons illustré l'intérêt d'utiliser des modèles pré-entraînés pour extraire des représentations pertinentes des différentes modalités et ainsi limiter l'effet de sur-apprentissage dû à la grande dimensionnalité des entrées. Par exemple, le Chapitre 3 relate nos deux participations à une compétition internationale de reconnaissance multimodale de l'émotion. Il a permis de distinguer une variété de modèles pré-entraînés sur d'autres tâches et d'autres bases de données, afin d'extraire de manière robuste des représentations de plus faible taille des modalités visuelles et audio. Le choix des modèles pré-entraînés à utiliser s'est révélé crucial, puisque les transferts de connaissances effectués en 2018, avec des modèles pré-entraînés sur des données plus variées et des tâches plus proches de la reconnaissance d'émotion ont conduit à des performances supérieures à celles de 2017. Cette importance du choix des modèles pré-entraînés s'est également manifestée dans le Chapitre 6, qui a consisté en la fusion multimodale de représentations issues de modèles pré-entraînés et étant déjà particulièrement performants sur une modalité donnée.

Le Chapitre 4 a permis également de s'intéresser à des approches d'apprentissages multi-domaines, en entraînant un modèle avec la même tâche (reconnaissance d'expressions faciales) sur 3 bases de données très différentes (taille, type d'images, qualité des annotations), permettant de montrer qu'un changement de domaine peut avoir beaucoup d'impact sur la performance. De plus, ce chapitre nous a permis d'affiner la notion de représentation de l'expression faciale et d'ainsi proposer l'utilisation de modèles avec des couches cachées de dimensions très réduites, conduisant à une bonne généralisation et une meilleure interprétation des résultats obtenus. Enfin, au lieu de chercher à sélectionner (de manière heuristique ou automatique) le meilleur modèle pour un transfert de connaissances, le Chapitre 5, a été l'occasion de proposer une nouvelle approche de transfert multi-source (*i.e.* dans le cas où nous souhaitons transférer simultanément les connaissances contenues par plusieurs modèles pré-entraînés). Nous avons ainsi proposé une méthode en deux étapes, consistant tout d'abord en à rassembler les représentations extraites par les différents modèles pré-entraînés en une unique représentation de dimension plus réduite. Ensuite, nous avons utilisé une approche de distillation pour ré-entraîner un modèle unique et plus léger à prédire cette représentation unique. Cette approche en deux étapes a permis de transférer la connaissance contenue dans différents modèles d'analyse faciale vers un unique réseau de neurones, donnant d'excellents résultats sur des tâches liées à cette connaissance et permettant donc une représentation

compacte et robuste des visages.

Limitations Nous avons proposé de résoudre le problème de la grande dimensionnalité du signal (à la vue du faible nombre de données d'entraînement) en utilisant des approches de transfert de connaissances. Celles-ci peuvent finalement être vues comme une manière d'augmenter implicitement le nombre de données d'entraînement et d'ainsi aider à la généralisation. En revanche, ces techniques nécessitent que la tâche et le domaine à traiter présentent des liens avec ceux utilisés pour pré-entraîner des modèles. En effet, nous avons eu la chance d'être confronté à un problème d'analyse faciale, qui est un sujet déjà très exploré et où de nombreuses tâches ont été résolues. En revanche, il pourrait exister également des problèmes avec des entrées hautement dimensionnelles et où un transfert de connaissances devient plus difficile, peu de tâches avec un domaine très large se révélant réellement pertinentes pour le transfert. Il serait alors plus pertinent de favoriser d'autres techniques (telles que les approches semi-supervisées [266, 17, 135], de l'augmentation des données [71, 174, 27], *etc*.).

7.1.3 Performances et complexité des modèles

Le fait de traiter des signaux de grandes dimensions et de disposer d'un grand nombre de données implique aussi de disposer d'une bonne capacité d'apprentissage de la part du modèle et donc souvent d'une complexité importante et d'un large nombre de paramètres. Or, cette complexité peut rendre le modèle inapte (notamment en termes de temps de calcul) à être ré-utilisé dans des applications réelles par la suite, voire l'amener à ne pas généraliser au-delà des données vues pendant son entraînement. Il est donc intéressant de chercher à réduire cette complexité, tout en conservant, voire en améliorant les performances.

Contributions C'est pourquoi nous avons abordé cette notion de sélection de modèles et de réduction de la complexité à travers différents chapitres. Dans le Chapitre 3 nous avons repris le principe du rasoir d'Occam, consistant à choisir systématiquement pour deux solutions ayant des performances équivalentes celle qui est la plus simple (*e.g*. au sens du nombre de paramètres, du nombre d'opérations nécessaires, *etc*. .). Cette heuristique visant à réduire drastiquement le nombre de paramètres a conduit à un gain de performance et à une réduction du temps de calcul, permettant d'envisager un usage en temps réel de la reconnaissance automatique de l'émotion.

La représentation cachée de très faible taille et le nombre de paramètres réduits du modèle *ResNet-disc* utilisé dans le Chapitre 4 se sont également avérés concluants, montrant qu'il est possible d'atteindre des performances élevées avec des modèles particulièrement légers. Le processus de distillation utilisé dans le Chapitre 5 a été une autre manière de résoudre ce problème de sélection des modèles, en rassemblant et distillant la connaissance de plusieurs modèles en un seul réseau de neurones avec un nombre de paramètres réduits. Le réseau de neurones obtenu a obtenu des performances du niveau de l'état de l'art sur différentes tâches

Enfin, bien que les approches de fusion multimodale utilisées dans le Chapitre 6 soient relativement lourdes en nombre de paramètres, si nous ne considérons que l'opération de fusion et non les opérations unimodales, la complexité des modèles reste réduite. Particulièrement dans le cas de CentralNet (*cf*. Section 6.3), l'opération de fusion en elle-même ne comporte qu'un seul paramètre par modalité et un paramètre pour la représentation qui précède, tandis que le réseau central conserve un nombre de paramètres inférieurs à celui des réseaux unimodaux déjà entraînés.

Limitations Bien que nous nous soyons intéressés à la complexité de nos modèles et que nous ayons recherché un compromis entre performances et taille de modèle, nous n'avons aucune certitude d'avoir trouvé le compromis optimal, du fait de l'impossibilité de tester l'ensemble des configurations de modèles. Par exemple, les dimensions des représentations utilisées dans les Chapitres 4 et 5 ont été sélectionnées par validation croisée, mais nous sommes loin d'avoir testé toutes les possibilités, puisque ces représentations dépendent également de la profondeur du modèle utilisé pour les extraire et des données

utilisées pour l'entraînement. Il serait donc intéressant de repartir de méthodes de sélection de modèle heuristiques tels que le critère d'Akaike [5]) et de les adapter dans le cadre particulier de ce manuscrit. En effet, nous pourrions par exemple modifier la recherche d'architectures effectuées au Chapitre 6 en ne mesurant non pas une performance classique sur l'ensemble de validation, mais en prenant également en compte la configuration d'architecture choisie (notamment la complexité de celle-ci), structurant ainsi plus facilement l'espace de recherche et poussant la sélection des modèles simples et performants.

7.1.4 Interprétabilité

Le fait d'utiliser un apprentissage neuronal peut souvent conduire à un effet "boîte noire", *i.e.* à une difficulté à interpréter les décisions produites par le modèle et à analyser ses représentations cachées. Nous avons proposé des solutions permettant d'obtenir des informations plus complètes sur la qualité de l'apprentissage qu'un simple indicateur de performance.

Contributions Dans les Chapitres 3 et 4 nous avons étudié différents protocoles d'évaluation des modèles dans le cadre de la classification, en prenant notamment en compte des différences de distributions entre les ensembles d'entraînement et de test.

Nous avons également introduit des mesures des performances plus complètes en réalisant plusieurs entraînements avec des initialisations différentes et en conservant la valeur moyenne et la variance des performances obtenues, ce qui s'est révélé être un indicateur précieux dans le cadre de la comparaison de différentes méthodes (*e.g.* en comparant différentes méthodes de fusion très proches dans le Chapitre 6). En proposant une couche cachée de dimension très réduite dans le Chapitre 4, nous avons pu facilement visualiser et interpréter des directions au sein de l'espace créé par cette représentation cachée. De plus, l'utilisation d'un GAN nous a permis de confirmer des liens avec des directions psychologiques de l'émotion, permettant ainsi de mieux comprendre la nature des représentations apprises par un réseau de neurones lors de la reconnaissance des expressions faciales.

Dans le Chapitre 6, nous avons aussi cherché à interpréter les choix de stratégies obtenues par notre modèle CentralNet, en étudiant les valeurs des poids accordés à chaque modalité. Cela a permis de valider le fait que les stratégies de fusion les plus optimales varient avec les problèmes multimodaux à résoudre. Quant à notre méthode MFAS, bien que les architectures obtenues soient moins interprétables, nous avons pu identifier certaines tendances en termes de choix de stratégies, notamment en termes de complexité des fusions suivant le problème à résoudre.

Limitations Il est important de garder à l'esprit que les interprétations que nous avons effectuées l'ont été dans des contextes précis et en restant tout de même à un niveau élevé d'interprétation. Cette explicabilité limitée est inhérente à de nombreuses approches d'apprentissage profond, même si des pistes d'amélioration existent [317, 211]. Pour pouvoir améliorer les interprétations de ce qui a été appris par nos modèles, il faudrait poursuivre des efforts d'explicabilité, en combinant nos approches à des analyses du processus de décision. Par exemple, il est possible de construire des CNN [316] dont le processus de décision est plus interprétable et il serait donc intéressant de distiller nos approches dans ce type de modèles explicables, comme cela a été fait pour d'autres travaux [62].

7.1.5 Émotion dans le monde réel

La reconnaissance des émotions est un problème résolu dans un cadre contrôlé et en se basant sur des expressions faciales exagérées [185]. En revanche, les techniques qui obtiennent d'excellents résultats dans ce contexte deviennent bien moins efficaces lorsque les expressions sont plus naturelles et dans des conditions plus réalistes (mouvements, occlusions du visage, *etc.*) [80], ce qui implique l'utilisation de nouvelles approches, permettant notamment de construire des représentations plus robustes des émotions.

Contributions Cette notion de robustesse a été traitée dans les Chapitres 3 car nous avons été confronté à des vidéos enregistrées dans des conditions très réalistes et nécessitant une véritable perception multimodale pour être correctement analysées. De plus, nous nous sommes concentrés sur la modalité particulière de l'expression faciale et avons veillé à obtenir un modèle avec une meilleure robustesse de représentation, avec notamment une meilleure stabilité temporelle des prédictions, mais également une représentation capable de généraliser sur différents domaines, comme étudié dans le Chapitre 4.

Nous avons également utilisé des ensembles de données très larges disposant d'une annotation de l'expression faciale issue de l'espace latent d'un réseau de neurones préalablement entraîné à reconnaître des émotions discrètes (*e.g.* colère, joie). Cette annotation automatique et continue a notamment permis d'obtenir un modèle de modification de l'expression faciale applicable dans des conditions réelles, du fait du très grand nombre d'exemples observés. Le Chapitre 5 a permis de dépasser l'état de l'art actuel (2019) en reconnaissance d'expression faciale sur RAF, notamment car celui-ci exploite des connaissances issues de diverses modèles, permettant potentiellement de compenser plus facilement le bruit dû à des conditions réalistes.

Enfin, l'exploitation de la multimodalité de l'émotion a également été étudiée rapidement dans les Chapitre 6, permettant d'obtenir des résultats encourageants sur la base de données AFEW, bien que ceux-ci n'aient pas pu être validés sur l'ensemble de test officiel mais seulement sur l'ensemble de validation disponible.

Limitations Nous avons proposé une solution de reconnaissance de l'expression faciale particulièrement robuste et performante. Pour améliorer notre approche multimodale de reconnaissance de l'émotion, nous pourrions consacrer autant d'attention à d'autres modalités, telles que les expressions vocale ou corporelle. De plus, les évaluations de reconnaissance multimodale de l'émotion ont été produites à partir d'une base de données de faible taille, avec des biais spécifiques (distribution des classes très déséquilibrée, bruit d'annotation, *etc.*). Pour complétement valider notre modèle, il serait donc pertinent de chercher à l'appliquer sur d'autres bases de données multimodales [45, 168], de la même manière que nous l'avons fait pour la reconnaissance d'expression faciale.

7.2 Perspectives générales

Après avoir détaillé nos contributions et certaines de leurs limitations, ouvrant à quelques perspectives à court terme, nous présentons des perspectives plus générales qui en découlent. Nous proposons tout d'abord d'élargir et d'améliorer notre système de reconnaissance de l'émotion, en considérant de nouvelles expressions et de nouveaux contenus. Ensuite, un élément observable de manière récurrente dans nos approches est la prise en compte limitée de la séquentialité. Cela nous conduit naturellement vers une extension de nos contributions pour explorer cette dimension essentielle à la résolution de nombreux problèmes. Nous proposons également quelques pistes relatives à la formulation même d'un problème multimodal, avec la nécessité de prendre en compte une interaction séquentielle entre les modalités. Enfin, nous revenons sur la question du transfert des connaissances et de l'apprentissage d'un modèle plus général.

7.2.1 Intérêt des expressions de l'émotion

Dans ce manuscrit, bien qu'utilisant différentes expressions de l'émotion à travers le Chapitre 3, nous nous sommes concentrés plus spécifiquement sur l'expression faciale. Comme explicité dans les limitations, il serait donc intéressant d'apporter des contributions sur des modalités moins exploitées dans la littérature. Un point important est alors d'identifier les raisons pour lesquelles les expressions vocales et corporelles ont été peu utilisées (par les approches neuronales) jusqu'ici.

Cela peut d'abord s'expliquer par le fait que la plupart des bases de données contenant du son ou des corps entiers sont de faibles tailles ou ne sont pas annotées en émotion. Il est alors difficile d'apprendre un réseau de neurones sur celles-ci. Au contraire, pour l'expression faciale, nous disposons de larges

bases de données de visages statiques, dont la tâche d'annotation a été beaucoup plus aisée que pour les autres modalités. Une autre raison est le fait qu'il est difficile de mesurer une émotion qu'à partir de l'expression corporelle, celle-ci pouvant se révéler beaucoup plus subtile qu'une simple expression faciale.

Des solutions pour améliorer la reconnaissance de l'expression vocale [6] ont consisté à annoter un large ensemble de vidéos de célébrités (VoxCeleb [201]) grâce à un modèle de reconnaissance de l'expression faciale, puis à apprendre à partir de ces annotations un réseau de neurones utilisant la partie audio des vidéos, construisant ainsi à faible coût un modèle de reconnaissance de l'expression vocale.

Une méthode similaire pourrait être utilisée dans le cadre de l'expression corporelle, à condition d'identifier suffisamment de données contenant des corps entiers dans un contexte émotionnel. Il serait alors possible d'apprendre à reconnaitre des expressions corporelles en se basant sur des annotations de l'expression faciale.

Enfin, pour mieux comprendre les différentes expressions et pouvoir créer certains liens entre elles, le contexte (*e.g.* la scène visuelle, les objets présents, la scène audio) pourrait se révéler extrêmement utile. C'est ce que cherchent à illustrer Lee *et al.* [168], qui ont notamment proposé très récemment une base de données de 13 000 vidéos annotées en émotion discrète et contenant des éléments contextuels nécessaires à la compréhension de l'émotion. Une perspective prometteuse serait alors d'adapter un processus de distillation mutuelle [321] au contexte multimodal, à partir de ces 13 000 vidéos, en entraînant simultanément : (a) un modèle de reconnaissance de l'expression faciale déjà pré-entraîné, (b) un modèle de reconnaissance de l'expression vocale (pouvant être pré-entraîné par exemple sur VoxCeleb), (c) un modèle de l'expression corporelle (pouvant être pré-entraîné sur des tâches de détection de squelettes [169]), (d) un ou des modèles contextuels (pouvant être pré-entraînés sur des tâches de reconnaissances d'objets, de description de scènes, de reconnaissance de son, *etc.*).

Les modèles ainsi entraînés permettraient probablement d'améliorer les performances en reconnaissance de l'émotion, mais également de mieux étudier les liens entre les différentes modalités d'expression de l'émotion, en s'approchant d'un processus de traitement plus humain ou tout du moins véritablement multimodal.

7.2.2 Prise en compte des données séquentielles

Nous avons été confrontés à différents types de données temporelles tout au long du manuscrit : son, séquences d'images RGB de différents type (visages, scène avec des êtres humains, animaux), d'articulation du corps, de cartes de profondeur, *etc.* . Nous avons consacré une partie du Chapitre 3 à l'exploitation de cette temporalité, *i.e.* cet ordre temporel et nous avons abouti dans le cas de la reconnaissance des émotions à une conclusion déjà observée par les gagnants de la compétition Youtube8M [196] et assez surprenante : l'ordre dans lequel la séquence de visages est observée ne change pas la performance. Et l'apprentissage de modèles temporelles complexes telles que les LSTM n'aboutit donc pas à une amélioration dans le cas de AFEW. Nous nous sommes donc contentés tout au long du manuscrit de résoudre le problème de la réduction temporelle en moyennant toutes les représentations en une seule. Des techniques assimilables à du clustering, telles que NetVLAD [15] ont été employées avec succès dans des contextes comme le challenge Youtube8M et pourraient s'avérer pertinentes dans notre cas.

Néanmoins, il est tout à fait possible que cette absence de bénéfice dans l'exploitation de l'ordre temporel soit spécifique à certaines problématiques. En effet, les deux tâches où l'ordre ne semble que peu importer sont la classification d'émotion et la classification d'objet (au sens large) dans des vidéos. Il s'agit de tâches très particulières et évaluées sur des bases de données spécifiques, avec des vidéos relativement courtes. Par exemple, comme proposé notamment par Barros *et al.* [31], la reconnaissance d'émotion pourrait être considérée à une échelle plus longue et avec des transitions plus continues entre différents états émotionnels. Pour ce type de tâches, il semble probable que l'ordre temporel deviendrait tout à fait pertinent et apporterait beaucoup. Notons que les bases de données permettant de résoudre ce type de problèmes restent malgré tout de faible taille, du fait du coup important de l'annotation, ce qui implique forcément des difficultés à entraîner un modèle sans sur-apprentissage. Comme évoqué dans les perspectives du Chapitre 3, il serait possible d'utiliser des approches semi-supervisées pour diminuer

ce problème.

Une autre explication possible serait que l'absence d'ordre temporel provient de l'extracteur de la représentation. En effet, dans les deux exemples que nous avons cités, l'ordre temporel était recherché au niveau de représentations extraites par un réseau déjà entraîné. Ces représentations ont été apprises à partir d'images, par nature statiques et il est donc possible qu'elles effacent implicitement des éléments pertinents pour déterminer un ordre temporel entre les images. Nous pouvons donc entrevoir à nouveau un problème similaire à celui que nous avons cherché à résoudre dans le cadre multimodal : existe-t-il un niveau de représentation plus pertinent que les autres pour effectuer la fusion temporelle ? Faut-il utiliser différentes couches de représentations de l'extracteur de représentations ?

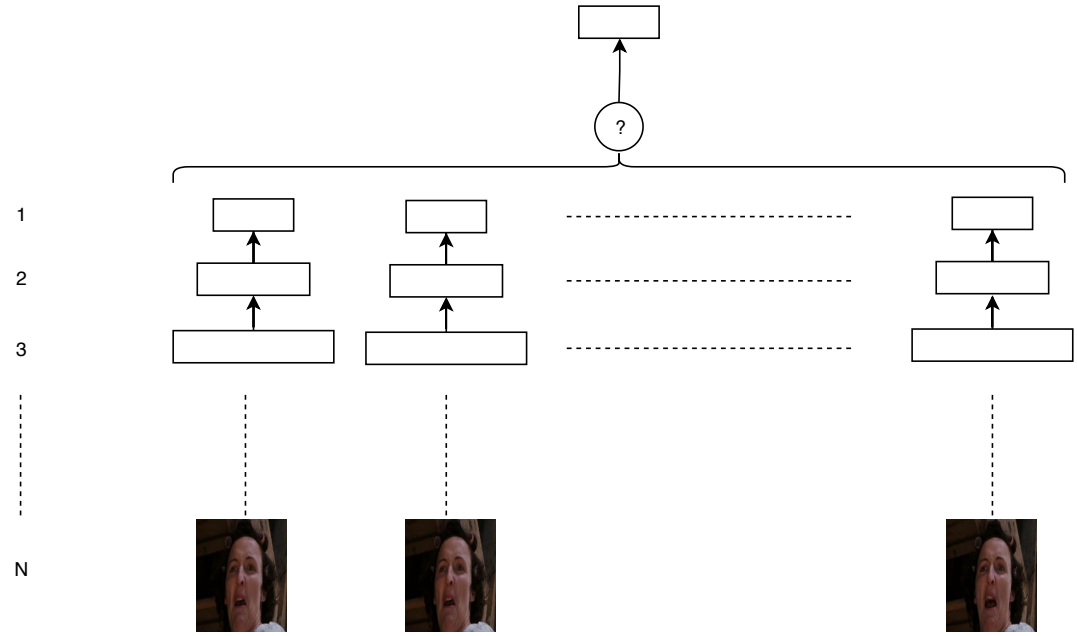


FIGURE 7.2.1 – Illustration de configurations de fusion temporelle : différents niveaux de représentations extraits par un CNN peuvent être utilisés et différents opérateurs de fusion temporelles (correspondant ici au point d'interrogation entouré) peuvent y être appliqués. Les visages sont issus de la base de données AFEW

Pour répondre à ces questions, il serait possible d'exploiter une méthode de recherche d'architecture, par exemple similaire à EPNAS. L'espace de recherche consisterait alors en différentes configurations illustrées en Figure 7.2.1 définies par (a) un choix du niveau i de représentation à utiliser et (b) un opérateur de fusion temporelle f (moyenne, moyenne pondérée, LSTM, attention, NetVLAD, *etc.*). Il serait alors possible de construire progressivement une configuration plus complexe, en combinant les représentations fusionnées obtenues par différents couples (i, f).

De plus, comme évoqué dans la conclusion du Chapitre 6, il serait alors possible d'intégrer cette recherche de fusion temporelle au sein de la recherche d'une architecture de fusion multimodale, en cherchant notamment à résoudre le problème non trivial de l'ordre dans lequel les deux opérations doivent être effectuées : fusion temporelle sur les réseaux unimodaux puis fusion multimodale ? Ou fusion multimodale pour chacun des composants des séquences, puis fusion temporelle des représentations résultantes ? Voire peut-être un procédé hybride, permettant de mêler les deux types de fusions.

7.2.3 Des liens avec d'autres types de problèmes multimodaux

Comme déjà évoqué dans les limitations relatives à nos contributions en fusion multimodale, les problèmes que nous avons présentés ont supposé une perception simultanée des différentes modalités. Mais cette perception n'est pas systématique dans le monde réel, puisque beaucoup de comportements sociaux découlent d'une interaction. Il est alors plus probable que la perception des modalités soit sé-

quentielle, *i.e.* qu'une modalité soit perçue en amont d'une autre, afin d'en piloter la perception. Par exemple, le *Visual Question Answering* [14] consiste à répondre à une question (modalité textuelle) à partir d'une modalité visuelle. Il est alors nécessaire de développer d'autres techniques de fusion permettant de construire des liens spécifiques entre les modalités et d'ainsi retrouver les informations visuelles souhaitées. Une variété de méthodes différentes est ainsi proposé dans ce domaine. Des extensions telles que le Video Question Answering [313] sont aussi proposées, consistant à rechercher la réponse à une question à partir de contenus temporels et multimodaux (séquence d'images + texte par exemple).

L'ensemble des techniques utilisées pour résoudre ces problèmes (fusion bilinéaire [153], mécanismes d'attention [10], factorisation [34]) présentent tout de même des points communs avec certaines approches appliquées à des problèmes plus classiques. Il serait donc intéressant de chercher à analyser les causes de cette différence d'utilisation : est-ce uniquement dû à la nature du problème ? Ou cela pourrait-il provenir d'une non-communication entre les deux communautés (*i.e.* fusion audiovisuelle et VQA) ?

Pour répondre à ces interrogations, il serait pertinent d'effectuer des recherches d'architectures multimodales dans le cadre du VQA, ou tout du moins en incluant les différents opérateurs de fusion proposés par les méthodes de VQA, et d'ainsi pouvoir mieux identifier si les améliorations apportées par ces opérateurs s'appliquent à un domaine spécifique ou peuvent également s'étendre à d'autres problématiques.

Nous avons utilisé le cadre du VQA comme perspective, puisqu'il s'agit d'un problème actuellement très exploré et donc où de nombreuses ressources sont à disposition. Néanmoins, il existe de nombreuses autres extensions vers d'autres types de problèmes multimodaux où il est nécessaire d'avoir une perception multimodale asynchrone. Nous pouvons plus spécifiquement citer le domaine large des interactions dyadiques [52], *i.e.* des dialogues, qui cherche à prendre en compte le fait que nos expressions (voire notre "manière d'être") sont influencées par notre interlocuteur, et vice versa.

En revenant au cas d'application particulier de l'analyse des émotions, cela permettrait par exemple de mieux les différencier en leur donnant un contexte social et réaliste, amenant à des distinctions liées à de véritables mécanismes internes, *i.e.* physiques. De plus, des recherches récentes [36] ont montré que la composante couleur seule du visage permettrait de déterminer très facilement si une expression faciale est suscitée par une émotion réelle ou jouée, cela s'expliquant par le fait que l'émotion aurait des conséquences physiques sur notre visage, au-delà de simples activations musculaires. Il serait alors possible de reconsidérer nos approches actuelles dans un contexte de dialogue, en considérant émotion réelle et jouée et en donnant ainsi un degré de complexité supérieure aux émotions reconnues.

Annexe A

Architectures unimodales pour CentralNet

Notations spécifiques à l'annexe **Conv** désigne un bloc convolutif, composé d'une convolution, d'une BN et d'une activation (reLU si non précisé). **Perceptron N.L.** désigne un perceptron suivi d'une BN et d'une activation (reLU si non précisé). **Perceptron** désigne un perceptron classique.

AV-MNIST Nous avons utilisé un réseau LeNet-5 [167] pour l'image. Pour l'audio, nous avons implémenté un réseau similaire au LeNet-5, mais avec deux blocs convolutifs supplémentaires au niveau de l'entrée, pour passer du spectrogramme en 112×112 à la sortie de la conv3 de dimension $14 \times 14 \times 32$, dimension égale à celle en sortie de la conv1 du LeNet-5.

Image		Fusion		Audio	
Couche	Dimensions	Couche	Dimension	Couche	Dimension
				Conv1	56 x 56 x 8
				Conv2	28 x 28 x 16
Conv1	14 x 14 x 32			Conv3	14 x14 x 32
Conv2	7 x 7 x 64	Conv	7 x 7 x 64	Conv4	7 x 7 x 64
Perceptron N.L.	1024	Perceptron N.L.	1024	Perceptron N.L.	1024
Perceptron	10	Perceptron	10	Perceptron	10

TABLE A.0.1 – Architecture complète pour AV-MNIST

Montalbano Les architectures de fusion sont identiques pour chacune des quatre modalités. Il s'agit d'un MLP composé de 3 couches de perceptron N.L. , passant par les dimensions 400, 128, 42 et d'un perceptron passant de 42 à 21. The fusion architecture includes one multilayer perceptron per modality, each having 3 layers of size : 400×128, 128×42, 42×21. De même pour le réseau central.

mmIMDb Nous avons utilisé la même approche que les auteurs de mmIMDb et avons donc appliqué des MLP sur les modalités extraites sous forme de descripteurs compactes.

Texte		Fusion		Image	
Couche	Dimensions	Couche	Dimensions	Couche	Dimensions
Perceptron N.L.	2048	Perceptron N.L.	2048	Perceptron N.L.	2048
Perceptron N.L.	512	Perceptron N.L.	512	Perceptron N.L.	512
Perceptron	23	Perceptron	23	Perceptron	23

TABLE A.0.2 – Architecture complète pour mmIMDb.

AFEW Nous avons utilisé le *ResNet-18* du Chapitre 3 comme base pour la modalité Image et un simple extracteur de descripteurs OpenSmile [97] pour la modalité audio. L'ensemble est traité par des MLP.

Image		Fusion		Audio	
Couche	Dimensions	Couche	Dimensions	Couche	Dimensions
Perceptron N.L.	512	Perceptron N.L.	2048	Perceptron N.L.	512
Perceptron N.L.	128	Perceptron N.L.	128	Perceptron N.L.	128
Perceptron	7	Perceptron	7	Perceptron	7

TABLE A.0.3 – Architecture complète pour AFEW.

Animaux Le modèle visuel est un Inflated Resnet-50 [54] et le modèle audio est composé de six conv, d'un pooling (7 par 7) et d'un perceptron. Le réseau central possède la même architecture que le réseau audio en enlevant les deux premiers conv. . Pour gérer la dimension temporelle dans la somme pondérée, les représentations sont moyennées sur l'axe temporel.

NTU-RGB+D Le modèle visuel est un Inflated Resnet-50 [54] tandis que le modèle appliqué au squelette est un réseau convolutif avec un module dit de co-occurence [169], permettant de prendre en compte plus particulièrement la différence entre deux instants t. Le modèle central est un réseau convolutif similaire au modèle appliqué au squelette, mais sans le module de co-occurence et sans les deux premiers conv. .

Bibliographie

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m : A large-scale video classification benchmark. In *CVPR Workshop*, 2016.

[2] D. Acharya, Z. Huang, D. Paudel, and L. Van Gool. Covariance pooling for facial expression recognition. *CVPR Workshop*, 2018.

[3] A. Achille, M. Lam, R. Tewari, A. Ravichandran, S. Maji, C. Fowlkes, S. Soatto, and P. Perona. Task2vec : Task embedding for meta-learning. *arXiv preprint arXiv :1902.03545*, 2019.

[4] M. A. Aizerman. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25 :821–837, 1964.

[5] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected papers of hirotugu akaike*. 1998.

[6] S. Albanie, A. Nagrani, A. Vedaldi, and A. Zisserman. Emotion recognition in speech using cross-modal transfer in the wild. *ACMM*, 2018.

[7] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *T-PAMI*, 2010.

[8] E. Alpaydin. *Introduction to machine learning*. MIT press, 2009.

[9] M. R. Amer, T. Shields, B. Siddiquie, A. Tamrakar, A. Divakaran, and S. Chai. Deep multimodal fusion : A hybrid approach. In *IJCV*, 2018.

[10] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[11] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *ICML*, 2013.

[12] G. Antipov, M. Baccouche, S.-A. Berrani, and J.-L. Dugelay. Apparent age estimation from face images combining general and children-specialized deep learning models. In *CVPR Workshop*, 2016.

[13] G. Antipov, M. Baccouche, S.-A. Berrani, and J.-L. Dugelay. Effective training of convolutional neural networks for face-based gender and age prediction. *Pattern Recognition*, 2017.

[14] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa : Visual question answering. In *ICCV*, 2015.

[15] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad : Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016.

[16] R. Arandjelovic and A. Zisserman. Look, listen and learn. In *ICCV*, 2017.

[17] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *arXiv preprint arXiv :1908.02983*, 2019.

[18] J. Arevalo, T. Solorio, M. Montes-y Gómez, and F. A. González. Gated multimodal units for information fusion. In *ICLR Workshop*, 2017.

[19] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv :1701.07875*, 2017.

[20] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli. Multimodal fusion for multimedia analysis : a survey. *Multimedia Systems*, 2010.

[21] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet : Learning sound representations from unlabeled video. In *NEURIPS*, 2016.

[22] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *International workshop on human behavior understanding*, 2011.

[23] F. R. Bach, G. R. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *ICML*, 2004.

[24] D. H. Ballard. Modular learning in neural networks. In *AAAI*, 1987.

[25] T. Baltrušaitis, C. Ahuja, and L.-P. Morency. Multimodal machine learning : A survey and taxonomy. *T-PAMI*, 2019.

[26] F. Baradel, C. Wolf, J. Mille, and G. W. Taylor. Glimpse clouds : Human activity recognition from unstructured feature points. In *CVPR*, 2018.

[27] I. Baran, O. Kupyn, and A. Kravchenko. Safe augmentation : Learning task-specific transformations from data. *arXiv preprint arXiv :1907.12896*, 2019.

[28] S. A. Bargal, E. Barsoum, C. C. Ferrer, and C. Zhang. Emotion recognition in the wild from videos using images. In *ICMI*, 2016.

[29] L. F. Barrett, B. Mesquita, and M. Gendron. Context in emotion perception. *Current Directions in Psychological Science*, 2011.

[30] L. F. Barrett and J. A. Russell. The structure of current affect : Controversies and emerging consensus. *Current directions in psychological science*, 1999.

[31] P. Barros, N. Churamani, E. Lakomkin, H. Siqueira, A. Sutherland, and S. Wermter. The omg-emotion behavior dataset. In *IJCNN*, 2018.

[32] C. D. Batson, L. L. Shaw, and K. C. Oleson. Differentiating affect, mood, and emotion : toward functionally based conceptual distinctions. *Current Directions in Psychological Science*, 1992.

[33] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *JMLR*, 2010.

[34] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome. Mutan : Multimodal tucker fusion for visual question answering. In *CVPR*, 2017.

[35] C. F. Benitez-Quiroz, R. Srinivasan, Q. Feng, Y. Wang, and A. M. Martinez. Emotionet challenge : Recognition of facial expressions of emotion in the wild. *arXiv preprint arXiv :1703.01210*, 2017.

[36] C. F. Benitez-Quiroz, R. Srinivasan, and A. M. Martinez. Facial color is an efficient mechanism to visually transmit emotion. *Proceedings of the National Academy of Sciences*, 2018.

[37] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *JMLR*, 2012.

[38] B. Biancardi, A. Cafaro, and C. Pelachaud. Could a virtual agent be warm and competent ? investigating user's impressions of agent's non-verbal behaviours. In *CHI Workshops*, 2017.

[39] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[40] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. In *Computer Graphics Forum*, 2003.

[41] A. Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 2019.

[42] A. Borovykh, S. Bohte, and C. W. Oosterlee. Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv :1703.04691*, 2017.

[43] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT'2010*. 2010.

[44] L. Breiman. Random forests. *JMLR*, 2001.

[45] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan. Iemocap : Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 2008.

[46] C. Busso, Z. Deng, S. Yildirim, M. Bulut, C. M. Lee, A. Kazemzadeh, S. Lee, U. Neumann, and S. Narayanan. Analysis of emotion recognition using facial expressions, speech and multimodal information. In *ACMM*, 2004.

[47] J. Cai, Z. Meng, A. S. Khan, Z. Li, J. O'Reilly, and Y. Tong. Island loss for learning discriminative features in facial expression recognition. In *FG*, 2018.

[48] C. Cangea, P. Veličković, and P. Liò. Xflow : 1d-2d cross-modal deep neural networks for audio-visual classification. *arXiv preprint arXiv :1709.00572*, 2017.

[49] J. Cao, Y. Li, and Z. Zhang. Partially shared multi-task convolutional neural network with local constraint for face attribute learning. In *CVPR*, 2018.

[50] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2 : A dataset for recognising faces across pose and age. In *FG*, 2018.

[51] W. Cao, V. Mirjalili, and S. Raschka. Consistent rank logits for ordinal regression with convolutional neural networks. *arXiv preprint arXiv :1901.07884*, 2019.

[52] J. N. Cappella. Mutual influence in expressive behavior : Adult–adult and infant–adult dyadic interaction. *Psychological Bulletin*, 1981.

[53] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[54] J. Carreira and A. Zisserman. Quo vadis, action recognition ? a new model and the kinetics dataset. In *CVPR*, 2017.

[55] R. Caruana. Multitask learning. *JMLR*, 1997.

[56] S. Chandar, M. M. Khapra, H. Larochelle, and B. Ravindran. Correlational neural networks. *Neural computation*, 2016.

[57] L. Chao, J. Tao, M. Yang, Y. Li, and Z. Wen. Audio visual emotion recognition with temporal alignment and perception attention. *arXiv preprint arXiv :1603.08321*, 2016.

[58] Y. Chebotar and A. Waters. Distilling knowledge from ensembles of neural networks for speech recognition. In *Interspeech*, 2016.

[59] J. Chen, Z. Chen, Z. Chi, and H. Fu. Emotion recognition in the wild with feature fusion and multiple kernel learning. In *ICMI*, 2014.

[60] L. Chen, S. Srivastava, Z. Duan, and C. Xu. Deep cross-modal audio-visual generation. In *ACMM Workshop*, 2017.

[61] M. Chen, S. Wang, P. P. Liang, T. Baltrušaitis, A. Zadeh, and L.-P. Morency. Multimodal sentiment analysis with word-level fusion and reinforcement learning. In *ICMI*, 2017.

[62] R. Chen, H. Chen, G. Huang, J. Ren, and Q. Zhang. Explaining neural networks semantically and quantitatively. *arXiv preprint arXiv :1812.07169*, 2018.

[63] S. Chen, C. Zhang, and M. Dong. Coupled end-to-end transfer learning with generalized fisher information. In *CVPR*, 2018.

[64] Y. Cheng, D. Wang, P. Zhou, and T. Zhang. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv :1710.09282*, 2017.

[65] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[66] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan : Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018.

[67] B. Chu, V. Madhavan, O. Beijbom, J. Hoffman, and T. Darrell. Best practices for fine-tuning visual classifiers to new domains. In *ECCV Workshop*, 2016.

[68] J. S. Chung and A. Zisserman. Lip reading in the wild. In *ACCV*, 2016.

[69] R. Collobert and J. Weston. A unified architecture for natural language processing : Deep neural networks with multitask learning. In *ICML*, 2008.

[70] C. Cortes and V. Vapnik. Support-vector networks. *JMLR*, 1995.

[71] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment : Learning augmentation policies from data. 2019.

[72] C. Darwin and P. Prodger. *The expression of the emotions in man and animals*. Oxford University Press, USA, 1998.

[73] A. Das, A. Dantcheva, and F. Bremond. Mitigating bias in gender, age and ethnicity classification : A multi-task convolution neural network approach. In *ECCV Workshop*, 2018.

[74] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet : A large-scale hierarchical image database. In *CVPR*, 2009.

[75] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface : Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[76] L. Deng, G. Hinton, and B. Kingsbury. New types of deep neural network learning for speech recognition and related applications : An overview. In *ICASSP*, 2013.

[77] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv :1708.04552*, 2017.

[78] A. Dhall, R. Goecke, S. Ghosh, J. Joshi, J. Hoey, and T. Gedeon. From individual to group-level emotion recognition : Emotiw 5.0. In *ICMI*, 2017.

[79] A. Dhall, R. Goecke, J. Joshi, J. Hoey, and T. Gedeon. Emotiw 2016 : Video and group-level emotion recognition challenges. In *ICMI*, 2016.

[80] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Static facial expression analysis in tough conditions : Data, evaluation protocol and benchmark. In *ICCV Workshop*, 2011.

[81] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Collecting large, richly annotated facial-expression databases from movies. *IEEE MultiMedia*, 2012.

[82] A. Dhall, O. Ramana Murthy, R. Goecke, J. Joshi, and T. Gedeon. Video and image based emotion recognition challenges in the wild : Emotiw 2015. In *ICMI*, 2015.

[83] S. Dieleman and B. Schrauwen. End-to-end learning for music audio. In *ICASSP*, 2014.

[84] H. Ding, K. Sricharan, and R. Chellappa. Exprgan : Facial expression editing with controllable expression intensity. In *aaai*, 2018.

[85] C. Doersch and A. Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017.

[86] S. Du, Y. Tao, and A. M. Martinez. Compound facial expressions of emotion. *Proceedings of the National Academy of Sciences*, 2014.

[87] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 2011.

[88] P. Ekman. Are there basic emotions ? *Psychological review*, 1992.

[89] P. Ekman and W. V. Friesen. Facial action coding system. 1977.

[90] R. Ekman. *What the face reveals : Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997.

[91] M. El Ayadi, M. S. Kamel, and F. Karray. Survey on speech emotion recognition : Features, classification schemes, and databases. *Pattern Recognition*, 2011.

[92] T. Elsken, J. H. Metzen, and F. Hutter. Neural architecture search : A survey. *JMLR*, 2019.

[93] M. Engilberge, L. Chevallier, P. Pérez, and M. Cord. Finding beans in burgers : Deep semantic-visual embedding with localization. In *CVPR*, 2018.

[94] S. Escalera, X. Baró, J. Gonzalez, M. A. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H. J. Escalante, J. Shotton, and I. Guyon. Chalearn looking at people challenge 2014 : Dataset and results. In *ECCV Workshop*, 2014.

[95] V. Escorcia, J. Carlos Niebles, and B. Ghanem. On the relationship between visual attributes and convolutional networks. In *CVPR*, 2015.

[96] G. Evangelopoulos, A. Zlatintsi, A. Potamianos, P. Maragos, K. Rapantzikos, G. Skoumas, and Y. Avrithis. Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention. *Ta-M*, 2013.

[97] F. Eyben, M. Wöllmer, and B. Schuller. Opensmile : the munich versatile and fast open-source audio feature extractor. In *ACMM*, 2010.

[98] C. Fabian Benitez-Quiroz, R. Srinivasan, and A. M. Martinez. Emotionet : An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In *CVPR*, 2016.

[99] Y. Fan, X. Lu, D. Li, and Y. Liu. Video-based emotion recognition using cnn-rnn and c3d hybrid networks. In *ICMI*, 2016.

[100] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[101] K. Fukushima. Neocognitron : A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 1980.

[102] R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *COLT*, 2015.

[103] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set : An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017.

[104] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget : Continual prediction with lstm. Technical report, 1999.

[105] R. C. Geyer, V. Wegmayr, and L. Corinzia. Transfer learning by adaptive merging of multiple models. In *MIDL*, 2019.

[106] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[107] R. Girshick. Fast r-cnn. In *ICCV*, 2015.

[108] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *JMLR*, 2010.

[109] I. Goodfellow. Generative adversarial networks. In *NEURIPS*, 2016.

[110] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[111] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NEURIPS*, 2014.

[112] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, et al. Challenges in representation learning : A report on three machine learning contests. In *ICONIP*, 2013.

[113] A. Graves, S. Fernández, and J. Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. *Artificial Neural Networks : Formal Models and Their Applications–ICANN 2005*, 2005.

[114] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 2010.

[115] Z. Gu, B. Lang, T. Yue, and L. Huang. Learning joint multimodal representation based on multi-fusion deep neural networks. In *ICONIP*, 2017.

[116] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m : A dataset and benchmark for large-scale face recognition. In *ECCV*, 2016.

[117] D. Han, J. Kim, and J. Kim. Deep pyramidal residual networks. In *CVPR*, 2017.

[118] H. Han, C. Otto, and A. K. Jain. Age estimation from face images : Human vs. machine performance. In *ICB*, 2013.

[119] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis : An overview with application to learning methods. *Neural computation*, 2004.

[120] A. W. Harley. An interactive node-link visualization of convolutional neural networks. In *ISVC*, 2015.

[121] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers : Surpassing human-level performance on imagenet classification. In *CVPR*, 2015.

[122] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[123] G. E. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv :1503.02531*, 2015.

[124] Z. Hodari, O. Watts, S. Ronanki, and S. King. Learning interpretable control dimensions for speech synthesis by using external data. *Interspeech*, 2018.

[125] C. Hori, T. Hori, T.-Y. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi. Attention-based multimodal fusion for video description. In *CVPR*, 2017.

[126] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[127] J.-F. Hu, W.-S. Zheng, J. Pan, J. Lai, and J. Zhang. Deep bilinear learning for rgb-d action recognition. In *ECCV*, 2018.

[128] P. Hu, D. Cai, S. Wang, A. Yao, and Y. Chen. Learning supervised scoring ensemble for emotion recognition in the wild. In *ICMI*, 2017.

[129] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[130] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild : A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[131] X. Huo, X. S. Ni, and A. K. Smith. A survey of manifold-based learning methods. *Recent advances in data mining of enterprise data*, 2007.

[132] A. Hyvärinen, J. Karhunen, and E. Oja. Independent component analysis. john willey & sons. *Inc., New York*, 2001.

[133] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer. Densenet : Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv :1404.1869*, 2014.

[134] S. Ioffe and C. Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[135] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Label propagation for deep semi-supervised learning. In *CVPR*, 2019.

[136] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. *Adaptive mixtures of local experts*. MIT Press, 1991.

[137] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *T-PAMI*, 2018.

[138] I. Jolliffe. *Principal component analysis*. Springer, 2011.

[139] M. Kächele, M. Schels, S. Meudt, G. Palm, and F. Schwenker. Revisiting the emotiw challenge : how wild is it really ? *Journal on Multimodal User Interfaces*, 2016.

[140] S. E. Kahou, X. Bouthillier, P. Lamblin, C. Gulcehre, V. Michalski, K. Konda, S. Jean, P. Froumenty, Y. Dauphin, N. Boulanger-Lewandowski, et al. Emonets : Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, 2016.

[141] G. Kang, J. Li, and D. Tao. Shakeout : A new regularized deep neural network training scheme. In *AAAI*, 2016.

[142] M. Kang, K. Ji, X. Leng, and Z. Lin. Contextual region-based convolutional neural network with multilayer fusion for sar ship detection. *Remote Sensing*, 2017.

[143] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *ICML*, 2011.

[144] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[145] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ICLR*, 2018.

[146] H. Kaya, F. Gürpınar, and A. A. Salah. Video-based emotion recognition in the wild using deep transfer learning and score fusion. *Image and Vision Computing*, 2017.

[147] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi. A survey of the recent architectures of deep convolutional neural networks. *arXiv preprint arXiv :1901.06032*, 2019.

[148] P. Khorrami, T. Paine, and T. Huang. Do deep neural networks learn facial action units when doing expression recognition ? In *ICCV Workshop*, 2015.

[149] D. Kiela, E. Grave, A. Joulin, and T. Mikolov. Efficient large-scale multi-modal classification. In *AAAI*, 2018.

[150] B.-K. Kim, J. Roh, S.-Y. Dong, and S.-Y. Lee. Hierarchical committee of deep convolutional neural networks for robust facial expression recognition. *Journal on Multimodal User Interfaces*, 2016.

[151] D. H. Kim, M. K. Lee, D. Y. Choi, and B. C. Song. Multi-modal emotion recognition using semi-supervised learning and multiple neural networks in the wild. In *ICMI*, 2017.

[152] H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Nießner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt. Deep video portraits. In *Siggraph*, 2018.

[153] J.-H. Kim, J. Jun, and B.-T. Zhang. Bilinear attention networks. In *NEURIPS*, 2018.

[154] D. E. King. Dlib-ml : A machine learning toolkit. *JMLR*, 2009.

[155] D. Kingma and J. Ba. Adam : a method for stochastic optimization. *ICLR*, 2015.

[156] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[157] A. Kleinsmith and N. Bianchi-Berthouze. Affective body expression perception and recognition : A survey. *T-AC*, 2012.

[158] B. Knyazev, R. Shvetsov, N. Efremova, and A. Kuharenko. Leveraging large face recognition data for emotion classification. In *FG*, 2018.

[159] A. Kołakowska, A. Landowska, M. Szwoch, W. Szwoch, and M. R. Wróbel. Emotion recognition and its application in software engineering. In *HSI*, 2013.

[160] J. Kossaifi, G. Tzimiropoulos, S. Todorovic, and M. Pantic. Afew-va database for valence and arousal estimation in-the-wild. *Image and Vision Computing*, 2017.

[161] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NEURIPS*, 2012.

[162] A. Krogh and J. A. Hertz. A simple weight decay can improve generalization. In *NEURIPS*, 1992.

[163] C.-M. Kuo, S.-H. Lai, and M. Sarkis. A compact deep learning model for robust facial expression recognition. In *CVPR Workshop*, 2018.

[164] D. Lahat, T. Adali, and C. Jutten. Multimodal data fusion : an overview of methods, challenges, and prospects. *IEEE*, 2015.

[165] A. Lanitis and T. Cootes. Fg-net aging data base. *Cyprus College*, 2(3) :5, 2002.

[166] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 2015.

[167] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *IEEE*, 1998.

[168] J. Lee, S. Kim, S. Kim, J. Park, and K. Sohn. Context-aware emotion recognition networks. In *ICCV*, 2019.

[169] C. Li, Q. Zhong, D. Xie, and S. Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *IJCAI*, 2018.

[170] F. Li, N. Neverova, C. Wolf, and G. W. Taylor. Modout : Learning to fuse face and gesture modalities with stochastic regularization. In *FG*, 2017.

[171] S. Li. Measure, manifold, learning, and optimization : A theory of neural networks. *arXiv preprint arXiv :1811.12783*, 2018.

[172] S. Li, W. Deng, and J. Du. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *CVPR*, 2017.

[173] X. Li, H. Xiong, H. Wang, Y. Rao, L. Liu, and J. Huan. Delta : Deep learning transfer using feature map with attention for convolutional networks. In *ICLR*, 2019.

[174] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim. Fast autoaugment. *arXiv preprint arXiv :1905.00397*, 2019.

[175] A. Lindt, P. Barros, H. Siqueira, and S. Wermter. Facial expression editing with continuous emotion labels. In *FG*, 2019.

[176] Z. C. Lipton, J. Berkowitz, and C. Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv :1506.00019*, 2015.

[177] C. Liu, T. Tang, K. Lv, and M. Wang. Multi-feature based emotion recognition for video clips. In *ICMI*, 2018.

[178] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy. Progressive neural architecture search. In *ECCV*, 2018.

[179] H. Liu, K. Simonyan, and Y. Yang. Darts : Differentiable architecture search. *ICLR*, 2018.

[180] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[181] M. Long and J. Wang. Learning multiple tasks with deep relationship networks. *arXiv preprint arXiv :1506.02117*, 2015.

[182] X. Long, C. Gan, G. de Melo, X. Liu, Y. Li, F. Li, and S. Wen. Multimodal keyless attention fusion for video classification. In *AAAI*, 2018.

[183] E. Loth, L. Garrido, J. Ahmad, E. Watson, A. Duff, and B. Duchaine. Facial expression recognition as a candidate marker for autism spectrum disorder : how frequent and severe are deficits ? *Molecular Autism*, 2018.

[184] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *CVPR*, 2017.

[185] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+) : A complete dataset for action unit and emotion-specified expression. In *CVPR Workshop*, 2010.

[186] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. Matthews. Painful data : The unbc-mcmaster shoulder pain expression archive database. In *FG*, 2011.

[187] D. C. Luvizon, D. Picard, and H. Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *CVPR*, 2018.

[188] M. J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, and J. Budynek. The japanese female facial expression (jaffe) database. In *FG*, 1998.

[189] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013.

[190] S. Mallat. *Sciences des données*. Fayard, 2018.

[191] M. Mancini, S. R. Bulò, B. Caputo, and E. Ricci. Best sources forward : domain generalization through source-specific nets. In *ICIP*, 2018.

[192] D. Matsumoto, M. G. Frank, and H. S. Hwang. *Nonverbal communication : Science and applications : Science and applications*. Sage, 2013.

[193] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 1976.

[194] G. McKeown, M. F. Valstar, R. Cowie, and M. Pantic. The semaine corpus of emotionally coloured character interactions. In *ICME*, 2010.

[195] A. Mehrabian. Pleasure-arousal-dominance : A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 1996.

[196] A. Miech, I. Laptev, and J. Sivic. Learnable pooling with context gating for video classification. *CVPR Workshop*, 2017.

[197] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016.

[198] A. Mollahosseini, B. Hasani, and M. H. Mahoor. Affectnet : A database for facial expression, valence, and arousal computing in the wild. *T-AC*, 2017.

[199] A. Mordvintsev, C. Olah, and M. Tyka. Deepdream-a code example for visualizing neural networks. *Google Research*, 2015.

[200] C. Murdock, Z. Li, H. Zhou, and T. Duerig. Blockout : Dynamic model selection for hierarchical deep networks. In *CVPR*, 2016.

[201] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb : a large-scale speaker identification dataset. In *Interspeech*, 2017.

[202] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[203] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov. Semeval-2016 task 4 : Sentiment analysis in twitter. In *ACL*, 2016.

[204] P. Natarajan, S. Wu, S. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, R. Prasad, and P. Natarajan. Multimodal feature fusion for robust event detection in web videos. In *CVPR*, 2012.

[205] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2). In *Doklady AN USSR*, 1983.

[206] N. Neverova, C. Wolf, G. Taylor, and F. Nebout. Moddrop : adaptive multi-modal gesture recognition. *T-PAMI*, 2016.

[207] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout. Multi-scale deep learning for gesture detection and localization. In *ECCV Workshop*, 2014.

[208] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[209] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler. Deep learning for emotion recognition on small datasets using transfer learning. In *ICMI*, 2015.

[210] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *ICML*, 2011.

[211] A. Nguyen, J. Yosinski, and J. Clune. Understanding neural networks via feature visualization : A survey. *arXiv preprint arXiv :1904.08939*, 2019.

[212] R. Niewiadomski, S. J. Hyniewska, and C. Pelachaud. Constraint-based model for synthesis of multimodal sequential expressions of emotions. *T-AC*, 2011.

[213] F. Noroozi, D. Kaminska, C. Corneanu, T. Sapinski, S. Escalera, and G. Anbarjafari. Survey on emotional body gesture recognition. 2018.

[214] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[215] M. Noroozi, H. Pirsiavash, and P. Favaro. Representation learning by learning to count. In *ICCV*, 2017.

[216] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks : A review. *Neural Networks*, 2019.

[217] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *BMVC*, 2015.

[218] F. Pecune, B. Biancardi, Y. Ding, C. Pelachaud, M. Mancini, G. Varni, A. Camurri, and G. Volpe. Lol—laugh out loud. In *AAAI*, 2015.

[219] J.-M. Pérez-Rúa, M. Baccouche, and S. Pateux. Efficient progressive neural architecture search. In *BMVC*, 2018.

[220] J.-M. Pérez-Rúa, M. Baccouche, and S. Pateux. Efficient progressive neural architecture search. In *BMVC*, 2018.

[221] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean. Efficient neural architecture search via parameter sharing. In *ICML*, 2018.

[222] R. W. Picard. *Affective computing*. MIT press, 2000.

[223] K. J. Piczak. Esc : Dataset for environmental sound classification. In *ICME*, 2015.

[224] S. Pini, O. B. Ahmed, M. Cornia, L. Baraldi, R. Cucchiara, and B. Huet. Modeling multimodal cues in a deep learning-based framework for emotion recognition in the wild. In *ICMI*, 2017.

[225] R. Plutchik and H. Kellerman. *Theories of emotion*. Academic Press, 2013.

[226] G. Pons and D. Masip. Multi-task, multi-label and multi-domain learning with residual convolutional networks for emotion recognition. *arXiv preprint arXiv :1802.06664*, 2018.

[227] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer. Ganimation : Anatomically-aware facial animation from a single image. In *ECCV*, 2018.

[228] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 1999.

[229] F. Qiao, N. Yao, Z. Jiao, Z. Li, H. Chen, and H. Wang. Emotional facial expression transfer from a single image via generative adversarial nets. *Computer Animation and Virtual Worlds*, 2018.

[230] J. R. Quinlan et al. Bagging, boosting, and c4. 5. In *AAAI*, 1996.

[231] F. Radenovic, G. Tolias, and O. Chum. Deep shape matching. In *ECCV*, 2018.

[232] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, and K. He. Data distillation : Towards omni-supervised learning. In *CVPR*, 2018.

[233] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande. Massively multitask networks for drug discovery. *arXiv preprint arXiv :1502.02072*, 2015.

[234] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. *ICLR*, 2016.

[235] F. Ringeval, B. Schuller, M. Valstar, J. Gratch, R. Cowie, S. Scherer, S. Mozgai, N. Cummins, M. Schmitt, and M. Pantic. Avec 2017 : Real-life depression, and affect recognition workshop and challenge. In *ACMM Workshop*, 2017.

[236] F. Ringeval, A. Sonderegger, J. Sauer, and D. Lalanne. Introducing the recola multimodal corpus of remote collaborative and affective interactions. In *FG*, 2013.

[237] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets : Hints for thin deep nets. In *ICLR*, 2014.

[238] O. Ronneberger, P. Fischer, and T. Brox. U-net : Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[239] F. Rosenblatt. The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review*, 1958.

[240] R. Rothe, R. Timofte, and L. Van Gool. Dex : Deep expectation of apparent age from a single image. In *ICCV Workshop*, 2015.

[241] R. Rothe, R. Timofte, and L. Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *IJCV*, 2018.

[242] S. Ruder12, J. Bingel, I. Augenstein, and A. Søgaard. Sluice networks : Learning what to share between loosely related tasks. *STAT*, 2017.

[243] D. Rumerlhar. Learning representation by back-propagating errors. *Nature*, 1986.

[244] J. A. Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6) :1161, 1980.

[245] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NEURIPS*, 2016.

[246] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox. On the information bottleneck theory of deep learning. 2018.

[247] M. Scholz, M. Fraunholz, and J. Selbig. Nonlinear principal component analysis : neural network models and applications. In *Principal manifolds for data visualization and dimension reduction*. 2008.

[248] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet : A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[249] B. W. Schuller. Speech emotion recognition : Two decades in a nutshell, benchmarks, and ongoing trends. *Communications of the ACM*, 2018.

[250] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d : A large scale dataset for 3d human activity analysis. In *CVPR*, pages 1010–1019, 2016.

[251] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang. Deep multimodal feature analysis for action recognition in rgb+ d videos. *T-PAMI*, 2017.

[252] K. Simonyan and A. Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. In *NEURIPS*, 2014.

[253] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2014.

[254] A. Smola and S. Vishwanathan. Introduction to machine learning. *Cambridge University, UK*, 32 :34, 2008.

[255] C. G. Snoek, M. Worring, and A. W. Smeulders. Early versus late fusion in semantic video analysis. In *ACMM*, 2005.

[256] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *NEURIPS*, 2012.

[257] C. Soladié, N. Stoiber, and R. Séguier. Invariant representation of facial expressions for blended expression recognition on unknown subjects. *CVIU*, 2013.

[258] L. Song, Z. Lu, R. He, Z. Sun, and T. Tan. Geometry guided adversarial facial expression synthesis. In *ACMM*, 2018.

[259] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*, 2017.

[260] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout : a simple way to prevent neural networks from overfitting. *JMLR*, 2014.

[261] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 2002.

[262] B. Sun, L. Li, T. Zuo, Y. Chen, G. Zhou, and X. Wu. Combining multimodal features with hierarchical classifier fusion for emotion recognition in the wild. In *ICMI*, 2014.

[263] J. M. Susskind, G. E. Hinton, J. R. Movellan, and A. K. Anderson. Generating facial expressions with deep belief nets. *T-AC*, 2008.

[264] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[265] M. A. Tanner. *Tools for statistical inference : observed data and data augmentation methods*. Springer Science & Business Media, 2012.

[266] A. Tarvainen and H. Valpola. Mean teachers are better role models : Weight-averaged consistency targets improve semi-supervised deep learning results. In *NEURIPS*, 2017.

[267] B. Thompson. Canonical correlation analysis. *Encyclopedia of statistics in behavioral science*, 2005.

[268] Y.-I. Tian, T. Kanade, and J. F. Cohn. Recognizing action units for facial expression analysis. *T-PAMI*, 2001.

[269] N. Tishby. The information bottleneck theory of deep neural networks. In *APS Meeting Abstracts*, 2018.

[270] N. Tits, F. Wang, K. E. Haddad, V. Pagel, and T. Dutoit. Visualization and interpretation of latent spaces for controlling expressive speech synthesis through audio analysis. *arXiv preprint arXiv :1903.11570*, 2019.

[271] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[272] L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In *CVPR*, 2017.

[273] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou. Adieu features ? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *ICASSP*, 2016.

[274] T.-W. Tsai and M.-Y. Lin. An application of interactive game for facial expression of the autisms. In *International Conference on Technologies for E-Learning and Digital Entertainment*, 2011.

[275] Z. Tu. Auto-context and its application to high-level vision tasks. In *CVPR*, 2008.

[276] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. Mocogan : Decomposing motion and content for video generation. In *CVPR*, 2018.

[277] M. Valstar, J. Gratch, B. Schuller, F. Ringeval, D. Lalanne, M. Torres Torres, S. Scherer, G. Stratou, R. Cowie, and M. Pantic. Avec 2016 : Depression, mood, and emotion recognition workshop and challenge. In *ACMM Workshop*, 2016.

[278] M. F. Valstar, T. Almaev, J. M. Girard, G. McKeown, M. Mehu, L. Yin, M. Pantic, and J. F. Cohn. Fera 2015-second facial expression recognition and analysis challenge. In *FG*, 2015.

[279] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet : A generative model for raw audio. *SSW*, 2016.

[280] M. Van Vugt and A. E. Grabo. The many faces of leadership : an evolutionary-psychology approach. *Current Directions in Psychological Science*, 2015.

[281] R. Vilalta and Y. Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 2002.

[282] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders : Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 2010.

[283] V. Vukotić, C. Raymond, and G. Gravier. Generative adversarial networks for multimodal representation learning in video hyperlinking. In *ICMR*, 2017.

[284] P. Wang, W. Li, J. Wan, P. Ogunbona, and X. Liu. Cooperative training of deep aggregation networks for rgb-d action recognition. In *AAAI*, 2018.

[285] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, et al. Tacotron : Towards end-to-end speech synthesis. In *Interspeech*, 2017.

[286] Z. Wang, K. Kuan, M. Ravaut, G. Manek, S. Song, Y. Fang, S. Kim, N. Chen, L. F. D'Haro, L. A. Tuan, et al. Truly multi-modal youtube-8m video classification with video, audio, and text. In *CVPR Workshop*, 2017.

[287] P. Washington, C. Voss, N. Haber, S. Tanaka, J. Daniels, C. Feinstein, T. Winograd, and D. Wall. A wearable social interaction aid for children with autism. In *CHI*, 2016.

[288] R. Weber, V. Barrielle, C. Soladié, and R. Séguier. Unsupervised adaptation of a person-specific manifold of facial expressions. *T-AC*, 2018.

[289] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.

[290] D. H. Wolpert, W. G. Macready, et al. No free lunch theorems for optimization. *Transactions on evolutionary computation*, 1997.

[291] Y. Wu, E. Y. Chang, K. C.-C. Chang, and J. R. Smith. Optimal multimodal fusion for multimedia data analysis. In *ACMM*, 2004.

[292] Y. Wu and K. He. Group normalization. In *ECCV*, 2018.

[293] L. Xie and A. Yuille. Genetic cnn. In *ICCV*, 2017.

[294] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013.

[295] B. Xu, Y. Fu, Y.-G. Jiang, B. Li, and L. Sigal. Heterogeneous knowledge transfer in video emotion recognition, attribution and summarization. *T-AC*, 2016.

[296] D. Xu, W. Ouyang, X. Wang, and N. Sebe. Pad-net : multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, 2018.

[297] F. Yang, J. Wang, E. Shechtman, L. Bourdev, and D. Metaxas. Expression flow for 3d-aware face component transfer. *TOG*, 2011.

[298] S. Yang, Y. Zhang, D. Feng, M. Yang, C. Wang, J. Xiao, K. Long, S. Shan, and X. Chen. Lrw-1000 : A naturally-distributed large-scale benchmark for lip reading in the wild. In *FG*, 2019.

[299] X. Yang, P. Molchanov, and J. Kautz. Multilayer and multimodal fusion of deep neural networks for video classification. In *ACMM*, 2016.

[300] Y. Yang and T. Hospedales. Deep multi-task representation learning : A tensor factorisation approach. *ICLR*, 2016.

[301] Y. Yang and T. M. Hospedales. Trace norm regularised deep multi-task learning. *ICLR Workshop*, 2016.

[302] A. Yao, J. Shao, N. Ma, and Y. Chen. Capturing au-aware facial features and their latent relations for emotion recognition in the wild. In *ICMI*, 2015.

[303] L. Yao and J. Miller. Tiny imagenet classification with convolutional neural networks. *CS 231N*, 2015.

[304] G. Ye, D. Liu, I.-H. Jhuo, and S.-F. Chang. Robust late fusion with rank minimization. In *CVPR*, 2012.

[305] J. Ye, R. Janardan, and Q. Li. Two-dimensional linear discriminant analysis. In *NEURIPS*, 2005.

[306] W. Ying, Y. Zhang, J. Huang, and Q. Yang. Transfer learning via learning to transfer. In *ICML*, 2018.

[307] Z. Yu and C. Zhang. Image based static facial expression recognition with multiple deep network learning. In *ICMI*, 2015.

[308] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency. Tensor fusion network for multimodal sentiment analysis. In *EMNLP*, 2017.

[309] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy : Disentangling task transfer learning. In *CVPR*, 2018.

[310] M. D. Zeiler. Adadelta : an adaptive learning rate method. *arXiv preprint arXiv :1212.5701*, 2012.

[311] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[312] J. Zeng, S. Shan, and X. Chen. Facial expression recognition with inconsistently annotated datasets. In *ECCV*, 2018.

[313] K.-H. Zeng, T.-H. Chen, C.-Y. Chuang, Y.-H. Liao, J. C. Niebles, and M. Sun. Leveraging video descriptions to learn video question answering. In *AAAI*, 2017.

[314] X. Zeng, W. Ouyang, J. Yan, H. Li, T. Xiao, K. Wang, Y. Liu, Y. Zhou, B. Yang, Z. Wang, et al. Crafting gbd-net for object detection. *T-PAMI*, 2017.

[315] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *Signal Processing Letters*, 2016.

[316] Q. Zhang, Y. Nian Wu, and S.-C. Zhu. Interpretable convolutional neural networks. In *CVPR*, 2018.

[317] Q.-s. Zhang and S.-C. Zhu. Visual interpretability for deep learning : a survey. *Frontiers of Information Technology & Electronic Engineering*, 2018.

[318] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *ECCV*, 2016.

[319] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders : Unsupervised learning by cross-channel prediction. In *CVPR*, 2017.

[320] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, and P. Liu. A high-resolution spontaneous 3d dynamic facial expression database. In *FG*, 2013.

[321] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu. Deep mutual learning. In *CVPR*, 2018.

[322] Y. Zhang, R. Zhao, W. Dong, B.-G. Hu, and Q. Ji. Bilateral ordinal relevance multi-instance regression for facial action unit intensity estimation. In *CVPR*, 2018.

[323] Z. Zhang, J. M. Girard, Y. Wu, X. Zhang, P. Liu, U. Ciftci, S. Canavan, M. Reale, A. Horowitz, H. Yang, et al. Multimodal spontaneous emotion corpus for human behavior analysis. In *CVPR*, 2016.

[324] Z. Zhang, Y. Song, and H. Qi. Age progression/regression by conditional adversarial autoencoder. In *CVPR*, 2017.

[325] G. Zhao, X. Huang, M. Taini, S. Z. Li, and M. PietikäInen. Facial expression recognition from near-infrared videos. *Image and Vision Computing*, 2011.

[326] Z. Zhong, J. Yan, and C.-L. Liu. Practical network blocks design with q-learning. *AAAI*, 2017.

[327] X. Zhou and B. Bhanu. Feature fusion of side face and gait for video-based human identification. *Pattern Recognition*, 2008.

[328] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012.

[329] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. *ICLR*, 2016.

[330] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017.

[331] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. In *CVPR*, 2018.