Développement de modèles de reconnaissance des expressions faciales à base d'apprentissage profond pour les applications embarquées

Mohammadmahdi Deramgozin

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Développement de modèles de reconnaissance des expressions faciales à base d'apprentissage profond pour les applications embarquées

THÈSE

présentée et soutenue publiquement le 18 décembre 2023

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention Systèmes électroniques)

par

MohammadMahdi Deramgozin

Composition du jury

Président :	Pr. El-Bay BOURENNANE	Université de Bourgogne, Dijon
Rapporteurs :	Pr. Francois Xavier COUDOUX	Université polytechnique Hauts-De-France, Valenciennes
	Dr. Virginie FRESSE	Université Jean-Monnet, Saint Etienne
Encadrants :	Dr. Slavisa JOVANOVIC	Université de Lorraine, Nancy
	Pr. Hassan RABAH	Université de Lorraine, Nancy

Institut Jean Lamour (UMR7198)
Département Nanomatériaux, Électronique Et Vivant (N2EV)
Equipe 406 (MAE)

Remerciements

Il est courant de croire que la poursuite d'un doctorat est une entreprise solitaire. En réalité, la réussite d'une thèse est le résultat des efforts collaboratifs de la chercheuse ou du chercheur et de ses encadrants, ainsi que du soutien et des conseils de collègues, pairs et proches. Les contributions de ces individus, qui apportent non seulement une expertise technique, mais également un soutien émotionnel et des encouragements, sont inestimables et ne doivent pas être négligées. Il est important de reconnaître et de témoigner de la gratitude envers ces héros méconnus, car ils jouent un rôle crucial dans la réussite de la recherche et de la chercheuse ou du chercheur.

Permettez-moi de commencer par exprimer ma profonde reconnaissance envers les membres de l'équipe de Mesure électronique et Architecture de l'Institut Jean Lamour et l'école doctorale IAEM pour leur guide technique et administratif. Je suis profondément reconnaissant pour leur dévouement et leur professionnalisme.

Je tiens également à exprimer ma gratitude toute particulière à Miguel Arevalillo de l'Université de Valence pour son expertise technique et son engagement envers l'amélioration de la qualité de ma thèse. Sa contribution a été précieuse et a considérablement contribué à la réussite de mon travail de recherche. Je ne saurais assez remercier mes directeurs de thèse, le Professeur Hassan Rabah et le Dr Slavisa Jovanovic, pour leur dévouement et leur professionnalisme dans ce travail. Leur aide et leur soutien ont été précieux pour moi et je leur suis infiniment reconnaissant.

Enfin, je tiens à exprimer ma plus profonde gratitude à ma famille et mes amis, qui ont été là pour moi tout au long de mon parcours universitaire. Je suis particulièrement reconnaissant envers mon épouse, qui m'a soutenu inconditionnellement tout au long de ces années difficiles. Elle a été mon roc, mon pilier, mon soutien moral, et je ne saurais comment remercier suffisamment pour tout ce qu'elle a fait pour moi. Je l'aime de tout mon coeur. À mes parents, dont l'amour et le soutien ont été une force constante, merci. Cette thèse est aussi un reflet de votre enseignement et de votre confiance en moi.

Pour Sara, et Odin,
qui ont été mes piliers de force
et mes plus grandes motivations
tout au long de ma carrière
professionnelle et académique.

Table des matières

Introduction

1 Contexte

La reconnaissance des émotions humaines à travers la détection des expressions faciales (*Facial Emotion Recognition - FER*) joue un rôle primordial dans l'interaction homme-machine. En effet, en comprenant les sentiments et émotions humains, les machines peuvent mieux répondre aux besoins des utilisateurs, créant ainsi une expérience plus immersive et personnalisée. Cette notion trouve son importance dans une panoplie de domaines, notamment la santé [1], où la détection précoce des états émotionnels peut aider à identifier et traiter des conditions telles que la dépression ou l'anxiété [2]. Ou encore dans le secteur des services robotiques, une machine capable de reconnaître les émotions peut améliorer l'assistance fournie aux personnes âgées ou aux enfants, rendant l'interaction plus naturelle et empathique [3].

Avec l'avènement des techniques d'apprentissage profond, de nombreuses approches ont vu le jour pour améliorer la FER. Les réseaux neuronaux convolutifs (CNN), par exemple, ont montré une capacité exceptionnelle à capturer des caractéristiques faciales subtiles, permettant une classification émotionnelle plus précise [4]. Cependant, les émotions humaines étant intrinsèquement complexes et souvent ambigües, des méthodes supplémentaires, telles que l'attention neuronale, sont explorées pour mieux pondérer les régions clés du visage, comme les yeux ou la bouche, qui sont souvent cruciales pour déterminer l'émotion de la façon la plus précise [5].

De manière générale, la FER pour déterminer les émotions s'articule autour de deux approches majeures. La première approche, qualifiée de "globale", établit une corrélation entre une image faciale donnée et une expression spécifique sans détailler les nuances des mouvements individuels du visage. La seconde approche, dite "locale", se concentre sur l'analyse approfondie des mouvements complexes du visage associés à une émotion donnée.

Ces mouvements spécifiques du visage, connus sous le nom d'Unités d'Action (*Action Units - AU)*, constituent les éléments fondamentaux des expressions faciales. Par exemple, un sourire est caractérisé par l'activation simultanée de plusieurs AU, comme la levée des coins de la bouche et la contraction des muscles périorbitaux. Chaque émotion, qu'il s'agisse de joie, de tristesse ou de colère, est donc caractérisée par un ensemble distinct d'AU. Cette relation entre émotions et AU est codifiée par le Système de Codage de l'Action Faciale (*Facial Action Coding System - FACS*) [6, 7].

2 Objectifs

L'objectif principal de cette thèse est la conception et développement d'un modèle de reconnaissance faciale des émotions (FER) respectant les exigences suivantes :

- **Type de modèle** : le modèle FER doit s'insérer dans la continuité des approches issues de l'état de l'art et utiliser notamment les techniques de l'apprentissage profond (i.e. CNN) pour la détection précise des AU et des émotions correspondantes.
- **Performances** : Le modèle FER proposé aspirera à surpasser les approches existantes en matière de performances mesurées avec les métriques standard de l'apprentissage profond (la précision, l'*accuracy*, le F1-score, etc.)
- **Efficacité et faible complexité** : Le modèle FER proposé doit être efficient en termes de ressources utilisées, avec un nombre réduit de paramètres sans compromettre les performances globales. Ces propriétés le rendront particulièrement

adapté à l'utilisation dans des environnements à ressources limitées comme les appareils mobiles ou systèmes embarqués de manière générale d'une part, et offriront également les possibilités d'exécution en temps réel de par sa simplicité d'autre part.

— **Interprétabilité** : Le modèle FER proposé doit offrir non seulement des prédictions sur l'activation des AU pour une émotion donnée, mais également des arguments et justifications supplémentaires expliquant leur activation en se focalisant en particulier sur les zones faciales pertinentes pour l'émotion détectée. Ainsi, ces données supplémentaires contribueront à l'interprétation de la décision finale du modèle et rendront moins opaque le côté « boite noire » souvent rattaché à tous les modèles d'apprentissage profond.

La Figure 1 présente un aperçu visuel des différents étapes de l'approche utilisée dans ce travail de thèse : les jeux de données ou *datasets* à utiliser selon les différents critères (taille, diversité d'émotions et de sujets, annotations, variabilité de conditions, etc) ; les opérations de prétraitement (détection du visage, alignement, normalisation de l'éclairage, redimensionnement, etc) ; les étapes de conception et de développement de modèle CNN (choix de différentes couches, leur assemblage et paramétrage) ; l'interprétabilité et l'explicabilité des résultats obtenus (visualisation des résultats, cartes de chaleur, etc) ; et l'adaptation pour une implantation embarquée (faible complexité, restructuration, optimisation, implantation). Cette approche est utilisée tout au long de ces travaux de recherche pour d'une part proposer des modèles originaux respectant les objectifs de la thèse et pour les tester et valider sur des jeux de données divers et variés tout en vérifiant leur performance et pertinence dans la détection des émotions et des AU, d'autre part.

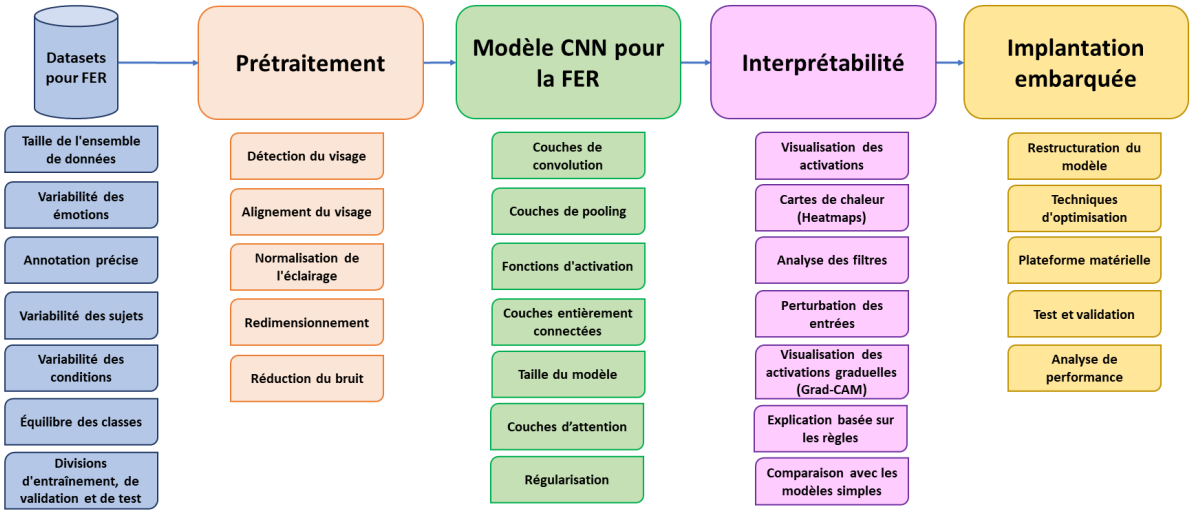


FIGURE 1 – Approche générale utilisée dans cette thèse pour la conception d'un système FER ou de reconnaissance d'émotions adapté à une implantation embarquée.

3 Organisation de la thèse

Cette thèse est organisée en 5 chapitres, chaque chapitre visant à introduire une partie du travail effectué et menant vers l'objectif final fixé étant un modèle FER adapté pour une implantation embarquée.

— **Chapitre 1** : Ce chapitre pose les fondations théoriques du travail, introduisant notamment les méthodes principales utilisées dans la reconnaissance des émotions ou systèmes FER, traditionnelles et les plus récentes basées sur l'apprentissage profond, ainsi que les étapes et les éléments principaux nécessaires pour la reconnaissance des émotions. Dans ce chapitre, les jeux de données couramment utilisés sont également introduits et discutés ainsi que le système d'encodage d'émotions, qui sera utilisé comme référence tout au long de ce travail. De surcroît, les étapes de prétraitement nécessaires pour les systèmes FER sont également introduites.

— **Chapitre 2** : Dans ce chapitre, la première contribution majeure de cette thèse est développée : la mise en œuvre d'un modèle CNN de faible complexité qui allie à la fois un nombre de paramètres relativement faible par rapport à l'état de l'art et les performances en termes de précision d'un bon niveau. Ce modèle sera validé

sur un ensemble de jeu de données et ses principaux avantages et inconvénients seront également discutés dans ce chapitre.

— **Chapitre 3** : Dans ce chapitre, la deuxième contribution majeure de cette thèse est présentée. Le chapitre présente une extension du modèle CNN présenté dans le chapitre 2 pour une détection précise des unités d'action faciales ou AU, un élément critique pour une FER réussie. Comme précédemment, ce nouveau modèle sera validé sur un ensemble de jeu de données et ses principaux avantages et inconvénients seront également discutés. Une comparaison avec les méthodes de référence de l'état de l'art sera également présentée et discutée.

— **Chapitre 4** : Dans le chapitre 4, la troisième contribution principale de cette thèse est présentée. Il s'agit de l'intégration des mécanismes d'attention dans les modèles CNN initiaux, présentés dans les chapitres 2 et 3. Ces mécanismes permettent d'augmenter la précision et l'interprétabilité d'un modèle d'apprentissage profond en se focalisant sur les régions du visage les plus pertinentes pour la FER et la détection des AU. Les étapes de validation et de test de ce modèle seront également présentées dans ce chapitre sur des jeux de données différents, ainsi que sa comparaison avec les approches les plus récentes de la littérature.

— **Chapitre 5** : Dans le chapitre 5, des étapes d'adaptation et d'optimisation du modèle le plus évolué et présenté dans le chapitre 4 en vue d'une implantation embarquée seront abordées. Les résultats présentés dans ce chapitre permettront de répondre à l'objectif final de cette thèse, d'arriver à créer un modèle efficace en termes de performances de détection des émotions et des unités d'action tout en ayant une frugalité en termes des ressources demandées (mémoire, de calcul), rendant la technologie plus accessible dans de nombreuses applications embarquées.

— **Conclusion et perspectives** : La dernière partie de cette thèse résume toutes les contributions présentées dans ce manuscrit et propose également des voies et perspectives de recherche futures.

-

Chapitre 1

Reconnaissance d'expressions faciales

Sommaire

1.1 Introduction

La reconnaissance des expressions faciales (FER) est un domaine en plein essor de la vision par ordinateur, qui vise à identifier les émotions et les intentions humaines à partir des expressions faciales. Les systèmes FER trouvent des applications dans divers domaines tels que la santé, la détection de l'état mental, la sécurité, la surveillance et la robotique [8]. Au cours des dernières décennies, le domaine de la FER a connu un essor considérable, marqué par des avancées technologiques et méthodologiques significatives [9]. Les approches utilisées pour aborder ce défi de reconnaissance des expressions faciales se sont progressivement diversifiées, allant des méthodes dites traditionnelles faisant appel à des outils divers et variés de traitement de signal et d'apprentissage machine (*machine learning*) aux techniques modernes basées sur l'apprentissage profond s'imposant de plus en plus ces dernières années [10].

Dans le chapitre présent, dans la première partie, nous examinerons les systèmes de reconnaissance des expressions faciales (les systèmes FER dans la suite de ce manuscrit), en mettant un accent particulier sur leurs caractéristiques et fonctionnalités principales. Dans la seconde partie de ce chapitre, nous présenterons également des différentes bases de données utilisées dans la littérature pour l'entraînement et la validation des systèmes FER, en soulignant leurs particularités distinctives et l'importance et l'influence de leur choix sur les performances du système FER.

1.2 Les systèmes FER

La reconnaissance des expressions faciales dans un système FER passe généralement par quatre étapes principales : la détection du visage, l'extraction des caractéristiques, la sélection des caractéristiques et la classification des émotions [11]. Ces étapes principales sont illustrées dans la figure 1.1. Avant de présenter les méthodes de l'état de l'art utilisées pour la reconnaissance des expressions faciales, il est important d'introduire la termino-

logie couramment utilisée dans la conception des systèmes FER ainsi que les expressions faciales que l'on cherche à détecter et étant à l'origine des différentes émotions. C'est pour cela que dans la suite de ce chapitre, nous allons tout d'abord introduire le système de référence pour le codage des expressions faciales et les relations entre ce système de codage avec les émotions les plus couramment recherchées. Ensuite, nous allons faire un tour des jeux de données trouvés dans la littérature scientifique et adaptés pour la conception, test et validation des systèmes FER.

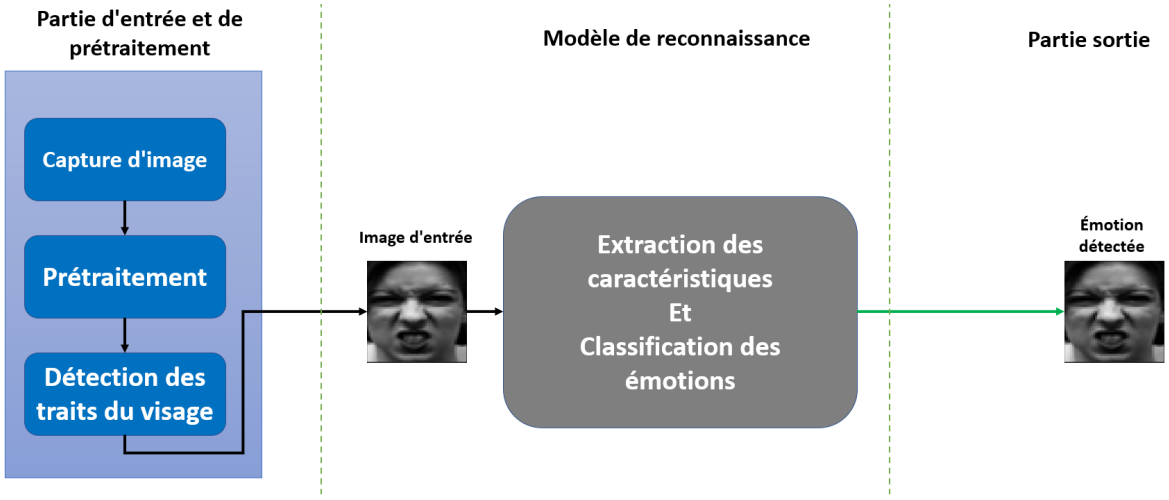


FIGURE 1.1 – Schéma général d'un système de reconnaissance des expressions faciales FER. Ce schéma illustre les principales étapes d'un système FER, comprenant la capture d'image d'entrée, les étapes de prétraitement, l'extraction des caractéristiques ou *features* et l'étape finale de classification des émotions.

1.2.1 FACS : Système de codage des expressions

Le système FACS (*Facial Action Coding System*) est une méthode développée par Ekman et Friesen pour décrire objectivement les mouvements du visage à l'aide des indicateurs nommés unités d'action (*AU - Action Unit*) [12]. Le système FACS est basé sur l'observation des muscles faciaux et des mouvements qu'ils produisent, permettant ainsi d'identifier les expressions faciales et de les coder de manière systématique [13].

Fondement anatomique Le système FACS est basé sur l'anatomie du visage et sur la compréhension des contractions musculaires qui produisent des changements visibles à la surface de la peau. Il identifie 44 AU et décrit la combinaison de ces AU permettant de représenter différentes expressions faciales [13]. Le tableau 1.1 présente des exemples de codage de certaines émotions selon le système FACS de Ekman et Friesen.

Analyse et codage Pour analyser et coder les mouvements du visage selon le système FACS, on utilise généralement les étapes suivantes [13] :

1. Identifier les AU présentes dans une expression faciale.
2. Mesurer l'intensité de chaque AU à l'aide d'une échelle de cinq points (A à E).
3. Noter la symétrie de chaque AU (symétrique ou asymétrique) par rapport au visage.
4. Enregistrer l'évolution temporelle de chaque AU, y compris l'apparition, l'apogée, et la disparition.

Représentation mathématique Dans le domaine de la reconnaissance des expressions faciales, les AU sont souvent quantifiées et peuvent être représentées en utilisant des vecteurs d'attributs. Formellement, pour une expression faciale donnée `v`, ce vecteur peut être exprimé comme suit [14] :

$$v = (AU_1, AU_2, \ldots, AU_n) \tag{1.1}$$

où AU_i représente l'intensité de l'AU i, codée sur une échelle allant de A (l'intensité la plus faible) à E (l'intensité la plus élevée).

À titre d'exemple, considérons une expression de joie. Selon Ekman, cette expression est principalement caractérisée par deux AU, AU_6 et AU_{12}, ayant des intensités les plus élevées [12] :

$$v_{\text{joie}} = (A, A, A, A, A, E, A, A, A, A, A, E, A) \tag{1.2}$$

TABLE 1.1 – Le tableau du système FACS pour les états émotionnels les plus communs

Émotion	AU actives
Colère	4+5+7+23
Dégoût	9+15+17
Peur	1+2+4+5+7+20+26
Joie	6+12
Tristesse	1+4+15
Surprise	1+2+5+26

Dans cette représentation spécifique, AU_6 et AU_{12} sont marquées avec une intensité 'E', signifiant qu'elles sont les unités d'action activées avec la plus grande intensité dans l'expression de joie.

Application dans les systèmes FER La méthode d'encodage selon le système FACS peut être utilisée comme base dans les systèmes FER, où les vecteurs caractéristiques des expressions observées dans une image sont comparés avec les vecteurs caractéristiques des expressions de référence (selon Ekman). Les mesures de distance, telles que la distance euclidienne, peuvent être utilisées pour évaluer la similarité entre les expressions observées et de référence [14] :

$$d(v_1, v_2) = \sqrt{\sum_{i=1}^{44} (AU_{v_{1_i}} - AU_{v_{2_i}})^2} \tag{1.3}$$

où v_1 et v_2 sont les vecteurs d'attributs des expressions faciales à comparer, $d(v_1, v_2)$ est la distance euclidienne entre ces deux vecteurs, et $AU_{v_{1_i}}$ et $AU_{v_{2_i}}$ représentent les AU des deux vecteurs v_1 et v_2 respectivement.

Pertinance Le système FACS offre une approche structurée et anatomiquement fondée pour coder et analyser les expressions faciales. Il permet une interprétation précise et standardisée des mouvements du visage, rendant possible une reconnaissance fiable des émotions dans diverses applications. Par conséquent, l'intégration du FACS dans les systèmes de FER fournit un cadre robuste pour la comparaison et l'analyse des expressions

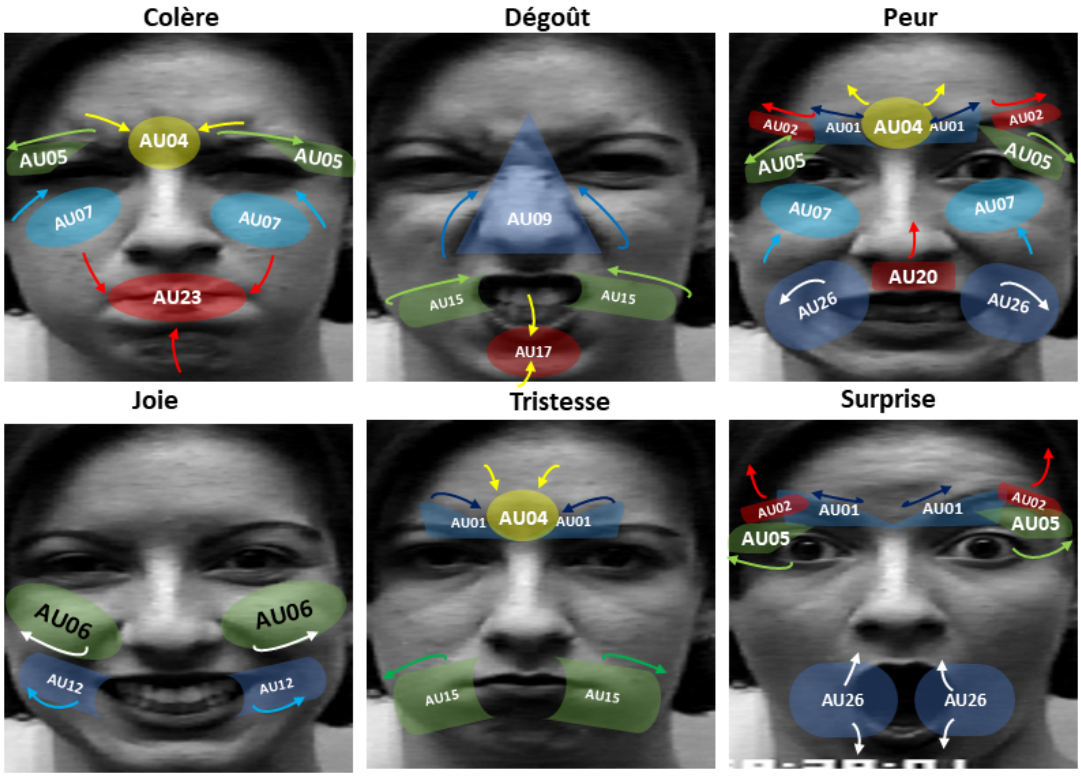


FIGURE 1.2 – Un exemple d'estimation de positionnement des AU selon la méthode FACS de Ekman appliquée sur les images représentant différents états émotionnels et se trouvant dans le jeu de données CK+ [15].

faciales, et est de ce fait primordiale dans le domaine de la reconnaissance des expressions faciales.

1.2.2 Repères faciaux et détection d'unités d'action

Repères faciaux Les repères faciaux ou *landmarks*, sont des points spécifiques sur le visage qui représentent des caractéristiques anatomiques ou géométriques distincts, tels que les coins des yeux, les bords de la bouche et les points représentant le contour du nez [16]. Ces repères sont souvent utilisés pour décrire et analyser les expressions faciales en termes de changements de position, de forme et d'apparence des traits du visage.

Méthodes de suivi des repères faciaux Le suivi des repères faciaux peut être effectué par diverses méthodes :

— **Identification de descripteurs locaux** : Identification des points clés sur un visage en utilisant des descripteurs locaux s'avère être une technique constructive dans le suivi des repères faciaux. Cette technique implique l'isolement et l'identification des points saillants du visage au moyen de descripteurs locaux qui, dans ce contexte, sont définis comme des vecteurs caractéristiques représentant des régions spécifiques du visage [17].

— **Gauchissement du modèle** : Le gauchissement du modèle se réfère à la procédure d'alignement de l'image d'entrée avec un modèle préalablement établi du visage [18]. Dans ce processus, une image faciale d'entrée est méticuleusement alignée avec un modèle de visage préétabli pour garantir une superposition précise des traits faciaux.

— **Détection précise** : Utilisation de réseaux de neurones profonds pour une détection précise des points caractéristiques du visage [19].

— **Suivi des repères faciaux** : Utilisation de diverses techniques, notamment d'apprentissage profond pour le suivi des repères faciaux [20].

Calcul des caractéristiques géométriques et d'apparence : Les points de repère détectés peuvent être utilisés pour calculer les caractéristiques géométriques et d'apparence qui décrivent les mouvements et les changements d'apparence des traits du visage, tels que la distance entre les yeux, la largeur de la bouche, et l'intensité des expressions faciales et des rides [21]. Par exemple, la distance euclidienne entre deux repères faciaux (x_1, y_1) et (x_2, y_2) est donnée par :

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{1.4}$$

Détection des unités d'action à partir de repères faciaux : L'utilisation des repères faciaux pour détecter les AU est basée sur l'hypothèse que les AU sont représentées par des changements spécifiques et mesurables dans les caractéristiques géométriques et

d'apparence des repères faciaux [22]. Les algorithmes de détection d'AU basés sur des repères faciaux cherchent à établir des relations entre les changements observés dans les caractéristiques des repères faciaux et les AU correspondantes. Ces relations peuvent être apprises et modélisées à l'aide de techniques de régression, de classification ou d'apprentissage profond [23, 24].

Étapes principales dans la détection des AU basées sur l'analyse des repères faciaux : Un algorithme de détection des AU basé sur des repères faciaux pourrait suivre les étapes suivantes [22] :

1. Repérer et extraire les points de repère faciaux à partir d'une image ou d'une séquence d'images faciales.
2. Calculer les caractéristiques géométriques et d'apparence des points de repère faciaux, telles que les distances, les angles et les intensités des points de repère faciaux.
3. Utiliser un modèle appris pour prédire les AU présentes dans l'expression faciale à partir des caractéristiques calculées.

Connaissance des repères faciaux et leurs applications : L'identification et le suivi des repères faciaux jouent un rôle critique dans de nombreux domaines spécialisés. Par exemple, en reconnaissance des expressions faciales (FER), une localisation précise de ces repères est indispensable pour extraire efficacement les AU, qui sont des indicateurs clés de l'état émotionnel [14, 12]. Dans le domaine de la réalité augmentée, les repères faciaux permettent de superposer de manière réaliste des éléments virtuels sur un visage humain [25]. En biométrie, la détection des repères faciaux contribue à la conception de systèmes d'identification plus sécurisés en permettant une comparaison géométrique plus fine des traits faciaux [26]. En réhabilitation médicale, la surveillance de ces repères peut être utilisée pour suivre les progrès en rééducation faciale après un traumatisme [27]. Ainsi,

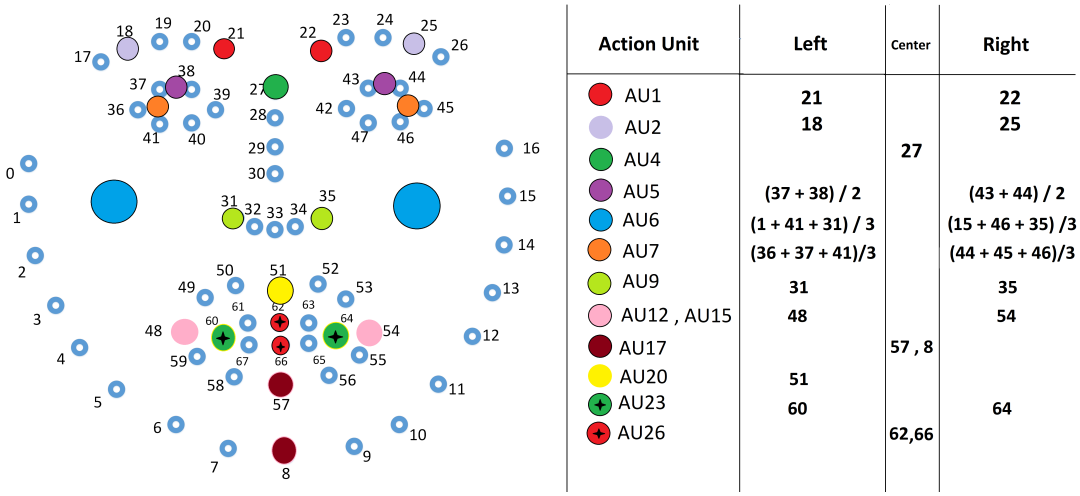


FIGURE 1.3 – Illustration des points de repère faciaux et des emplacements estimés des 13 AU courantes sur une image du visage. Les points de repère sont indiqués par des points colorés et les AU sont représentées par des flèches et des cercles, montrant la direction et la zone d'influence de chaque AU sur le visage.

une détection précise et robuste des repères faciaux est un prérequis dans la recherche de nouvelles méthodes et techniques plus efficaces et fiables favorisant ainsi l'innovation et le progrès dans ces domaines spécifiques [28].

1.3 Bases de données pour les systèmes FER

Les bases de données pertinentes sont fondamentales dans l'entraînement et l'évaluation des systèmes FER, fournissant d'une part une collection d'images faciales correspondant à différents états émotionnels et d'autre part des annotations précises des expressions faciales, souvent analysées et validées par les experts. Ces bases de données sont donc indispensables pour la conception, l'entraînement et la validation des modèles FER dont le rôle est à discerner et interpréter correctement les émotions et les AU à partir des repères faciaux [29].

Organisation des bases de données pour la FER

Une base de données typique pour la FER peut inclure une variété d'images : en niveaux de gris, en couleur, et parfois des images thermiques, facilitant l'analyse approfondie des expressions faciales [30]. Ces bases de données se répartissent en deux catégories principales : « in-the-lab » qui contiennent des images prises dans un environnement contrôlé, et « in-the-wild » où les images proviennent de sources variées, offrant une diversité naturelle et non scénarisée. Les éléments principaux d'une base de données dans le domaine de reconnaissance des expressions faciales sont les suivants :

- **Résolution des images** Les résolutions dans ces bases de données peuvent varier considérablement, permettant ainsi des analyses de l'impact de la qualité de l'image sur la précision d'un système FER.
- **Diversité d'états émotionnels** Les bases de données contiennent généralement des exemples diversifiés d'émotions et AU pour tester la robustesse des modèles FER dans différents scénarios [31].
- **Étiquetage** Les images sont souvent accompagnées d'étiquettes (*labels*) spécifiant l'émotion exprimée ou les AU activées, favorisant ainsi une classification précise et une analyse détaillée de l'image faciale observée.
- **Diversité de sujets** Les bases de données peuvent inclure des méta-informations pour permettre une analyse plus diversifiée et plus approfondie sur d'autres aspects annexes tels que la variété ethnique, l'âge, le genre, les conditions d'éclairage, et les orientations de visage [32].
- **Fiabilité** La fiabilité des annotations, assurée par des experts ou des annotateurs formés, est cruciale pour avoir la confiance totale dans les données présentées dans une base et joue un facteur prépondérant dans l'évaluation des performances des modèles FER proposés et leur pertinence finale [33].

1.3.1 Principales bases de données pour la FER

La sélection des bases de données appropriées est cruciale pour la reconnaissance des expressions faciales. Les bases de données listées dans le tableau 1.2 sont les bases les plus communément utilisées dans la recherche dans le domaine de la FER. On peut noter une diversité de paramètres dans les éléments principaux listés précédemment au niveau de : type de données d'entrée (vidéo ou images) ; différentes résolutions d'images d'entrée allant de 48×48 jusqu'à 1280×720 ; nombre total d'échantillons présents ; nombre différent d'états émotionnels traités ; la présence ou non des unités d'action dans les annotations ; conditions d'acquisition des données présentes. Nous pouvons également remarquer qu'une grande majorité des bases de données présentées sont relativement récentes et que cela montre l'importance de la qualité des données représentées dans une base dans la conception de modèles FER fiables et robustes et leur impact sur une utilisation dans les scénarios réels caractérisés par une diversité d'information acquises dans les conditions très variables et non-contrôlées.

TABLE 1.2 – Les caractéristiques principales des bases de données communément utilisées pour la FER et la détection d'AU.

Nom	Média	Résolution	Nombre d'échantillons	Labels	Diversité	Type de base
CK+ [15]	Image	640x480	593	7 émotions	Faible	In-the-lab
BP4D [34]	Vidéo	640x480	140	5 AU	Faible	In-the-lab
DISFA [35]	Vidéo	960x720	27	12 AU	Faible	In-the-lab
FER2013 [36]	Image	48x48	35,887	7 émotions	Élevé	In-the-wild
AffectNet [37]	Image	Multiples	1,100,000	11 émotions	Élevé	In-the-wild
EmoReact [38]	Vidéo	1280x720	40	6 émotions	Élevé	In-the-wild
ExpW [39]	Image	Multiples	91,793	8 émotions	Élevé	In-the-wild
RAF-DB [40]	Image	Multiples	29,672	7 émotions	Élevé	In-the-wild
JAFFE [41]	Image	256x256	213	7 émotions	Faible	In-the-lab
SEMAINE [42]	Vidéo	640x480	232	4 émotions	Faible	In-the-lab

1.3.2 Défis et considérations

Bien que les bases de données dans le domaine de la FER soient riches et diversifiées, elles présentent néanmoins plusieurs défis qui peuvent affecter les performances des systèmes FER. Parmi ces défis, les biais liés à la composition des bases de données sont particulièrement notables. Ces biais peuvent résulter d'une représentation insuffisante de divers groupes en fonction de l'âge, du sexe, de l'ethnicité et d'autres facteurs sociodémographiques. Ces limitations peuvent entraîner une performance inégale des systèmes FER lorsqu'ils sont appliqués à différentes populations [43]. De plus, la variabilité des expressions faciales peut ne pas refléter fidèlement les scénarios réels, impactant la généralisation des systèmes dans le monde réel [44]. Des recherches supplémentaires sont nécessaires pour surmonter ces défis et créer des bases de données plus robustes et représentatives pour les systèmes FER.

1.4 Prétraitement des images pour les systèmes FER

Le prétraitement des images faciales joue un rôle essentiel dans les systèmes FER. Cette phase est cruciale pour préparer les images pour une extraction efficace des caractéristiques et une meilleure performance dans la reconnaissance des expressions faciales [45]. Les principales étapes de prétraitement dans le domaine de la reconnaissance des expressions faciles sont les suivantes :

— **Détection du visage** La première étape du prétraitement consiste à détecter et à isoler le visage dans une image ou une vidéo. Des techniques telles que l'algorithme de Viola-Jones ou les CNN peuvent être utilisées pour cette tâche [46].

— **Normalisation de l'éclairage** Les variations de l'éclairage peuvent affecter les performances des systèmes FER. Des techniques de normalisation, comme l'égalisation d'histogramme ou la normalisation gamma, sont donc souvent utilisées pour minimiser l'impact de l'éclairage sur les images faciales [47].

— **Alignement du visage** Cette étape vise à aligner les visages pour minimiser les variations dues à la position du visage. Les méthodes d'alignement comprennent la rotation, la réduction d'échelle, et la translation de l'image faciale [48].

— **Détection des points caractéristiques du visage** Pour des systèmes FER plus sophistiqués, des points caractéristiques du visage tels que les yeux, le nez, la bouche, et les contours du visage sont détectés. Ces points sont ensuite utilisés pour extraire les caractéristiques locales du visage qui sont informatives pour la reconnaissance des expressions faciales [49].

— **Normalisation géométrique** Enfin, les images sont souvent normalisées pour avoir une taille fixe avant d'être introduites dans le modèle FER. Cette normalisation permet de réduire les variations de taille et de forme des visages dans les images [50].

1.5 Méthodes utilisées pour la FER

Dans le domaine de la reconnaissance faciale (FER), une multitude d'approches différentes ont été explorées jusqu'à présent pour décrypter notamment les complexités des expressions faciales, ayant des applications dans des domaines variés tels que la robotique, le marketing, la santé et plus encore. Les méthodes classiques ont permis de poser les bases en utilisant diverses techniques d'extraction de caractéristiques et de classification, tandis que les modèles de deep learning utilisés de manière croissante ces dernières années ont apporté une nouvelle dimension avec leur capacité à traiter directement les images brutes et à apprendre des représentations hiérarchiques des données [51]. Cette section explore en détail ces méthodes, mettant en avant leur évolution, leurs succès et les défis restant à relever dans le domaine de la FER.

1.5.1 Méthodes classiques

Les méthodes classiques de FER, dominant le domaine avant la montée du deep learning, se basent essentiellement sur l'extraction de caractéristiques faciales via deux approches principales : les méthodes géométriques et celles basées sur l'apparence [52, 53]. Les premières se focalisent sur la forme et la position des éléments faciaux, tandis que les secondes analysent la texture du visage [54]. Des techniques comme *Support Vector Machine (SVM)*, *Dynamic Bayesian Network* et *Fuzzy Logic* ont été largement adoptées dans des études précédentes, démontrant une certaine efficacité dans divers scénarios de FER [55, 56, 57, 58, 59, 60]. D'autres méthodes, comme LBP *(Local Binary Pattern)*, LDP *(Local Directional Pattern)*, FFT+CLAHE *(Fast Fourier Transform + Contrast Limited Adaptive Histogram Equalization)* et HOG *(Histogram of Oriented Gradients)*, ont également été explorées, chacune présentant ses propres avantages et défis [61, 62, 63]. Néanmoins, ces méthodes classiques sont souvent confrontées à des difficultés, notamment dans la gestion des variations d'éclairage et des expressions faciales naturelles.

1.5.2 Modèles de deep learning

L'introduction des réseaux neuronaux profonds a marqué un tournant décisif dans le domaine de la FER, avec des architectures telles que les CNN qui ont montré une capacité remarquable à apprendre directement à partir d'images, minimisant ainsi le besoin d'une extraction manuelle de caractéristiques [64]. Les architectures innovantes, comme VGG-Face et ResNet, ont redéfini les normes de performance en élevant de manière significative les niveaux de précision dans le domaine de la FER [65]. En parallèle, d'autres structures de réseaux neuronaux, y compris les perceptrons multicouches (MLP) et les réseaux de neurones généraux, ont été exploitées, élargissant ainsi l'éventail des méthodologies appliquées dans ce domaine [66, 67]. Les mécanismes d'attention, intégrés récemment dans les architectures de réseaux neuronaux, ont introduit une nouvelle dimension dans le domaine de la FER, en optimisant davantage la précision globale et la compréhension des

modèles et en focalisant l'apprentissage uniquement sur les aspects les plus importants des données d'entrée utilisées [68, 69]. Il convient toutefois de mentionner que, bien que puissantes, ces méthodes avancées ne sont pas exemptes de défis, tels que la sensibilité aux variations de pose et d'éclairage, qui demeurent toujours des obstacles majeures à la précision des modèles de FER [70].

Au cours des années, diverses méthodes, qu'il s'agisse de modèles classiques ou de deep learning, ont été développées et évaluées pour améliorer la précision et l'efficacité des systèmes de FER. Pour mettre en perspective ces approches méthodologiques variées, le tableau 1.3 présente un résume des approches les plus significatives de le domaine de la FER en mettant en lumière les techniques employées, les bases de données utilisées pour l'entraînement et l'évaluation, ainsi que la performance respective de chaque méthode. En explorant ce tableau, il est possible de discerner les tendances, les réussites et les défis rencontrés par différentes méthodes au fil du temps, offrant ainsi une vue d'ensemble des voies de recherche passées, présentes et potentiellement futures dans le domaine de la FER.

1.6 Conclusion

En résumé, dans ce chapitre le système FACS a été abordé comme un moyen clé pour retrouver les expressions faciales à partir des AU. L'importance des repères ou *landmarks* faciaux dans le repérage précis des AU a également été discutée. De plus, nous avons expliqué les critères pour le choix des bases de données, en soulignant l'importance de la diversité et de l'étiquetage précis des AU et des expressions. De surcroît, ce chapitre a fourni également une revue des méthodes couramment utilisées dans le domaine de la FER. Comme résultat de cette analyse, il apparait clairement que les méthodes dominantes actuellement dans le domaine de la reconnaissance faciale des émotions sont basées majoritairement sur l'utilisation de réseaux de neurones convolutifs CNN.

TABLE 1.3 – Évaluation des systèmes FER proposés au cours de la dernière décennie basés sur les modèles classiques et de deep learning.

Classifieur	Référence/Année	Méthode	Jeu de données
SVM	[71] / 2014	HOG, dense SIFT features, SVM kernel, logistic regression, partial least squares	AFEW 4.0, (in-the-wild)
	[32] / 2014	Fonctionnalités STM, UMM, SVM linéaire multiclasse	AFEW 4.0, (in-the-wild)
KNN	[63] / 2018	FFT-CLAHE, fonctionnalités MBPC, KNN, MLP, logistique simple, SMO, J48	SWEF, (in-the-wild)
	[72] / 2017	Domaine logarithme-laplace (LL), en DLBP, théorème d'expansion de Taylor	CK+ (in-the-lab)
Réseau profond	[73] / 2015	Fonctionnalités LBP, réseau neuronal convolutif (CNN)	SWEF, (in-the-wild)
	[62] / 2017	Caractéristiques MLDP-GDA, Deep Belief Network (DBN)	40 vidéos réalisées, auprès de 10 sujets, (in-the-lab)
	[74] / 2018	Réseau neuronal convolutif spatio-temporel	MMI, CK+, Florentin (in-the-wild)

Le chapitre suivant se concentrera sur l'utilisation des réseaux CNN pour la reconnaissance des expressions faciales, notamment au niveau de la détection d'émotions spécifiques. L'objectif principal, qui sera maintenu tout au long de cette thèse, est de proposer des modèles d'apprentissage profond, alliant à la fois les performances en termes de précision de détection et la complexité faible permettant leur utilisation plus aisée dans les systèmes embarqués à ressources limitées.

Chapitre 2

Modèle CNN pour la FER

Sommaire

2.1 Introduction

La FER est un domaine de recherche en rapide développement qui a pour objectif de créer des systèmes aptes à identifier et à classifier les émotions humaines à partir de mouvements faciaux. Les réseaux de neurones convolutifs CNN se sont imposés ces dernières années comme des modèles performants dans ce domaine en raison de leur capacité à apprendre des représentations complexes et inhérentes dans les données liées à la reconnaissance faciale se trouvant dans de nombreuses bases de données.

Cependant, il existe plusieurs problèmes inhérents aux solutions existantes. Un des défis majeurs est le compromis entre la précision des modèles proposés et leur complexité computationnelle. De nombreux modèles de référence, bien qu'efficaces, nécessitent des ressources de calcul considérables et sont donc moins adaptés pour des applications embarquées ou des environnements avec des contraintes de ressources limitées.

Dans ce contexte, le premier objectif de ce chapitre est de proposer un modèle CNN léger pour la reconnaissance des expressions faciales. De surcroît, le modèle que nous proposons vise à offrir un équilibre optimal entre la précision et l'efficacité computationnelle, permettant son déploiement dans des applications en temps réel ou des dispositifs à ressources limitées.

Dans ce chapitre, nous allons tout d'abord introduire le concept général des réseaux de neurones convolutifs CNN avant de présenter les approches à base de réseaux CNN utilisés dans le domaine de la reconnaissance des expressions faciales, avec leurs principales caractéristiques. Ensuite, dans la seconde partie de ce chapitre nous allons introduire le modèle CNN que nous proposons répondant aux exigences fixées au préalable, notamment un compromis entre la précision et la complexité computationnelle influençant directement son empreinte mémoire et son déploiement potentiel dans les systèmes embarqués à ressources limitées.

2.2 Réseaux neuronaux convolutifs (CNN)

Les réseaux de neurones convolutifs CNN représentent une classe de réseaux de neurones artificiels largement utilisés pour la reconnaissance d'images, y compris la détection et la reconnaissance des expressions faciales, comme nous l'avons abordé précédemment [75]. Ils ont été introduits pour la première fois dans les années 1980 et ont gagné en popularité ces dernières années en raison d'améliorations significatives des supports de calcul utilisés pour leur exécution d'une part, et de la disponibilité d'un grand nombre de bases ou d'ensembles de données, d'autre part [76].

2.2.1 Architecture CNN

Les CNN sont basés sur un ensemble de neurones et de connexions neuronales organisées en différentes couches comme illustré à la figure 2.1. Les couches de convolution, dont le rôle principal est l'extraction des caractéristiques ou des *features* d'une image, sont souvent associées à d'autres couches, comme les couches de regroupement nommées *maxpooling*, les couches d'activation apportant une non-linéarité dans la chaîne ainsi que les couches entièrement connectées (*fully connected*), pour améliorer les performances de classification [77]. Ces architectures complexes sont particulièrement adaptées à des applications différentes dans le domaine de traitement d'images et vidéo ou de la vision par ordinateur de manière générale telles que la reconnaissance des expressions faciales. La description détaillée des différents éléments d'un réseau CNN est donnée en annexe A.

2.2.2 Systèmes FER à base de réseaux CNN

Les systèmes FER ont bénéficié d'une transformation révolutionnaire grâce à l'application et utilisation des CNN. Étant largement utilisés dans divers domaines de la vision par ordinateur, les CNN se sont révélés particulièrement utiles pour les systèmes de FER et la détection des AU [78, 79, 80]. Leur capacité à apprendre des traits de visage com-

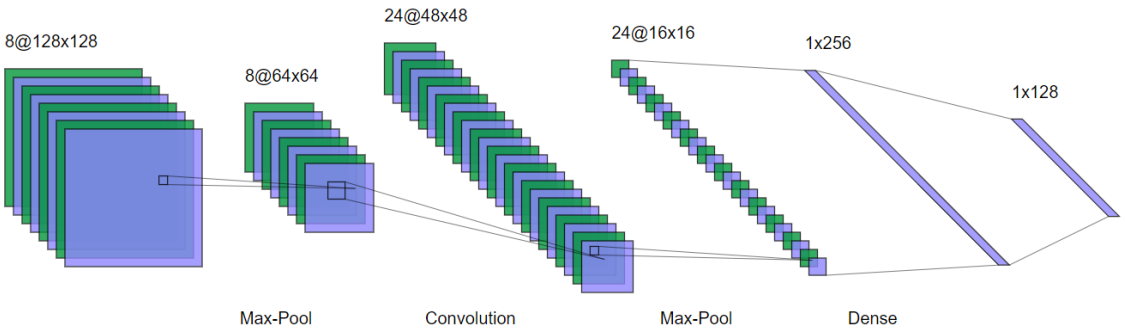


FIGURE 2.1 – Schéma d'un modèle de réseau neuronal convolutif (CNN) comprenant des couches de convolution et de *maxpooling*. Les couches de convolution servent à extraire des caractéristiques spatiales à partir de l'image d'entrée, tandis que les couches de *maxpooling* réduisent la dimensionnalité, permettant au réseau d'apprendre des caractéristiques plus abstraites et robustes.

plexes et subtils a rendu les architectures complexes et souvent nommées *backbones* telles que VGG, ResNet et Inception très populaires pour la reconnaissance des émotions [81]. Les avancées récentes ont vu l'adoption de techniques innovantes de CNN dans la FER, telles que l'utilisation de l'apprentissage par transfert [82], l'intégration d'attention spatiale [83], et la mise en œuvre de méthodes multi-modales [84]. En outre, ces architectures sont également employées pour détecter et classer les AU à partir d'images faciales, en capturant les nuances des mouvements musculaires, et sont formées sur des ensembles de données tels que DISFA [35]. Ces applications diverses aident à analyser et à interpréter les expressions faciales humaines à des diverses fins, comme la surveillance de l'humeur, l'interaction homme-ordinateur et les études psychologiques, témoignant de la polyvalence et de l'efficacité des CNN dans le domaine de la FER [85, 86, 87, 88, 89, 90].

2.2.3 Motivation pour un modèle CNN spécifique à la FER

Plusieurs raisons justifient la création d'un nouveau modèle ou d'une nouvelle architecture à base de CNN pour la FER :

— **Complexité des modèles existants** : Les architectures de CNN existantes comme VGG, ResNet et Inception sont souvent conçues pour des tâches de vision

par ordinateur plus générales et peuvent être trop complexes pour des applications spécifiques de FER, entraînant une utilisation inefficace des ressources [91].

— **Applicabilité limitée** : Bien que des méthodes comme l'apprentissage par transfert puissent adapter ces modèles pré-entraînés aux tâches de FER, l'efficacité de ces fonctionnalités apprises dans des contextes variés n'est pas garantie [92].

— **Adaptabilité et flexibilité** : La conception d'un CNN propre au domaine de la FER permet une adaptation plus précise aux besoins spécifiques de la reconnaissance des émotions faciales, y compris l'intégration de mécanismes d'attention ou de couches spécialisées [93].

Ainsi, proposer et développer un modèle à base de CNN propre à la FER offre une flexibilité indispensable pour traiter et aborder les défis spécifiques du domaine d'une part, et permet également une contribution originale et taillée sur mesure, d'autre part.

2.2.4 Complexité des modèles CNN dans les systèmes FER

La taille des modèles de réseaux de neurones CNN utilisés dans les systèmes FER a un impact significatif sur leurs performances et l'efficacité. Les modèles avec un grand nombre de paramètres peuvent fournir une meilleure précision et des performances plus robustes, mais sont généralement gourmands en calculs et en mémoire [94]. À l'inverse, les modèles légers sont conçus pour être plus efficaces en termes de l'empreinte mémoire et de ressources de calcul nécessaires, tout en conservant une précision raisonnable. Dans cette section, nous discuterons des avantages et des inconvénients des modèles CNN de grande taille et des modèles légers et de petite taille utilisés dans les systèmes FER, en présentant notamment des travaux de recherche issus de la littérature scientifique récente.

Modèles CNN de grande taille pour la FER

Les avancées récentes dans la technologie des semiconducteurs, introduisant les circuits de microprocesseurs CPU à haute vitesse et les processeurs graphiques GPU puissants

dans la vie quotidienne, ont rendu plus facile l'utilisation de réseaux de neurones CNN ou de l'apprentissage profond de manière générale dans divers domaines, y compris les systèmes de FER. L'utilisation de modèles de grande taille, qui intègrent des architectures de réseaux de neurones convolutifs CNN telles que ResNet [95] et VGG [96] comme support de base (*backbone*), dans les systèmes de FER a conduit à des améliorations significatives des performances de tels systèmes, notamment en termes de l'exactitude et des scores F1 des résultats obtenus (voir la section 2.5.3 où les metriques courantes utilisées dans les systèmes FER sont introduites). Cependant, ces systèmes possèdent souvent un grand nombre de paramètres et sont très gourmands en ressources de calcul demandées pour leur exécution. A titre d'exemple, Shao et al. ont proposé une méthode utilisant l'apprentissage basé sur l'attention spatiale [97] pour améliorer le taux de détection des AU dans des images de taille $200 \times 200 \times 3$ et l'architecture VGGNet avec plus de 138 millions de paramètres. De même, Zhang et al. dans [98] ont défini un système de FER basé sur le modèle HRNetV2-W18 pour étendre les cartes de chaleur aux cartes de régions d'intérêt en utilisant un modèle de graphe convolutif, qui a également abouti à une architecture de 138 millions de paramètres. De plus, l'utilisation de graphes CNN est proposée dans [99] dans le cadre d'apprentissage de la relation spatio-temporelle et de l'attention pour la détection des AU avec plus de 26 millions de paramètres et une couche d'entrée de $200 \times 200 \times 3$. Par conséquent, on peut conclure que l'incorporation d'architectures CNN connues et déjà validées dans les différents domaines de la vision par ordinateur comme support de base dans les systèmes de FER se traduit souvent par une augmentation du nombre total de paramètres dans ces modèles. Le tableau 2.1 présente un récapitulatif des systèmes FER issus de l'état de l'art récent basés sur l'utilisation de modèles CNN de grande taille.

Modèles CNN de petite taille pour la FER

Les systèmes de reconnaissance des expressions faciales (FER) exploitent de plus en plus les modèles légers, en raison de leur consommation moindre en termes de mémoire

TABLE 2.1 – Comparaison de la taille des modèles profonds utilisés dans les systèmes FER. Ce tableau met en évidence les principales spécifications de divers modèles FER, y compris leurs architectures de réseaux de neurones, les tailles d'image d'entrée et le nombre de paramètres entraînables (en millions).

Modèle	**Architecture**	**Taille d'entrée**	**Paramètres (Millions)**
[97]	VGGNet	200×200	138
[98]	HRNetV2-W18	256×192 384×288	138
[100]	CNN Amélioré	260×260	27
[99]	Apprentissage des Relations Spatio-Temporelles Attentionnelles	200×200	26
[101]	Module de convolution de groupe avec cartes de saillance différentielles	256×256	16
[102]	A-MobileNet	224×224	3.4
[103]	Simple CNN	48×48	2.5

et de ressources de calcul. Bien que les modèles FER traditionnels, basés sur des architectures CNN, soient généralement riches en paramètres et gourmands en ressources, leur application est restreinte dans des contextes tels que les dispositifs mobiles et les systèmes embarqués, où les ressources de calcul sont limitées.

Les modèles légers se réfèrent à des architectures de réseaux de neurones présentant un nombre réduit de paramètres et une complexité de calcul moindre par rapport aux modèles plus conséquents. Bien que certains modèles légers, tels que MobileNet [104], ShuffleNet [105], et SqueezeNet [106], ne soient pas spécifiquement conçus pour la FER, leur nature économe en ressources sert d'inspiration pour le développement de modèles FER légers. Toutefois, il est essentiel de noter que l'utilisation de modèles légers peut entraîner une diminution des performances, notamment en termes de précision, et nécessite par conséquent une conception attentive pour maintenir un équilibre entre ressources consommées et performances. Des modèles exemplaires sont explicités ci-après.

Octave-CNN : Le modèle Octave-CNN, proposé dans [103], est un réseau de neurones convolutif destiné à la FER et caractérisé par sa légèreté avec seulement 2.5 millions de paramètres. En exploitant des images de $48 \times 48 \times 3$ pour la reconnaissance des expressions

faciales, ce modèle s'établit comme une référence notable dans le domaine.

A-MobileNet : Le modèle A-MobileNet, exploré dans [107], est une autre réalisation notable dans le champ de la FER. Ce CNN léger, adapté à la reconnaissance de 7 à 8 émotions distinctes à partir d'images de $224 \times 224 \times 3$, comporte 3.4 millions de paramètres. Il exploite des modules d'attention pour réduire le nombre de paramètres et utilise une combinaison de fonctions de perte, y compris les fonctions *centered loss* et *softmax*, présentant une robustesse notable dans la détection des expressions faciales malgré sa complexité computationnelle réduite.

Mini-Xception : Le modèle Mini-Xception, avec son efficacité reconnue, a rencontré divers défis tels que la convergence lente et une précision de classification relativement faible en matière de reconnaissance des expressions faciales. Cependant, des améliorations substantielles ont été apportées en intégrant une convolution séparable en profondeur et en substituant la fonction d'activation ReLU par Mish, ce qui a amélioré tant la précision que la vitesse de convergence lors de la phase d'entraînement. De plus, l'adoption de CSPNet, une structure hiérarchique de type cross-stage, a enrichi les combinaisons de gradients et amélioré la fusion des caractéristiques multiples, atteignant une précision de validation maximale de 67.33% et 99.00% sur les ensembles de données FER2013 et CK+ respectivement [108].

EfficientNet : Introduit dans [109], EfficientNet est un modèle qui allie performances satisfaisantes et économie de ressources. En optimisant simultanément la profondeur, la largeur et la résolution de l'image des réseaux de neurones, il offre des performances supérieures par rapport à ses prédécesseurs, tout en conservant des tailles de modèles comparables. Bien que les modèles légers comme EfficientNet puissent parfois présenter une performance légèrement inférieure à celle de leurs homologues plus grands, leur efficacité en matière de ressources, combinée à une performance respectable, leur trouve une place dans diverses applications, en particulier là où les ressources informatiques sont limitées, comme dans les systèmes embarqués, les applications mobiles et les dispositifs

IoT [110].

2.2.5 Interprétation des modèles

L'interprétabilité des modèles CNN est une question se posant de plus en plus notamment dans les applications critiques où la décision prise par le modèle doit être justifiée [111]. Plusieurs approches récentes trouvées dans la littérature abordent ce problème en proposant différentes méthodes pour améliorer l'interprétabilité des modèles CNN. Les approches discutées dans la suite de cette section ont été appliquées dans le cas des systèmes FER.

Décomposition des contributions de caractéristiques : Une première approche pour l'interprétabilité des modèles consiste à décomposer les contributions des différentes caractéristiques des couches d'entrée à la prédiction finale du modèle. Lundberg et Lee [112] ont proposé la méthode SHAP (*SHapley Additive exPlanations*), qui attribue une importance à chaque caractéristique en fonction de sa contribution à la prédiction.

Visualisation des cartes d'activation : Les cartes d'activation permettent de visualiser les régions de l'image d'entrée auxquelles le modèle accorde de l'importance pour faire la prédiction. Des méthodes telles que Grad-CAM (*Gradient-weighted Class Activation Mapping*) [113] et LIME (*Local Interpretable Model-agnostic Explanations*) [114] ont été développées pour générer ces visualisations et fournir des informations sur les parties du visage pertinentes pour la prédiction d'émotions.

Modèles explicatifs : Les modèles explicatifs, tels que les arbres de décision, les réseaux bayésiens et les modèles linéaires généralisés, sont plus faciles à interpréter que les modèles d'apprentissage profond. Ils peuvent être utilisés comme approximations de modèles FER plus complexes pour mieux comprendre le processus de prise de décision [115].

Techniques d'explication post-hoc : Ces techniques sont appliquées après que le modèle a été entraîné pour expliquer ses décisions. Par exemple, Anchors [116] est une méthode post-hoc qui explique les prédictions d'un modèle en identifiant les conditions

minimales suffisantes pour qu'une prédiction donnée soit valable.

Zhang et al. [117] ont introduit un modèle FER interprétable basé sur la décomposition des régions faciales en parties significatives faisant apparaître les AU et l'apprentissage de caractéristiques distinctes pour chacune de ces parties. Cette approche permet de mieux comprendre comment les différentes régions du visage contribuent à la prédiction d'émotions et offre une représentation plus explicative du processus de décision du modèle.

Dans une étude de Liu et al. [118], des mécanismes d'attention ont été utilisés pour améliorer l'interprétabilité des systèmes FER. Cette approche permet d'identifier les parties les plus importantes du visage pour la détection des expressions de la douleur. Les mécanismes d'attention ont également été appliqués avec succès dans la détection des unités d'action (AU), qui sont des mouvements musculaires faciaux spécifiques associés à des expressions faciales émotionnelles [119].

L'ajout de mécanismes d'attention aux systèmes FER permet non seulement d'améliorer les performances de ces modèles, mais aussi de fournir des informations précieuses sur les régions du visage qui sont les plus discriminantes pour chaque expression émotionnelle [120]. En mettant en évidence les zones clés du visage, les mécanismes d'attention facilitent l'interprétation des résultats des modèles FER et aident les chercheurs et les praticiens à mieux comprendre le processus de prise de décision sous-jacent.

De plus, l'incorporation de méthodes d'interprétabilité dans les systèmes de FER peut contribuer à la réduction de biais potentiels. En identifiant les caractéristiques discriminantes et les régions faciales importantes pour la prédiction des expressions, il est possible de détecter les biais inhérents aux données d'entraînement ou aux modèles et de prendre des mesures pour les atténuer.

2.3 Modèle à base de CNN proposé

Dans cette section, nous détaillerons le modèle CNN pour la FER que nous avons conçu dans ces travaux de thèse. Il s'agit d'une architecture simple et efficace basée principalement sur le CNN, conçue pour la reconnaissance d'émotions. L'élément clé de notre proposition est la simplicité de l'architecture du modèle proposé le rendant adapté pour une implantation embarquée.

Pour démontrer la capacité de notre modèle à interpréter les résultats de prédictions, nous lui avons associé l'algorithme LIME (*Local Interpretable Model-agnostic Explanations*) [114]. L'utilisation de LIME, qui est une méthode d'interprétation de modèle d'apprentissage profond, aide à comprendre et à expliquer les prédictions générées par le modèle FER proposé dans ce chapitre [114] (voir annexe B.1.1 pour plus de détails sur la méthode LIME).

De surcroît, un des objectifs principaux de ce modèle simple proposé dans ce chapitre était d'explorer l'influence des paramètres du modèle sur sa complexité (traduit par le nombre de paramètres) sans sacrifier les performances globales du système FER.

2.3.1 Architecture proposée

La conception d'une architecture CNN pour la FER repose sur plusieurs principes clés permettant de garantir des performances élevées et une structure générale peu complexe et légère. Les choix effectués dans cette phase de conception sont essentiels pour obtenir un modèle efficace et adapté pour une implantation embarquée.

Couches de convolution : L'architecture, illustrée dans la figure 2.2, se compose de cinq couches convolutives. Ces couches sont suivies de trois couches de *MaxPooling* ayant un noyau de taille 2×2, dont le rôle principal est de réduire les dimensions spatiales des résultats de sortie d'une couche de convolution avant d'entrer dans la couche suivante. De plus, des couches de *Dropout* et de normalisation par lots (*Batch Normalization*) sont

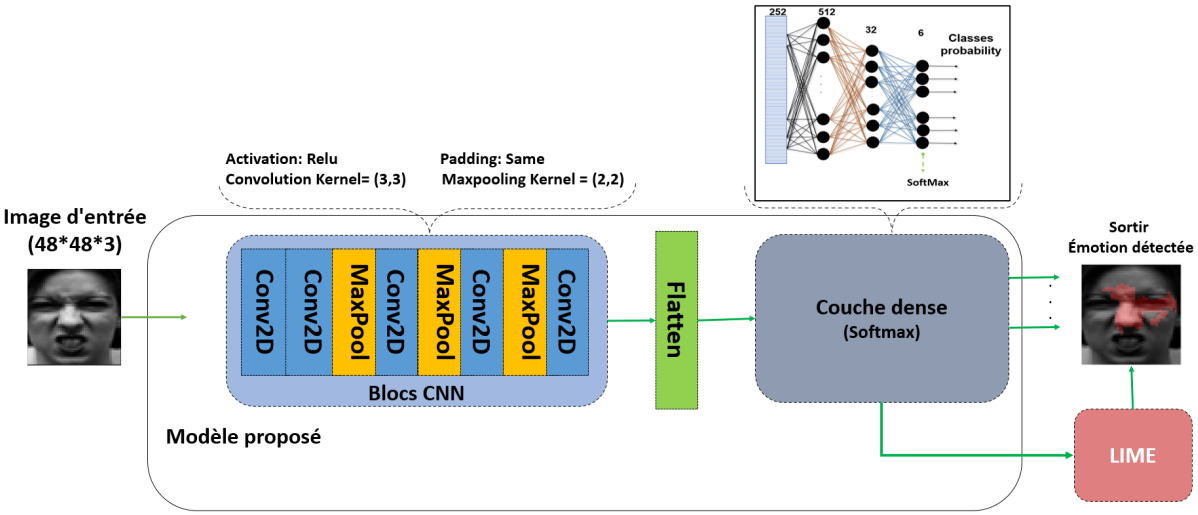


FIGURE 2.2 – Schéma des principales couches du modèle CNN proposé pour la FER [66]. Le modèle comprend plusieurs couches de convolution et de *pooling*, suivies de couches de normalisation de lots, de réduction de dimension et de classification. La sortie du modèle est une distribution de probabilités pour chaque expression faciale.

intégrées pour améliorer la robustesse et l'efficacité du modèle.

Nombre de filtres : Le choix du nombre de filtres dans les couches convolutives est crucial pour équilibrer la complexité du modèle et sa performance. Le modèle utilise un nombre réduit de filtres (32, 64, 128, 128 et 64) pour maintenir sa légèreté tout en préservant des performances satisfaisantes.

Taille du filtre : Des filtres de taille 3×3 sont utilisés dans toutes les couches convolutives. Cette taille de filtre est couramment employée dans les réseaux de neurones CNN pour les tâches de vision par ordinateur, car elle permet de conserver les détails de l'image d'entrée ou des couches de *features* tout en minimisant les coûts de calcul.

Couches entièrement connectées : La partie de classification du modèle se compose de couches entièrement connectées (les couches *Dense*). Ces couches transforment les caractéristiques extraites par les blocs de convolution en prédictions de probabilité pour les différentes expressions faciales. La dernière couche *dense* utilise une fonction d'activation comme la fonction *softmax* pour générer les probabilités finales pour chaque classe d'expression faciale.

Couches de *MaxPooling* : Trois couches de *MaxPooling* avec un noyau de taille 2×2 sont utilisées après certaines couches convolutives dans le modèle proposé, comme illustré à la figure 2.2. Elles sont utilisées pour réduire progressivement la dimensionnalité spatiale des cartes de caractéristiques, ce qui minimise le nombre total de paramètres à apprendre et réduit ainsi le coût en termes de ressources de calcul.

Couches de *Dropout* et de *BatchNormalization* : Le modèle proposé utilise des couches de *Dropout* après certaines couches entièrement connectées et convolutives. Ces couches sont utilisées pour prévenir le surapprentissage en désactivant de manière aléatoire certains neurones pendant l'entraînement. De plus, des couches de normalisation par lots (*BatchNormalization*) sont également appliquées pour améliorer la stabilité du réseau en normalisant les entrées à chaque couche du modèle, permettant ainsi un apprentissage plus rapide et plus efficace.

Algorithme LIME : Après la dernière couche convolutive, l'algorithme LIME est appliqué pour fournir l'interprétabilité et l'explicabilité des prédictions générées par modèle proposé.

2.4 Préparation des données

Dans cette section, nous présenterons les détails des étapes de préparation de données effectuées pour pouvoir entraîner et valider l'architecture du modèle proposé dans la section précédente. Les données (pour toutes les bases de données utilisées) ont été regroupées en six catégories d'émotions primaires. Ces émotions sont désignées par des étiquettes de classe (décrit en détails dans la suite de cette section). Dans le but d'exposer notre modèle à une diversité de données et de mettre au défi sa capacité à fonctionner de manière optimale dans différents environnements, nous avons choisi de le former en utilisant les bases de données CK+ [15] (une base de données de type « in-the-lab »), FER2013+ [121] (de type « in the wild »), et RAFdb [40] (de type « in the wild »). Cette

stratégie garantit une représentation adéquate et variée des émotions pendant la phase d'apprentissage :

$$E = \{\text{Colère}, \text{Dégoût}, \text{Peur}, \text{Joie}, \text{Tristesse}, \text{Surprise}\} \tag{2.1}$$

où chaque élément de E désigne une catégorie d'émotion spécifique. L'entraînement sur des ensembles de données variés permet non seulement de couvrir une large gamme d'émotions mais aussi de tester la robustesse du modèle CNN proposé dans des environnements et conditions divers et variés, assurant ainsi sa capacité de discriminer ces six catégories d'émotions à partir des images fournies à l'entrée.

2.4.1 Traitement préalable des données

Dans cette étude, nous utilisons la méthode de détection de visages frontaux Haar Cascade [10] pour identifier et localiser les visages frontaux dans les images avant de procéder à la détection des émotions. L'utilisation de Haar Cascade permet une détection efficace et précise des visages frontaux dans les images, ce qui est important pour garantir que le processus de détection des émotions soit effectué sur la bonne région du visage.

Après avoir identifié et extrait le visage frontal dans les images, les images d'entrée, quelque soit leur taille, sont redimensionnées à une taille de 48×48 pixels afin de réduire la taille du modèle. La nouvelle taille de l'image redimensionnée, I', est calculée comme suit :

$$I' = \text{redimensionner}(I, W', H') \tag{2.2}$$

où I est l'image d'origine, et W' et H' sont respectivement la largeur et la hauteur de l'image redimensionnée ($W' = H' = 48$ pixels).

Cette technique est couramment utilisée en apprentissage automatique pour réduire le nombre de paramètres du modèle et améliorer son efficacité, car les modèles plus grands

sont souvent plus sujets au surapprentissage et peuvent nécessiter davantage de ressources de calcul pour l'entraînement et le déploiement [122].

2.4.2 Augmentation des données

Pour entraîner le modèle, nous utilisons un générateur de données d'image qui lit, à partir des bases de données choisies, les images et les valeurs des emotions associées. Afin d'augmenter la taille de l'ensemble de données d'entraînement et d'améliorer la généralisation du modèle proposé, nous appliquons plusieurs techniques d'augmentation des données, notamment :

- **Mise à l'échelle** : Les valeurs des pixels sont normalisées en divisant chaque valeur par 255.
- **Rotation** : Les images sont pivotées aléatoirement dans une plage de ±30 degrés.
- **Cisaillement** : Les images subissent un cisaillement aléatoire avec une plage de 0.3.
- **Zoom** : Les images sont zoomées aléatoirement avec une plage de 0.3.
- **Retournement horizontal** : Les images sont retournées horizontalement (*flipped horizontally*) avec une probabilité de 50 %.

Les transformations d'augmentation sont appliquées sur chaque image I pour obtenir une image augmentée I_{aug} :

$$I_{aug} = \mathcal{T}(I) \tag{2.3}$$

où $\mathcal{T}(I)$ représente la combinaison des transformations d'augmentation appliquées à l'image I. Les images augmentées sont ensuite utilisées pour l'entraînement du modèle.

2.5 Apprentissage du modèle

2.5.1 Ajustement des hyperparamètres

Au cours des expérimentations, les hyperparamètres optimaux du modèle ont été déterminés en utilisant différentes combinaisons de paramètres et en observant les performances de validation. Les paramètres d'apprentissage optimaux du modèle ont été obtenus avec l'algorithme d'optimisation Adam, un taux d'apprentissage de 0.0001, un taux de décroissance de 10^{-6} et un total de 150 époques sur les images d'ensemble de données CK+, FER2013+ et RAFdb.

L'algorithme d'optimisation Adam (*Adaptive Moment Estimation*) est une méthode qui ajuste les taux d'apprentissage pour chaque paramètre en utilisant des estimations adaptatives des moments de premier et de second ordre [123]. Les mises à jour des paramètres sont effectuées en utilisant les équations suivantes :

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{2.4}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{2.5}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{2.6}$$

$$\hat{v}t = \frac{v_t}{1 - \beta_2^t} \tag{2.7}$$

$$\theta t + 1 = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{2.8}$$

où m_t et v_t sont les estimations des moments de premier et de second ordre, g_t est le gradient à l'étape t, β_1 et β_2 sont les coefficients de décroissance pour les estimations des moments, et ϵ est un terme de régularisation pour éviter la division par zéro.

Les paramètres d'apprentissage optimaux utilisés pour le modèle proposé sont résumés dans le Tableau 2.2.

TABLE 2.2 – Hyperparamètres optimaux pour le modèle proposé

Hyperparamètre	**Valeur**
Algorithme d'optimisation	Adam
Taux d'apprentissage	0.0001
Taux de décomposition	10^{-6}
Époques	700

2.5.2 Processus d'apprentissage

De manière générale, le processus d'apprentissage d'un modèle FER se déroule en plusieurs étapes cruciales pour garantir des performances optimales du modèle proposé. L'objectif principal est de minimiser la fonction de perte et d'augmenter la précision de la classification des émotions [124].

Dans le cas d'un problème multi-classes comme la reconnaissance des expressions faciales, la fonction de perte d'entropie croisée est souvent utilisée [125]. Elle mesure la divergence entre les probabilités prédites par le modèle et les étiquettes de référence des émotions recherchées. Le valeur faible en sortie d'une fonction de perte indique que le modèle est capable de bien apprendre les caractéristiques discriminantes des expressions faciales présentes dans les images d'entrée utilisées dans la phase d'apprentissage.

2.5.3 Évaluation des performances du modèle proposé

Pour évaluer l'efficacité du modèle proposé de manière rigoureuse, des paramètres spécifiques ont été utilisés. Ces paramètres sont détaillés en Annexe A (A.1.6). Cette section comporte trois parties principales pour chacun des trois ensembles de données utilisés (CK+ [15], RAFdb [40], et FER2013+ [121]) : les diagrammes d'entraînement, les rapports de classification, et la comparaison avec les travaux de l'état de l'art dans le domaine de la FER.

Diagrammes d'entraînement

Les diagrammes relatifs à la perte et à la précision du modèle proposé durant les phases d'entraînement et de validation pour les jeux de données CK+, RAFdb et FER2013+ sont illustrés dans les figures 2.3, 2.4 et 2.5, respectivement. Une convergence satisfaisante des courbes dans ces figures indique l'absence de sur-apprentissage ou de sous-apprentissage. Elle suggère également que le modèle a efficacement appris les caractéristiques pertinentes et qu'il manifeste une bonne généralisation sur des données inédites et non-utilisées dans la phase d'entraînement.

Analyse des performances sur le jeu de données CK+ Les graphiques relatifs à la perte, à la précision et à la matrice de confusion pour le jeu de données CK+, présentés dans la figure 2.3, montrent une convergence satisfaisante entre les courbes de l'ensemble d'entraînement et de validation. Ce phénomène indique une bonne adaptation du modèle aux données sans signes de sur-apprentissage ou de sous-apprentissage. De plus, le rapport de classification pour le CK+ (voir le tableau 2.3) a une forme presque parfaitement diagonale, indiquant que le modèle excelle dans la classification correcte des émotions faciales.

Analyse des performances sur le jeu de données RAFdb Les diagrammes de perte, de précision et la matrice de confusion pour le jeu de données RAFdb, illustrés dans la figure 2.4, montrent des signes manifestes de sur-apprentissage. Cela est particulièrement notable dans les phases ultérieures de l'entraînement, où l'écart entre les courbes de l'ensemble d'entraînement et de validation devient de plus en plus grand. En ce qui concerne le rapport de classification (voir le tableau 2.4), il est évident que le modèle éprouve des difficultés à classer correctement les différentes émotions. Ceci sera discuté en détail dans la section d'interprétation, où les défis associés à l'application d'un modèle relativement simple sur un jeu de données « sauvage » (de type « in-the-wild ») avec un

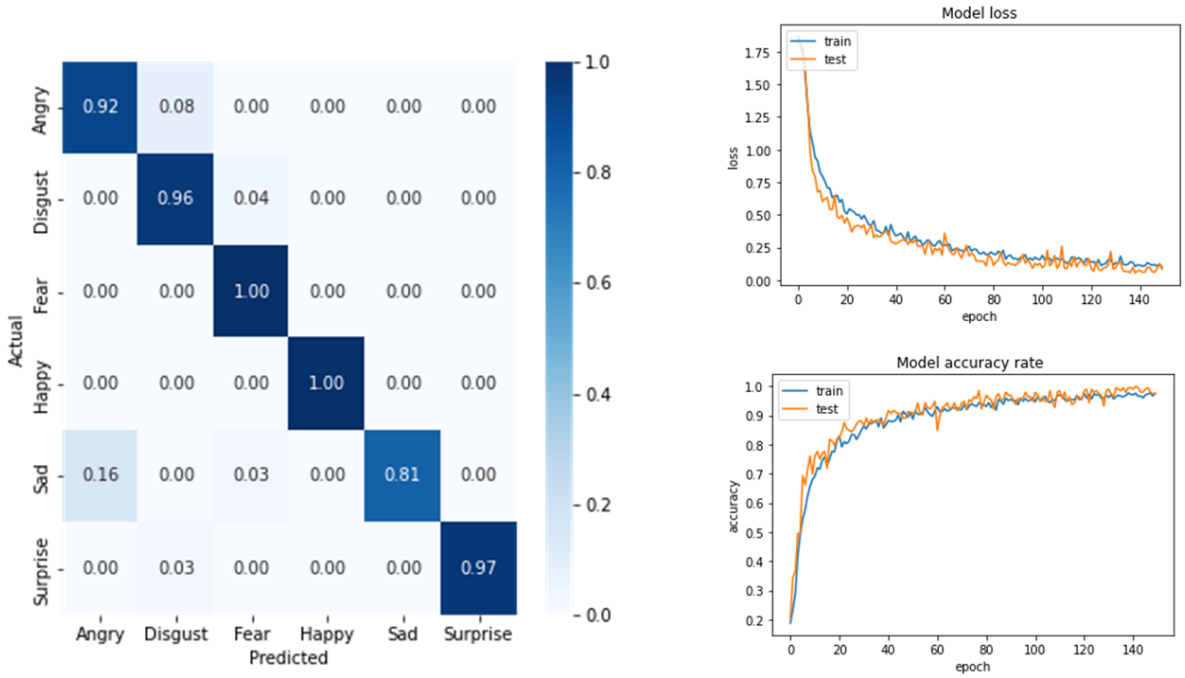


FIGURE 2.3 – Fonction de perte, de précision et la matrice de confusion du jeu de données CK+ pendant l'entraînement et la validation

grand nombre d'images et un déséquilibre entre les classes seront abordés.

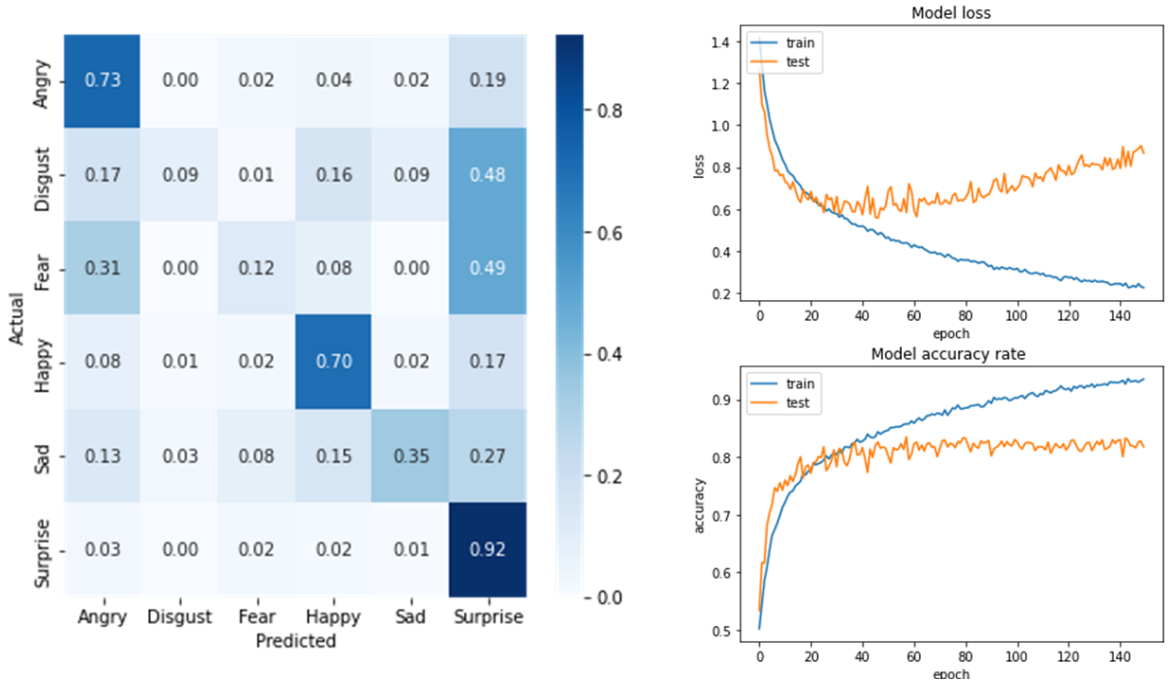


FIGURE 2.4 – Fonction de perte, de précision et la matrice de confusion du jeu de données RAFdb pendant l'entraînement et la validation

Analyse des performances sur le jeu de données FER2013+ Les indicateurs de la perte, la précision et la matrice de confusion pour l'ensemble de données FER2013+, illustrés dans la figure 2.5, révèlent également un sur-apprentissage notable. Cela est confirmé par l'augmentation de l'écart entre les courbes de perte et de précision pour les ensembles d'entraînement et de validation au cours de l'entraînement. De plus, le rapport de classification (voir le tableau 2.5) met en évidence que le modèle est peu performant en termes de classification correcte des différentes émotions. Ces problématiques seront examinées en détails dans la section d'interprétation, où nous aborderons les défis liés à l'utilisation d'un modèle plus simple sur des ensembles de données non contrôlés, contenant un grand nombre d'images et un déséquilibre notable entre les classes.

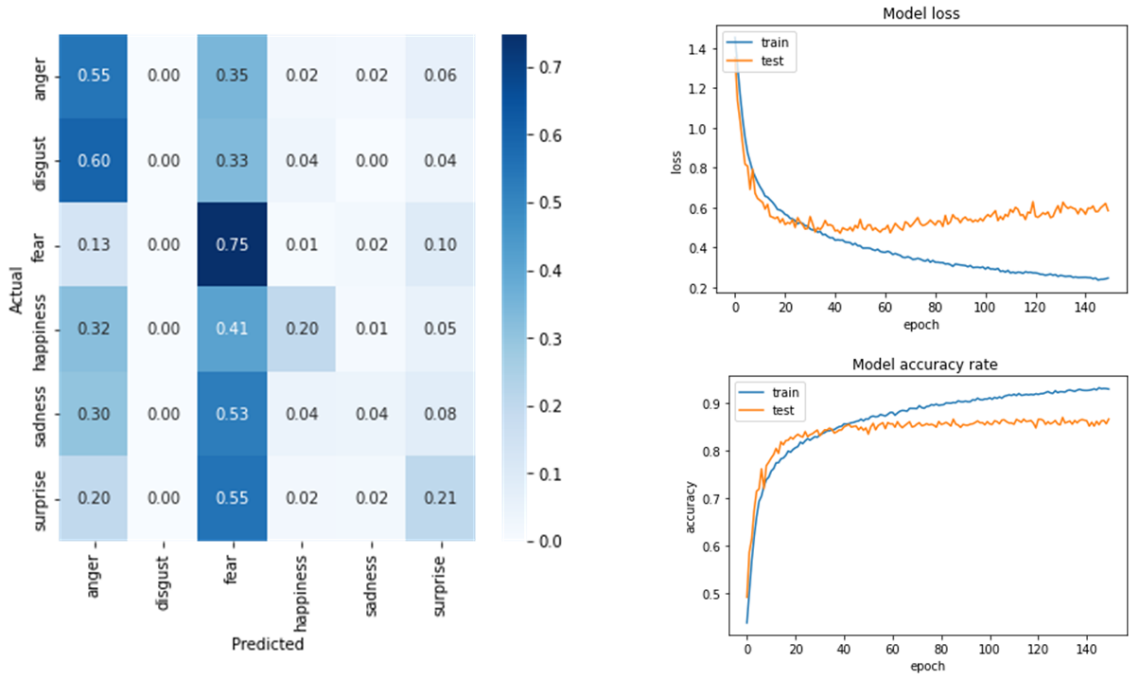


FIGURE 2.5 – Fonction de perte, de précision et la matrice de confusion du jeu de données FER2013+ pendant l'entraînement et la validation

Rapports de classification

Pour une compréhension plus approfondie des performances globales obtenues avec le modèle proposé dans ce chapitre, des rapports de classification sont présentés dans cette

section pour chaque ensemble de données traité (CK+, RAFdb et FER2013+). Ces rapports offrent un aperçu complet sur les performances globales du modèle en termes de précision, du rappel, et du score F1 pour chaque classe d'émotion (voir l'annexe A.1.6 pour les détails sur les métriques utilisées). Les tableaux 2.3, 2.4 et 2.5 présentent respectivement les rapports de classification pour CK+, RAFdb et FER2013+.

TABLE 2.3 – Rapport de classification pour l'ensemble de données CK+.

Émotion	**Précision**	**Rappel**	**Score F1**	**Nombre d'images**
Colère	0.87	0.92	0.89	36
Dégoût	0.87	0.96	0.91	27
Peur	0.94	1.00	0.97	32
Joie	1.00	1.00	1.00	36
Tristesse	1.00	0.81	0.89	31
Surprise	1.00	0.97	0.99	36
Exactitude			0.94	
Moyenne macro	0.95	0.94	0.94	198
Moyenne pondérée	0.95	0.94	0.94	198

Les résultats sur l'ensemble de données CK+ indiquent des performances très satisfaisantes du modèle proposé. L'exactitude totale est de 94%, et les émotions sont reconnues avec un score F1 élevé, particulièrement dans les catégories de la peur et de la surprise. La métrique de rappel pour la tristesse est légèrement plus basse (81%), mais globalement, le modèle montre une forte aptitude à généraliser sur cet ensemble.

TABLE 2.4 – Rapport de classification pour l'ensemble de données RAFdb.

Émotion	**Précision**	**Rappel**	**Score F1**	**Nombre d'images**
Colère	0.36	0.73	0.48	162
Dégoût	0.39	0.09	0.14	160
Peur	0.10	0.12	0.11	74
Joie	0.88	0.70	0.78	1186
Tristesse	0.79	0.35	0.48	478
Surprise	0.39	0.92	0.55	329
Exactitude			0.60	
Moyenne macro	0.49	0.48	0.42	2389
Moyenne pondérée	0.70	0.60	0.61	2389

Les performances sur l'ensemble RAFdb sont plus hétérogènes. L'exactitude globale

atteint 60%. La catégorie "Joie" est bien reconnue avec un score F1 de 78%, tandis que la catégorie "Peur" a une performance faible avec un score F1 de 11%. Il y a également une grande variabilité dans les scores de rappel, ce qui suggère que le modèle nécessite des ajustements pour cet ensemble.

TABLE 2.5 – Rapport de classification pour l'ensemble de données FER2013+

Émotion	**Précision**	**Rappel**	**Score F1**	**Nombre d'images**
Colère	0.25	0.55	0.34	644
Dégoût	0.00	0.00	0.00	57
Peur	0.06	0.75	0.11	167
Joie	0.83	0.20	0.32	1828
Tristesse	0.41	0.04	0.08	856
Surprise	0.45	0.21	0.28	900
Exactitude			0.24	
Moyenne macro	0.33	0.29	0.19	4452
Moyenne pondérée	0.55	0.24	0.26	4452

Les performances sur le FER2013+ sont assez inégales. L'exactitude globale est de 24%, ce qui indique des défis significatifs dans la classification des émotions. Les valeurs de précision et de rappel varient considérablement entre les catégories. Par exemple, la catégorie "Peur" présente un rappel très élevé de 75%, mais une précision extrêmement basse, résultant en un score F1 faible de 11%. La catégorie "Joie" montre une précision élevée de 83% mais un rappel faible de 20%. En outre, le modèle a une perte de 0.5763 et une précision de 86.99% pendant l'entraînement, ce qui suggère une inadéquation entre l'entraînement et les performances de validation. Les améliorations sont clairement nécessaires pour renforcer la robustesse du modèle proposé sur cet ensemble de données.

Comparaison avec l'état de l'art

Dans cette section, pour évaluer sa compétitivité par rapport à l'état de l'art, le modèle proposé dans ce chapitre a été comparé avec une série de travaux de recherche récents en reconnaissance des expressions faciales. Sur l'ensemble de données CK+, les travaux comparés incluent Sun et al. [126], DIA [127], Zhang et al. [128], Elfatih et al. [129],

Shahid et al. [130], SACNN-LSTM [131], et eXnet [132], comme présenté dans le tableau 2.6. Pour le jeu de données RAFdb, les travaux avec lesquels une comparaison est faite sont Muhamad et al. [133], A-MobileNet [134], SSA-ICL [5], LibreFace [135], DNFER [136], ECAN [137], Xiaoyu et al. [138], NCCTFER [139], et ces résultats sont exposés dans le tableau 2.7. Pour l'ensemble de données FER2013+, la comparaison inclut FST-MWOS [140], DNFER [136], NCCTFER [139], Sunyoung et al. [141], et A-MobileNet [134], comme présenté dans le tableau 2.8. La comparaison est principalement basée sur le critère de la précision car ce paramètre est le seul fourni dans ces travaux.

TABLE 2.6 – Comparaison en termes de précision entre le modèle proposé et les méthodes de l'état de l'art en reconnaissance des expressions faciales sur le jeu de données CK+. Les valeurs les plus élevées sont indiquées en gras tandis que les valeurs non fournies ou indisponibles sont représentées par un tiret (-).

Méthode	**Précision (en %)**
Sun et al. [126]	87.20
DIA [127]	89.51
Zhang et al. [128]	92.73
Elfatih et al. [129]	92.73
Le modèle proposé	94.00
Shahid et al. [130]	94.90
SACNN-LSTM [131]	95.15
eXnet [132]	**96.75**

Le tableau 2.6 présente une comparaison des performances en termes de précision entre le modèle proposé dans ce chapitre et d'autres méthodes de l'état de l'art en FER sur le jeu de données CK+. Il est à noter que notre modèle enregistre une précision globale de 94.00%, ce qui le positionne favorablement parmi les modèles étudiés, même s'il n'est pas le plus performant. Cette performance respectable indique que le modèle proposé est bien adapté aux spécificités de cet ensemble de données, très vraisemblablement dû à sa moindre complexité par rapport à des jeux de données comportant des images acquises dans des conditions plus proches de la réalité.

Le tableau 2.7 révèle que le modèle proposé dans ce chapitre obtient une précision de 60.00% sur l'ensemble de données RAFdb. Cette performance est nettement inférieure à

TABLE 2.7 – Comparaison en termes de précision entre le modèle proposé et les méthodes de l'état de l'art en reconnaissance des expressions faciales sur le jeu de données RAFdb. Les valeurs les plus élevées sont indiquées en gras tandis que les valeurs non-fournies ou indisponibles sont représentées par un tiret (-).

Modèles	**Précision (en %)**	**Paramètres (M)**
DNFER [136]	60.41	-
Le modèle proposé	60.00	**1.5**
LibreFace [135]	82.79	43
A-MobileNet [134]	84.49	3.4
Muhamad et al. [133]	84.91	2
Xiaoyu et al. [138]	87.58	-
NCCTFER [139]	87.97	-
SSA-ICL [5]	89.44	11
ECAN [137]	**89.77**	-

celle des autres modèles de l'état de l'art ; par exemple, ECAN [137] enregistre la plus haute précision de 89.77%. Néanmoins, notre modèle se distingue par un faible nombre de paramètres, soit 1.5 M. Cette modeste performance en terme de précision indique une inadaptation du modèle proposé aux images présentes dans l'ensemble de données RAFdb et montre notamment le problème de surapprentissage et les erreurs de classification observées dans la matrice de confusion. Ces questions seront explorées et discutées plus en détails dans la section d'interprétation.

TABLE 2.8 – Comparaison en termes de précision entre le modèle proposé et les méthodes de reconnaissance des expressions faciales de l'état de l'art sur l'ensemble de données FER2013+. Les valeurs les plus élevées sont indiquées en gras tandis que les valeurs non-fournies ou indisponibles sont représentées par un tiret (-).

Modèles	**Précision (en %)**	**Paramètres (M)**
FST-MWOS [140]	**90.41**	-
DNFER [136]	89.32	-
SUNYOUNG [141].	88.45	-
NCCTFER [139]	88.21	-
A-MobileNet [134]	88.11	3.4
Le modèle proposé	24.00	**1.5**

Selon le tableau 2.8, le modèle proposé dans ce chapitre montre des performances en termes de précision inacceptables (24.00%) et très loin des performances présentées

dans les autres travaux de l'état de l'art sur l'ensemble de données FER2013+. A titre d'exemple, l'approche FST-MWOS [140] obtient la meilleure performance avec une précision de 90.41%. Cette importante différence met en évidence des défis spécifiques à ce jeu de données, tels que l'équilibre des classes et la complexité des images, qui n'ont pas été efficacement traités par le modèle proposé. Nous aborderons ces problèmes plus en détail dans la section d'interprétation.

Interprétabilité des résultats obtenus Comme indiqué précédemment, le LIME est une méthode d'explication de modèle utilisée en FER pour comprendre et expliquer les prédictions générées par un modèle [142]. Le LIME vise à approximer la décision d'un modèle complexe (par exemple un modèle CNN) en le remplaçant par un modèle linéaire plus simple dans le voisinage local de l'espace de données utilisé. Cette approche permet d'obtenir une meilleure compréhension et interprétation des prédictions du modèle complexe [143]. En mettant l'accent sur le voisinage local, LIME offre une explication précise et pertinente de la manière dont le modèle prend ses décisions pour des échantillons individuels, rendant ainsi les modèles opaques plus transparents et compréhensibles pour les utilisateurs [143]. Mathématiquement, cela peut être représenté comme la minimisation de la fonction de perte suivante [142] :

$$\mathcal{L}(f, g, \pi_x) = \sum_i \pi_x(z_i)(f(z_i) - g(h(z_i)))^2 + \Omega(g) \tag{2.9}$$

où f est le modèle complexe, g est le modèle linéaire simple, π_x est une mesure de la proximité, z_i est un échantillon dans le voisinage, h est une fonction qui transforme l'échantillon dans l'espace interprétable, et $\Omega(g)$ est une pénalité sur la complexité du modèle g. Cette méthode fournit des explications visuelles et compréhensibles des prédictions d'émotions.

La Figure 2.6 présente les résultats obtenus en utilisant la méthode LIME pour essayer d'expliquer et interpréter les prédictions générées par le modèle proposé sur le jeu de

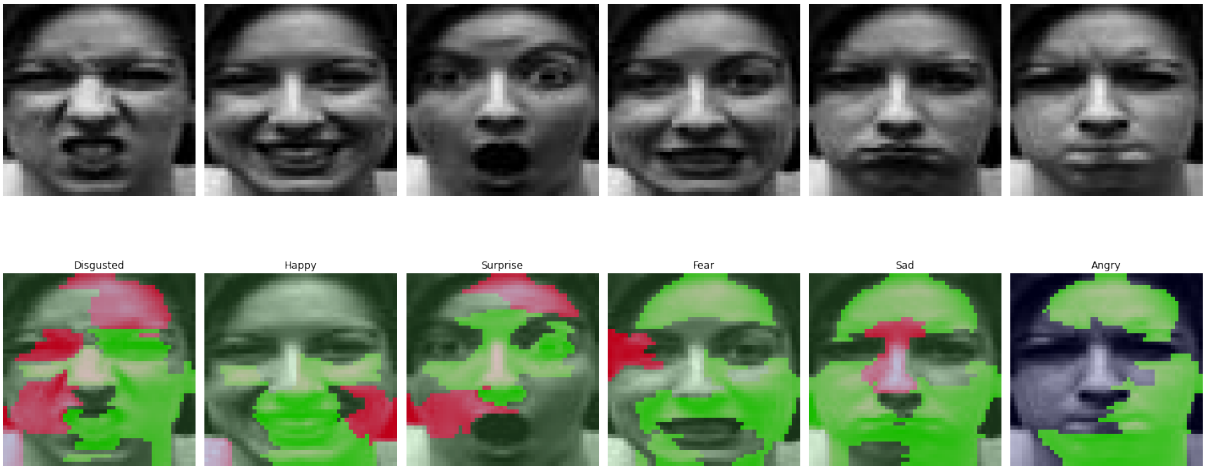


FIGURE 2.6 – La méthode LIME appliquée au jeu de données CK+.

données CK+. Les zones en vert sur la figure mettent en évidence les régions d'intérêt sur lesquelles le modèle se focalise pour réaliser des prédictions exactes. Compte tenu des très bonnes performances obtenues par le modèle proposé sur cet ensemble de données, corroborée par une précision de 94.00%, il n'est pas surprenant de voir une abondance de régions vertes dans la visualisation LIME. Cette concentration sur les régions pertinentes pour les AU intervenant dans l'émotion observée révèle une forte cohérence entre les prédictions du modèle et les caractéristiques qui sont généralement considérées comme significatives dans la FER humaine. Par conséquent, ces résultats montrent la capacité du modèle proposé à généraliser efficacement à partir de cet ensemble de données.

Contrairement à CK+, la figure 2.7 montre des résultats moins satisfaisants pour le modèle proposé appliqué au jeu de données RAFdb. Le modèle proposé atteint une précision de seulement 60.00%, et cela se reflète également dans la visualisation LIME. On observe moins de zones vertes, ce qui suggère que le modèle ne se concentre pas efficacement sur les régions du visage qui sont prépondérantes pour une FER précise. Ce comportement pourrait expliquer pourquoi le modèle proposé ne parvient pas à rivaliser avec les modèles de l'état de l'art pour ce jeu de données en particulier.

La figure 2.8 présente une interprétation visuelle des performances du modèle proposé sur l'ensemble de données FER2013+. Le modèle affiche une faible précision de 24.00%.

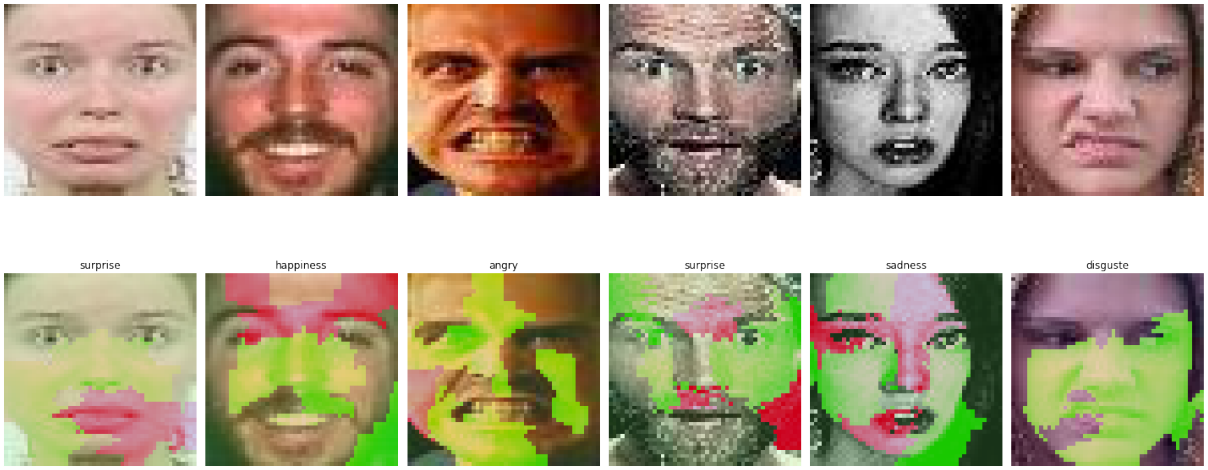


FIGURE 2.7 – La méthode LIME appliquée au jeu de données RAFdb.

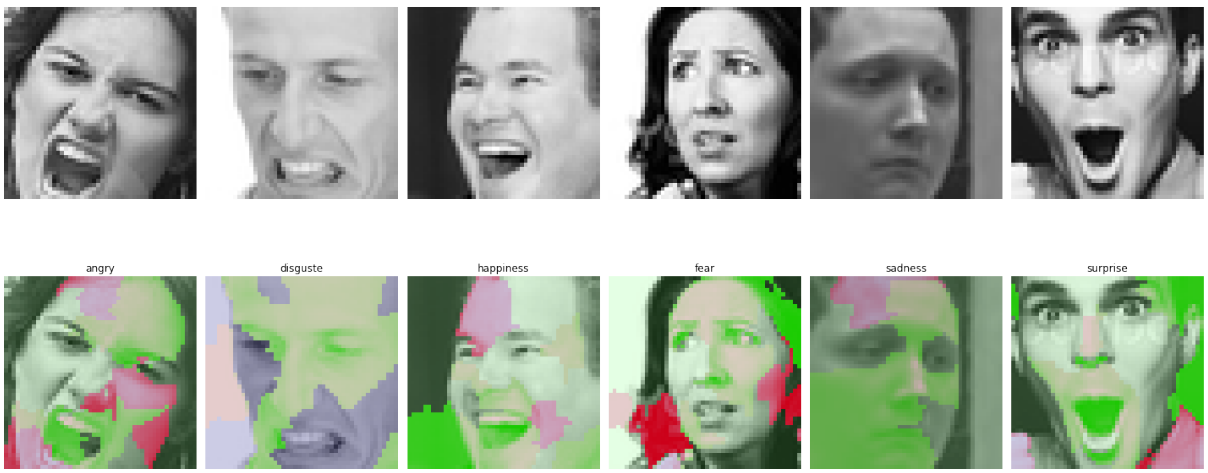


FIGURE 2.8 – La méthode LIME appliquée au jeu de données FER2013+.

La visualisation LIME renforce ces statistiques, car elle présente des régions vertes peu cohérentes ou absentes, indiquant que le modèle ne se concentre pas correctement sur les régions du visage qui sont typiquement informatives pour la FER.

2.5.4 Analyse comparative et interprétation des résultats

Après l'évaluation du modèle proposé sur divers ensembles de données, il est à noter que les performances varient significativement d'un ensemble de données à l'autre. Sur l'ensemble de données CK+, la précision obtenue est la meilleure sur les trois ensembles

de données considérés et atteint 0.94 ; sur RAFdb, elle est de 0.60 ; et sur FER2013+, elle n'est que de 0.24. Les résultats obtenus démontrent une hétérogénéité notable dans la classification des différentes émotions à travers les ensembles de données.

La variation des performances du modèle dans la classification des différentes émotions sur les ensembles de données CK+, RAFdb, et FER2013+ offre des perspectives intéressantes pour comprendre les nuances et les défis associés à la FER. Voici une analyse détaillée de ces variations.

— **Colère** : Le modèle proposé excelle à classer la colère dans CK+ avec une précision de 0.87. Cependant, il a des performances modestes sur RAFdb (0.36) et FER2013+ (0.25). Ce décalage peut être attribué aux différentes complexités des ensembles de données et aux problèmes de généralisation du modèle proposé.

— **Dégoût** : Le modèle montre une forte précision de 0.87 sur CK+, mais rencontre des défis sur RAFdb (0.39) et échoue complètement sur FER2013+ (0.00). Cela suggère que la reconnaissance de cette émotion nécessite un ajustement du modèle proposé, surtout pour des ensembles de données plus complexes comme le sont RAFdb et FER2013+.

— **Peur** : Avec une très bonne précision de 0.94 sur CK+, le modèle proposé démontre sa capacité élevée de reconnaissance de cette émotion. Néanmoins, cette capacité chute drastiquement sur RAFdb (0.10) et FER2013+ (0.06). Cette instabilité indique que la caractérisation de la peur est particulièrement sensible aux variations dans les ensembles de données.

— **Joie** : C'est l'émotion qui est la mieux classée sur tous les ensembles de données, atteignant le score parfait sur CK+ (1.00) et des scores élevés sur RAFdb (0.88) et FER2013+ (0.83). Ces résultats indiquent que le modèle est relativement robuste dans la classification de la joie.

— **Tristesse** : Bien que le modèle affiche une performance parfaite sur CK+ (1.00), il démontre une variabilité significative sur RAFdb (0.79) et FER2013+ (0.41).

Cette fluctuation suggère que des ajustements sont nécessaires pour améliorer la classification de la tristesse sur des ensembles de données plus divers.

— **Surprise** : Comme pour la joie, le modèle excelle sur CK+ avec un score parfait de 1.00. Cependant, il démontre des difficultés sur RAFdb (0.39) et FER2013+ (0.45). Ceci pourrait être dû à des facteurs tels que la variabilité des échantillons et les problèmes de généralisation du modèle.

— **Sur-apprentissage sur FER2013+** : Le faible score sur cet ensemble peut indiquer un problème de sur-apprentissage.

— **Inconsistances dans RAFdb** : Les variations importantes dans la précision des émotions suggèrent que le modèle a des difficultés avec cet ensemble.

— **Améliorations du Modèle** : Une attention particulière devra être portée aux techniques de régularisation et d'augmentation des données.

L'analyse des performances met en lumière les forces et les faiblesses du modèle actuel dans le domaine de la FER. Les résultats serviront de base pour des recherches ultérieures, visant à améliorer la généralisation du modèle sur différents ensembles de données.

2.6 Conclusion

Dans ce chapitre, plusieurs dimensions ont été analysées concernant la performance et l'interprétabilité du modèle de FER proposé dans ce chapitre. Alors que le modèle a obtenu des résultats convaincants sur des bases de données comme CK+, il a rencontré des difficultés sur des bases plus diverses de type « in-the-wild » dont les conditions d'acquisition sont moins contrôlées et incertaines, comme RAFdb et FER2013+. Ces contra-performances indiquent les limitations du modèle proposé en matière de généralisation aux conditions proches de la réalité [142].

L'interprétabilité, abordée via l'utilisation de LIME, a également montré ses limites en ne fournissant pas une vision suffisamment claire du comportement du modèle. De plus, le

problème de déséquilibre dans les données pose un défi supplémentaire, suggérant que des métriques et des fonctions de perte alternatives doivent être envisagées pour augmenter la robustesse du modèle.

Pour aborder ces défis, le chapitre suivant se concentrera sur l'utilisation d'une nouvelle architecture légère, basée également sur l'utilisation d'un réseau CNN. Cette architecture a l'avantage d'être suffisamment légère pour être déployée dans des environnements avec des ressources limitées tout en offrant de meilleures performances.

De surcroît, un système de codage binaire sera employé pour améliorer les paramètres d'évaluation en utilisant des AU. Cette approche vise non seulement à améliorer la performance mais également à augmenter l'interprétabilité du modèle. Parallèlement, des fonctions de perte et des mesures spécifiques seront introduites pour adresser les incohérences dues au déséquilibre des données.

Chapitre 3

Modèle CNN pour les unités d'action

Sommaire

3.1 Introduction

Le chapitre précédent a mis en évidence les défis associés à l'application des CNN pour la FER, notamment au niveau de la faible précision dans les ensembles de données réels, le manque d'interprétabilité et les problèmes de déséquilibre de classes. En dépit des limites, l'architecture légère proposée dans le chapitre précédent a montré un certain potentiel, mais elle nécessite des améliorations pour être applicable à des ensembles de données plus complexes [142].

Afin d'améliorer le modèle présenté dans le chapitre précédent, l'approche choisie et qui sera détaillée dans ce chapitre est d'aller vers l'intégration des Unités d'Action Faciales (AU) dans le modèle initial. Cette méthode devrait permettre d'améliorer non seulement la précision du modèle initial, mais également son interprétabilité. En utilisant les AU comme une entrée binaire supplémentaire, nous avons la possibilité de relier plus directement les mouvements faciaux à des émotions spécifiques, offrant ainsi une base plus solide pour la compréhension du fonctionnement du modèle.

De surcroît, afin d' atténuer les problèmes liés aux ensembles de données déséquilibrés, nous avons intégré la fonction de perte à entropie croisée pondérée (*weighted cross entropy*) et adopté la distance euclidienne comme nouvelle métrique de classification. Ces deux changements visent à optimiser la robustesse de l'architecture légère de CNN initiale proposée et détaillée dans le chapitre précédent.

Dans ce chapitre, nous détaillerons cette nouvelle approche basée sur les AU, ainsi que les nouvelles techniques liées aux déséquilibres des classes. L'objectif principal est de construire, sur les bases établies dans le chapitre précédent, une architecture plus robuste, fiable et interprétable pour la FER, tout en maintenant une complexité globale faible du modèle proposé le rendant utilisable dans les systèmes embarquées aux ressources limitées.

3.2 Modèle proposé

Le modèle proposé dans ce chapitre est illustré dans la Figure 3.1. Il repose sur une architecture CNN similaire à celle décrite précédemment pour la FER, avec notamment cinq couches convolutives suivies de couches de *MaxPooling*. Ces couches sont essentielles pour extraire les caractéristiques de bas niveau des images d'entrée, comme les bords et les textures. Pour prévenir le sur-apprentissage et accélérer l'apprentissage de manière générale, des couches de *Dropout* et de normalisation par lots (*BatchNormalization*) ont été ajoutées après chaque couche convolutive. Le nombre de filtres dans les couches convolutives a été soigneusement choisi pour équilibrer la complexité du modèle et les performances avec une taille de 3×3 sont dans toutes les couches convolutives.

La dernière partie du modèle se compose de couches entièrement connectées (*Dense*) qui transforment les caractéristiques extraites en probabilités de présence pour les différentes AU faciales. La couche finale utilise une fonction d'activation de type sigmoide, reflétant le besoin de prédire la probabilité de présence de chaque AU.

Une originalité de l'architecture proposée est dans la configurabilité de la dernière couche du réseau neuronal, qui varie selon les spécificités des ensembles de données utilisés, à savoir : 5 neurones de sortie pour le BP4D (« in-the-lab »), 13 pour le CK+ (« in-the-lab »), le RAFdb (« in-the-wild »), le FER2013 (« in-the-wild »), et le FER2013+ (« in-the-wild »), et 12 pour le DISFA (« in-the-lab »). Cette adaptabilité permet au modèle de gérer adéquatement les configurations différentes des AU et de s'aligner avec les particularités distinctives intrinsèques de chaque base de données.

Enfin, pour la visualisation et l'interprétation du modèle proposé dans ce chapitre, nous avons remplacé la méthode LIME par Grad-CAM, qui offre la possibilité de générer des "cartes thermiques" (*heat maps*) indiquant les régions de l'image qui contribuent de manière prépondérante à la prédiction de chaque unité d'action AU. L'intégration de Grad-CAM à la dernière couche convolutive du modèle proposé dans ce chapitre offre une compréhension plus précise de l'activité des AU dans les images d'entrée.

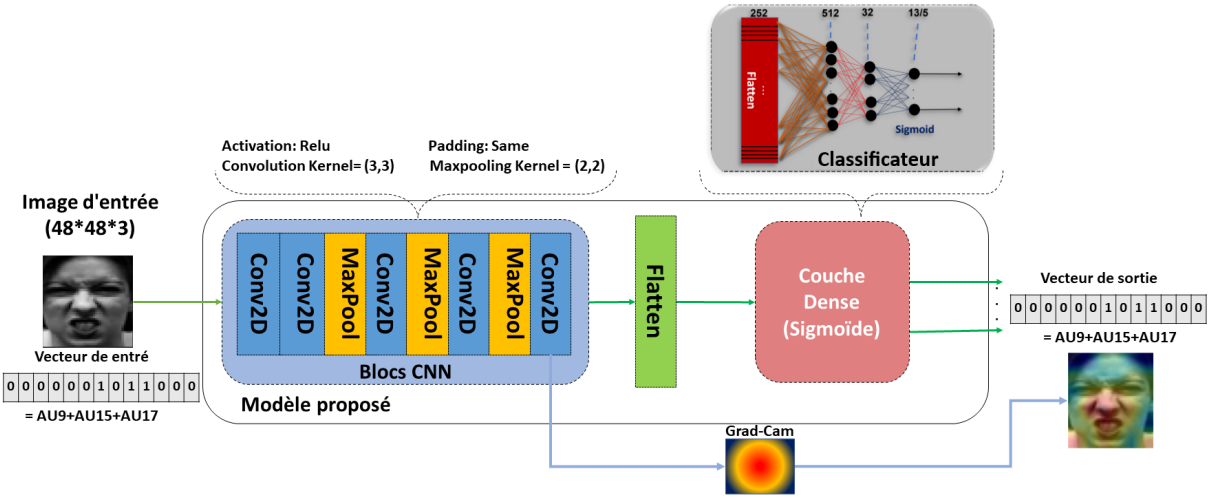


FIGURE 3.1 – L'architecture CNN pour la détection des AU comporte cinq couches convolutives, des couches de regroupement et d'abandon pour la robustesse. Les couches pleinement connectées gèrent la classification avec une activation sigmoide en sortie. Grad-CAM est utilisé pour la visualisation des AU.

3.3 Prétraitement des données

Le succès d'un système de détection des AU repose grandement sur la qualité et la représentativité des données d'entrée utilisées. Dans cette étude, nous exploitons deux types de données : les images faciales et leurs vecteurs d'AU correspondants, encodés en format binaire.

Il est crucial de noter que, contrairement au premier modèle présenté dans le chapitre précédent où chaque image était attribuée à une seule émotion, le contexte de la détection des AU est différent. Une seule image peut comporter plusieurs AU activées simultanément. Cette particularité nous place dans un scénario de classification multi-labels, ce qui complexifie le processus d'apprentissage et de classification.

La labélisation des AU est effectuée conformément à la table du FACS, introduite par Ekman et al. [12] (détaillé dans la section 1.2.1 du chapitre 1). Ce système de codage fournit un cadre méthodologique pour l'identification et la classification des AU.

L'ensemble de données que nous introduisons dans notre architecture comprend des vecteurs caractéristiques. La première colonne de chaque vecteur identifie le nom de

l'image, tandis que les colonnes suivantes contiennent des valeurs binaires, indiquant l'absence (0) ou la présence (1) de chaque AU particulière en accord avec le FACS.

Les méthodes de prétraitement appliquées à ces données, avant leur utilisation dans le modèle, sont exposées en détail dans les sections ultérieures de ce chapitre.

3.3.1 Prétraitement des images

Comme indiqué dans le chapitre précédent, nous avons recours à plusieurs techniques pour le prétraitement des images. Nous utilisons tout d'abord la méthode de détection de face frontale Haar Cascade [10] pour identifier et localiser les visages frontaux dans les images. Cette étape est cruciale pour garantir que le processus de détection des AU s'effectue sur la bonne zone du visage.

Suite à la détection du visage, les images sont redimensionnées à une taille standard de 48×48 pixels. Cette standardisation des dimensions des images réduit la complexité du modèle, rendant ainsi l'apprentissage plus efficace. En effet, les modèles de grande taille sont souvent plus susceptibles d'être surajustés et nécessitent davantage de ressources de calcul pour l'apprentissage et le déploiement [122].

3.3.2 Codage binaire des AU

L'incorporation du système FACS à travers les AU, notamment au cours de la phase d'apprentissage et sur divers ensembles de données, vise principalement à améliorer l'interprétabilité des expressions faciales dans les images d'entrée. En employant plusieurs bases de données distinctes, dont CK+, RAFdb, FER2013+, BP4D, et DISFA, pour élaborer un modèle de FER robuste, il est impératif de tenir compte de la spécificité de chaque ensemble de données lors de la phase d'apprentissage pour assurer une formation réussie du modèle. Les détails pertinents sont présentés dans les sous-sections suivantes.

Prétraitement des jeux de données comportant des annotations AU

Pour des bases de données comme BP4D et DISFA, qui fournissent déjà des annotations AU, le prétraitement est relativement simplifié. Chaque AU active dans une image est marquée par une valeur binaire "1", tandis que les AU inactives sont indiquées par un "0". Ces vecteurs binaires, au-delà de servir de cibles pour l'entraînement supervisé, facilitent également la gestion et la manipulation des données durant le processus d'apprentissage en créant une structure de données homogène.

Gestion des jeux de données dépourvus d'annotations AU

Pour les jeux de données CK+, RAFdb, et FER2013+, qui ne fournissent pas d'annotations AU, nous avons classifié les images par état émotionnel. Un vecteur de 13 AU a été créé pour chaque état émotionnel en se basant sur le codage FACS, décrit auparavant. Ces vecteurs sont utilisés comme des référentiels pendant les phases d'entraînement et de validation, établissant ainsi un standard cohérent contre lequel les prédictions du modèle peuvent être vérifiées et affinées.

Structuration de données pour l'apprentissage

Suite au traitement des différents jeux de données, nous avons compilé et structuré ces informations en *dataframes* pour les phases d'entraînement et de validation. Ces *dataframes* incluent des vecteurs binaires indiquant la présence ou l'absence d'AU pour chaque image, servant d'input au modèle proposé dans ce chapitre. Il est à noter qu'une *dataframe*, dans ce contexte, sert de structure de données bidimensionnelle étiquetée, capable de contenir des données de types variés et facilitant l'organisation des informations de manière propre et accessible.

Gestion des disparités entre les jeux de données

Étant donné que chaque base de données présente ses propres particularités, telles que des états émotionnels manquants ou des sujets affichant diverses expressions faciales, des étapes de curation de données ont été mises en œuvre pour homogénéiser ces ensembles, garantissant ainsi un entraînement plus uniforme du modèle proposé dans ce chapitre. Cette gestion attentive des données assure non seulement la qualité mais également la robustesse des prédictions du modèle en contextes variés.

TABLE 3.1 – Le tableau FACS montrant la liste d'émotions et les AU correspondants (au moins actives pour une émotion donnée selon Ekman [12]). NB : L'expression naturelle n'a pas d'AU activée.

émotions	**AU activés**
Colère	4, 5, 7, 23
Dégoût	9, 15, 17
Peur	1, 2, 4, 5, 7, 20, 26
Joie	6, 12
Tristesse	1, 4, 15
Surprise	1, 2, 5, 26

Fichiers d'entraînement des jeux de données CK+, RAFdb et FER2013+ Les images issues des jeux de données CK+, RAFdb et FER2013+ ont été classifiées selon leur état émotionnel. Pour chaque jeu de données, une portion d'environ 20% des images a été allouée pour la validation, tandis que le reste a été consacré à l'entraînement. S'appuyant sur le codage FACS [12], comme illustré dans le tableau 3.1, 13 AU ont été scrupuleusement sélectionnées pour coder les expressions émotionnelles manifestes dans les images de ces trois jeux de données.

Les 13 AU choisies couvrent un spectre large et diversifié d'expressions faciales, et sont suffisantes pour représenter les expressions les plus couramment observées, conformément aux directives du FACS (comme en témoigne le tableau 3.1) et présentes dans les jeux de données exploités. Ainsi, chaque image a été transformée en un vecteur de 13 valeurs binaires via une technique d'encodage à plusieurs étiquettes (*multi-label one-hot encoding*).

Chacun des jeux de données a été ensuite organisé en un fichier d'entraînement, structuré avec 14 colonnes : la première colonne est consacrée au nom de l'image et les 13 colonnes suivantes signalent la présence (1) ou l'absence (0) des AU spécifiées (AU01, AU02, AU04, AU07, AU09, AU12, AU15, AU17, AU20, AU23, et AU26).

Il est important de noter que ces fichiers d'entraînement, ou *Dataframes*, jouent un rôle crucial dans les phases d'entraînement et de validation du modèle en offrant une structure ordonnée et accessible pour le stockage des données. La *Dataframe*, par sa structure, permet une manipulation aisée des données, facilitant ainsi les opérations de traitement et d'analyse ultérieures, tout en garantissant l'intégrité et la clarté des données tout au long du processus d'apprentissage machine.

filename	AU1	AU2	AU4	AU5	AU6	AU7	AU9	AU12	AU15	AU17	AU20	AU23	AU26
0_S042_006_00000017.png	1	1	0	0	1	0	0	0	0	0	0	0	0
0_S109_004_00000013.png	1	1	1	0	1	1	1	0	0	0	0	0	0
0_S034_005_00000010.png	1	1	0	0	1	0	0	0	0	0	0	0	0
0_S097_004_00000030.png	1	0	1	0	1	1	1	0	0	0	0	0	0
0_S124_007_00000024.png	1	1	0	0	1	0	0	0	0	0	0	0	0
0_S032_005_00000016.png	0	1	1	0	1	1	1	0	0	0	0	0	0

(a) CK+, FER2013+ et RAFdb

filename	AU1	AU2	AU4	AU5	AU6	AU9	AU12	AU15	AU17	AU20	AU25	AU26
SN016_2471.jpg	1	1	0	0	0	0	0	0	0	0	0	0
SN016_951.jpg	0	0	0	0	1	0	1	0	0	0	1	0
SN007_1992.jpg	0	0	0	0	1	0	1	0	0	0	0	0
SN017_1452.jpg	0	0	0	0	1	0	1	0	0	0	0	0
SN030_1599.jpg	0	0	1	0	1	1	0	0	0	0	1	0

(b) DISFA

filename	AU06	AU10	AU12	AU14	AU17
F001_T1_2455.jpg	0	1	0	0	0
F008_T4_0846.jpg	1	1	1	1	0
F016_T6_1271.jpg	0	1	0	0	0
F021_T1_0301.jpg	1	0	1	0	0
M003_T4_1871.jpg	0	1	1	0	0
M008_T6_0315.jpg	1	1	0	1	0
M008_T6_0295.jpg	1	1	0	1	0

(c) BP4D

FIGURE 3.2 – Un exemple de données d'entrée créé pour entraîner le modèle en utilisant les jeux de données suivants : (a) CK+, FER2013+ et RAFdb, (b) BP4D et (c) DISFA. Ces vecteurs démontrent comment les informations sur les unités d'action ou AU sont intégrées dans les données d'entrée lors de l'entraînement du modèle.

Fichier d'entraînement des jeux de données BP4D et DISFA Malgré l'organisation des jeux de données BP4D et DISFA par émotions, il est à noter que plusieurs émotions dans ces jeux de données ne sont pas bien regroupées, et certaines d'entre elles ne sont même pas disponibles. Par conséquent, nous avons décidé d'utiliser les intensités des unités d'action (AU) fournies par les créateurs de ces jeux de données.

Le jeu de données BP4D fournit des intensités pour cinq AU (AU06, AU10, AU12, AU14 et AU17) avec des valeurs allant de 0 à 9. De même, le jeu de données DISFA offre des intensités pour douze AU (AU1, AU2, AU4, AU5, AU6, AU9, AU12, AU15, AU17, AU20, AU25, et AU26) sur une échelle de 0 à 5. Pour préparer les jeux de données pour l'entraînement et la validation, nous avons initialement sélectionné uniquement les images qui avaient au moins une AU activée avec une intensité supérieure à 4. Les vecteurs d'entrée de chaque jeu de données sont illustrés dans la figure 3.2.

3.4 L'entraînement du modèle proposé

3.4.1 Processus d'entraînement

Nous avons déterminé les paramètres d'entraînement optimaux pour le modèle proposé dans ce chapitre de manière expérimentale, en utilisant une approche *test-and-trial*. Les hyperparamètres sélectionnés sont fournis dans le tableau 3.2. Comme mentionné auparavant, pour un problème multi-étiquettes (multi-classes), nous utilisons la fonction de perte d'entropie croisée pondérée, calculée à partir de la racine carrée de l'erreur quadratique moyenne (MSE) et de la distance euclidienne, pour suivre la métrique lors de l'entraînement du modèle.

$$\text{Loss}_{\text{WCE}}(\mathbf{AU}_{\text{true}}, \mathbf{AU}_{\text{pred}}) = -\sum_{i} w_{\text{AU},i} \cdot \mathbf{AU}_{\text{true},i} \log(\mathbf{AU}_{\text{pred},i}) \tag{3.1}$$

où :

— $\mathbf{AU}_{\text{true}}$ est le vecteur des activations réelles des AU.

— $\mathbf{AU}_{\text{pred}}$ est le vecteur des prédictions du modèle pour les AU.

— $w_{\text{AU},i}$ est le poids associé à l'AU i.

$$\text{Loss}_{\text{MSE}}(\mathbf{AU}_{\text{true}}, \mathbf{AU}_{\text{pred}}) = \frac{1}{N} \sum_{i} (\mathbf{AU}_{\text{true},i} - \mathbf{AU}_{\text{pred},i})^2 \tag{3.2}$$

où :

— $\mathbf{AU}_{\text{true}}$ est le vecteur des activations réelles des AU.

— $\mathbf{AU}_{\text{pred}}$ est le vecteur des prédictions du modèle pour les AU.

— N est le nombre total d'AU dans $\mathbf{AU}_{\text{true}}$ (ou $\mathbf{AU}_{\text{pred}}$).

3.4.2 Augmentation de données

Pour accroître la diversité des données d'entraînement et améliorer la robustesse du modèle, nous mettons en œuvre des stratégies d'augmentation de données. Les images subissent diverses transformations pour générer de nouvelles instances pour l'entraînement. Il convient de souligner que les techniques d'augmentation déployées dans ce contexte de détection des AU sont analogues à celles employées dans le chapitre précédent consacré à la FER (voir la section 2.4.2 pour plus de détails).

3.4.3 Réglage des hyperparamètres

Les hyperparamètres utilisés pour le modèle de détection des AU demeurent identiques à ceux déployés dans le chapitre précédent pour la FER.

TABLE 3.2 – Récapitulatif des hyperparamètres optimaux utilisés pour la détection des AU et la FER.

Hyperparamètre	**Valeur**
Algorithme d'optimisation	Adam
Taux d'apprentissage	0.0001
Taux de décomposition	10^{-6}
Époques	700

Problème multi-classes dans des ensembles de données déséquilibrés

Dans la détection des AU, nous sommes confrontés à une problématique de classification multi-étiquettes sur des ensembles de données déséquilibrés. Pour pallier cette difficulté, le choix d'une fonction de perte adéquate est d'une importance capitale. Dans ce contexte, nous avons opté pour la fonction de perte d'entropie croisée pondérée. La pondération pour chaque classe d'AU est définie comme suit :

$$\text{poids}(AU) = \frac{N}{\sum_{i=1}^{N} I(AU_i = 1)} \tag{3.3}$$

où :

— N est le nombre total d'échantillons ;

— AU désigne la classe d'unité d'action spécifique pour laquelle le poids est calculé ;

— I est la fonction indicatrice, renvoyant 1 si la condition est satisfaite, et 0 sinon.

Il est à noter que cette formule est en réalité équivalente à la division du nombre total d'échantillons par le nombre de présences d'une classe d'AU particulière.

En matière de métriques de suivi, nous avons également employé la distance euclidienne, pour évaluer l'efficacité globale du modèle sur l'ensemble des classes d'AU. Le processus de réglage des hyperparamètres, identiques à ceux utilisés dans le chapitre précédent, a été itératif et a impliqué divers ajustements, notamment en ce qui concerne la taille des lots, la régularisation et les taux d'abandon.

3.5 Évaluation des performances du modèle proposé

Pour garantir une évaluation rigoureuse de l'efficacité du modèle proposé dans ce chapitre, nous avons mis en œuvre des métriques d'évaluation spécifiques, détaillées en Annexe A (A.1.6). Cette section est subdivisée en trois volets principaux pour chaque ensemble de données utilisé (CK+, RAFdb, FER2013+, BP4D et DISFA spécifiques à ce chapitre) :

1. Les courbes d'apprentissage, qui illustrent la performance du modèle tout au long de la phase d'entraînement.

2. Les rapports de classification, fournissant des analyses détaillées des résultats obtenus sur différentes classes d'AU/Émotions.

3. La comparaison avec des travaux antérieurs, notamment ceux spécialisés dans la détection des AU/Émotions, afin d'établir le positionnement de notre modèle par rapport à l'état de l'art.

3.5.1 Diagrammes d'entraînement

Les figures 3.3, 3.4, 3.5, 3.6 et 3.7 présentent les diagrammes de perte et de précision du modèle au cours des phases d'entraînement et de validation sur divers ensembles de données.

Le but principal de cette section est d'évaluer minutieusement l'entraînement du modèle sur plusieurs jeux de données. Cela implique de surveiller la réduction des valeurs d'erreur, qui sont évaluées à l'aide de mesures telles que l'entropie croisée pondérée et la distance euclidienne, tout au long du processus d'entraînement.

De plus, il est essentiel de veiller à ce qu'il n'y ait pas de phénomènes de sur- ou sous-apprentissage pendant les phases d'entrainement et de validation du modèle.

Analyse des performances sur le jeu de données CK+ Les résultats obtenus sur le jeu de données CK+, présentés dans la figure 3.3, révèlent une convergence satisfaisante des courbes au cours des étapes d'entraînement et de validation. Cette convergence signale l'absence de sur- et sous-apprentissage durant ces phases. Il est à noter qu'en exploitant la nouvelle approche basée sur l'intégration des AU, le modèle proposé dans ce chapitre sur le jeu de données CK+ démontre une amélioration des performances, notamment en matière d'entropie croisée pondérée et de distance euclidienne.

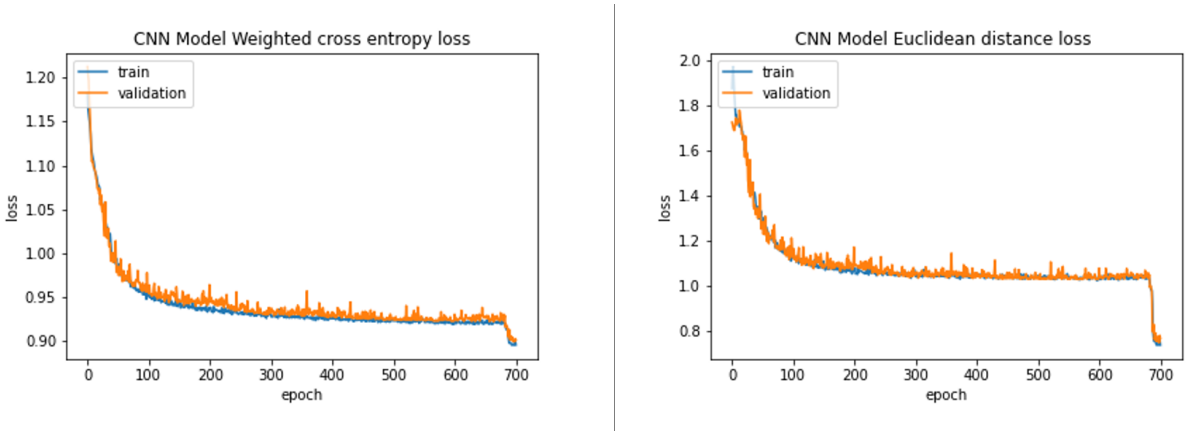


FIGURE 3.3 – Analyse de l'entropie croisée pondérée et de la distance euclidienne sur le jeu de données CK+ pendant les étapes d'entrainement et de validation du modèle.

Analyse des performances sur le jeu de données RAFdb Les résultats relatifs au jeu de données RAFdb, présentés dans la figure 3.4, mettent en avant une progression notable des fonctions de perte par rapport au modèle du chapitre précédent. Malgré la présence d'un sur-apprentissage léger, ce dernier reste insignifiant, attestant ainsi que le modèle proposé est bien plus abouti comparé aux essais antérieurs. Cette observation indique que l'ajustement méthodologique appliqué dans ce chapitre a porté ses fruits, principalement concernant l'entropie croisée pondérée et la distance euclidienne.

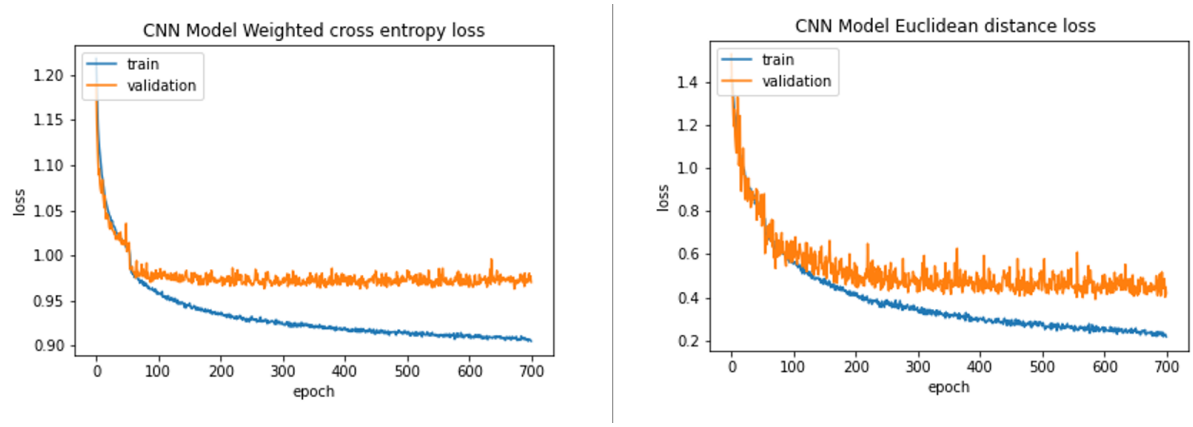


FIGURE 3.4 – Analyse de l'entropie croisée pondérée et de la distance euclidienne sur le jeu de données RAFdb pendant les étapes d'entraînement et de validation du modèle.

Analyse des performances sur le jeu de données FER2013+ Les résultats associés au jeu de données FER2013+, exposés dans la figure 3.5, révèlent une progression notable des fonctions de perte en comparaison avec les résultats du chapitre précédent. Il convient néanmoins de signaler l'apparition de signes de sur-apprentissage, nécessitant des rectifications supplémentaires afin d'optimiser le modèle. Malgré cette présence de sur-apprentissage, les fonctions de perte illustrent une amélioration des performances, indiquant encore une fois que l'approche basée sur l'intégration des AU confère des avantages appréciables, en particulier au niveau de l'entropie croisée pondérée et de la distance euclidienne.

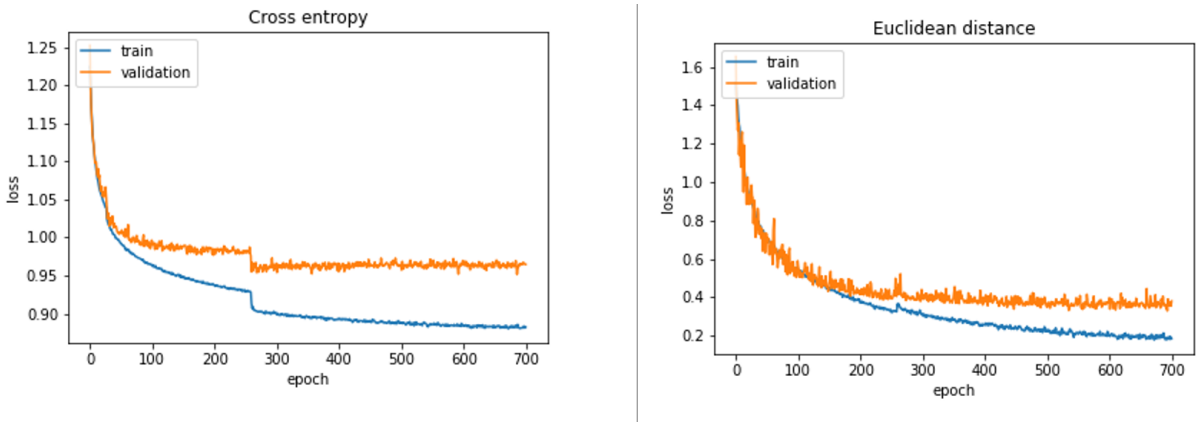


FIGURE 3.5 – Analyse de l'entropie croisée pondérée et de la distance euclidienne sur le jeu de données FER2013+ pendant les étapes d'entraînement et de validation du modèle.

Analyse des performances sur le jeu de données BP4D Les résultats associés au jeu de données BP4D, présentés dans la figure 3.6, mettent en lumière une concordance optimale des courbes pendant les phases d'entraînement et de validation. Ce phénomène témoigne d'une absence complète de sur- ou sous-apprentissage, illustrant ainsi que le modèle a été entraîné de façon optimale sur cette base de données.

Analyse des performances sur le jeu de données DISFA En ce qui concerne le jeu de données DISFA, les représentations graphiques mettent en évidence une harmonisation réussie des courbes durant les étapes d'entraînement et de validation. Cette harmonisation,

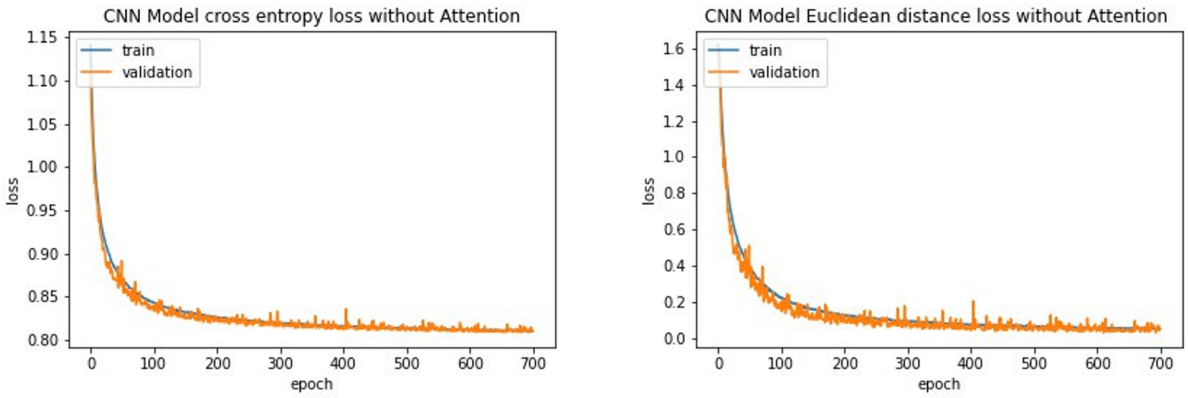


FIGURE 3.6 – Analyse de l'entropie croisée pondérée et de la distance euclidienne sur le jeu de données BP4D pendant les étapes d'entraînement et de validation du modèle.

observable dans la figure 3.7, révèle que le modèle s'est non seulement bien accommodé au jeu de données, mais aussi qu'il ne présente aucun symptôme de sur- ou sous-apprentissage. En se basant sur l'entropie croisée pondérée et la distance euclidienne comme baromètres de performance, il est évident que le modèle affiche des résultats positifs avec ce jeu de données.

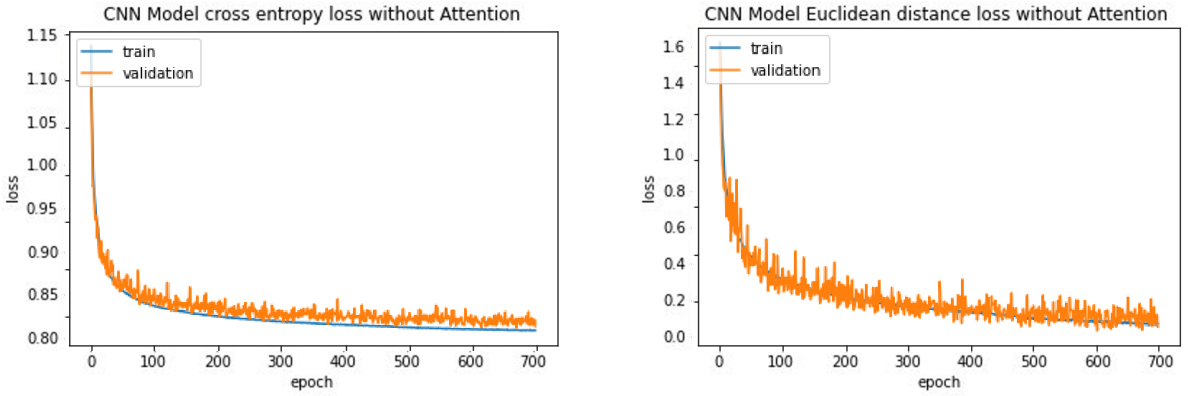


FIGURE 3.7 – Fonction de perte d'entropie croisée pondérée et distance euclidienne pour le jeu de données DISFA durant les phases d'entraînement et de validation.

3.5.2 Comparaison avec l'état de l'art

Dans cette section, nous avons structuré la présentation des résultats en deux parties distinctes. La première partie se concentre sur la détection des AU et fait référence à

trois ensembles de données : CK+, BP4D et DISFA. Dans cette section, nous avons comparé le modèle proposé dans ce chapitre avec différentes méthodes de l'état de l'art : Zhang et al. [144], FS [145], ARL [97], STRAL [99], LGR [146], MCFE [147], IDEN [148], CWCF [149], et JAU [150] pour BP4D et DISFA ; JPML [151], Simge [81], Chen1 [101], Chen2 [101], et Elef [58] pour CK+.

La deuxième partie de cette section se rapporte à la reconnaissance des expressions faciales et englobe trois ensembles de données supplémentaires : CK+, RAFdb et FER2013+. Dans cette partie, le modèle proposé dans ce chapitre est comparé avec plusieurs autres méthodes de l'état de l'art dans le domaine : LibreFace [135], SSA-ICL [5], ECAN [137], A-MobileNet [134], DNFER [136], Muhamad et al.[133], Xiaoyu et al. [138], et NCCT-FER [139], ainsi que des travaux supplémentaires comme FST-MWOS [140] et Sunyoung et al. pour FER2013+.

Comparaison du modèle proposé avec l'état de l'art en détection d'AU sur CK+, BP4D et DISFA.

Dans cette sous-section, nous mettons en avant la comparaison du modèle proposé dans ce chapitre avec d'autres méthodes avancées en détection des AU. Cette évaluation s'étend sur trois bases de données distinctes : CK+, BP4D et DISFA. Le score F1 sert de métrique principale pour cette évaluation. Les résultats correspondant à chaque base de données sont compilés dans trois tableaux différents 3.3, 3.4, et 3.5 pour CK+, BP4D et DISFA respectivement.

Comparaison sur l'ensemble de données CK+ Après une analyse des résultats présentés dans le tableau 3.3, plusieurs observations peuvent être faites. D'abord, le modèle proposé dans ce chapitre montre une performance inférieure en termes de score F1 moyen par rapport aux autres méthodes de l'état de l'art. En particulier, les AU telles que AU07 et AU09 montrent une nette sous-performance. Cette baisse de performance peut être

attribuée à plusieurs facteurs, qui pourraient inclure la qualité des données (par exemple, l'encodage utilisé sur les images étant un mélange de plusieurs émotions), la complexité du modèle, ou les défis intrinsèques associés à la détection de ces AU spécifiques.

TABLE 3.3 – Comparaison des scores F1 entre le modèle proposé et les méthodes de l'état de l'art en matière de détection des AU sur l'ensemble de données CK+. Les valeurs les plus élevées sont indiquées en gras tandis que les valeurs indisponibles sont représentées par un tiret (-).

Method /AUs	**JPML** [151]	**Simge [81]**	**Chen [101]**	**Chen [101]**	**Elef [58]**	**Modèle proposé**
AU01	0.90	**0.92**	0.85	0.87	0.82	0.77
AU02	**0.93**	0.86	0.88	0.86	0.86	0.76
AU04	-	**0.89**	0.80	0.82	0.79	0.74
AU05	-	-	0.74	0.74	0.73	**0.75**
AU06	0.74	**0.76**	0.70	0.68	0.72	0.74
AU07	0.66	**0.81**	0.61	0.55	0.57	0.37
AU09	-	0.87	**0.89**	**0.89**	0.87	0.20
AU12	0.80	0.80	**0.87**	**0.87**	**0.87**	0.75
AU15	-	**0.91**	-	-	0.76	0.71
AU17	0.83	-	**0.86**	0.84	**0.86**	0.22
AU20	-	-	-	-	**0.70**	0.39
AU23	-	-	0.45	0.32	**0.67**	0.46
AU24	-	-	0.46	-	**0.51**	-
AU25	-	-	**0.93**	0.43	0.91	-
AU26	-	-	-	0.71	0.21	**0.76**
AU27	-	-	0.89	-	**0.91**	-
AVG	0.81	**0.85**	0.76	0.73	0.73	0.59

Comparaison sur l'ensemble de données BP4D Le tableau 3.4 montre les résultats de comparaison entre le modèle proposé dans ce chapitre et les méthodes issues de l'état de l'art sur la détection des AU sur l'ensemble de données BP4D. Cette comparaison est illustrée en deux catégories principales : les modèles de grande taille ou *fully sized* modèles et les modèles légers, comme en témoigne la dernière ligne désignée par le "Type de modèle".

En observant les résultats présentés dans le tableau 3.4, il est clair que le modèle présenté dans ce chapitre affiche des scores F1 généralement inférieurs à ceux des autres

TABLE 3.4 – Comparaison des scores F1 entre le modèle proposé dans ce chapitre et les méthodes de l'état de l'art pour la détection des AU sur l'ensemble de données BP4D, catégorisées en deux groupes : les modèles de grande taille dans les quatre premières colonnes et modèles légers dans les cinq dernières colonnes. Les valeurs les plus élevées sont indiquées en gras, les secondes valeurs les plus élevées sont soulignées, tandis que les valeurs indisponibles et/ou non-fournies sont représentées par un tiret (-).

Mèthode /AUs	**Zhang [144]**	**ARL [97]**	**STRAL [99]**	**JAU [150]**	**IDEN [148]**	**MCFE [147]**	**LGR [146]**	**FS [145]**	**CWCF [149]**	**Modèle proposé**
AU06	0.76	0.77	0.77	-	0.77	**0.79**	0.78	0.77	**0.79**	0.52
AU10	0.82	0.84	0.83	-	0.83	0.80	**0.85**	0.83	0.84	0.50
AU12	0.87	0.87	0.88	-	0.85	0.88	0.88	0.87	**0.90**	0.67
AU14	0.60	0.62	0.60	-	0.63	0.64	**0.66**	0.61	0.62	0.44
AU17	**0.64**	0.60	0.63	-	0.61	0.60	0.50	0.63	0.63	0.56
AVG	0.73	**0.74**	**0.74**	-	0.73	**0.74**	0.73	**0.74**	0.53	0.69
Nombre Total de Param.	>138	>138	>26	>11	>6,572	>3	>4	8,19	26	**1,5**
Taille d'entrée	>256*256	200*200	200*200	256*256	128*128	170*170	176*176	256*256	224*224	**48*48**
Type de modèle	***Full sized* modèles**						**Modèles légers**			

modèles issus de la littérature. La moyenne générale du score F1 atteint 0.69 sur l'ensemble des AU considérées et se trouve non loin des valeurs moyennes des modèles utilisés pour la comparaison. D'un autre côté, une inspection attentive des valeurs pour chacune des AU cherchées dans l'ensemble de données BP4D révèle que le modèle proposé dans ce chapitre n'est pas compétitif ni au même niveau avec ces travaux de référence. Cependant, les performances moins compétitives par rapport à l'état de l'art peuvent s'expliquer en partie par un faible nombre de paramètres (1.5 Million) et l'utilisation des images d'entrée de petite taille (48×48).

Comparaison sur l'ensemble de données DISFA L'examen du tableau 3.5 indique que le modèle proposé dans ce chapitre se distingue nettement dans la détection des AU. Pour la majorité des AU, le modèle affiche soit le meilleur soit le deuxième meilleur résultat en termes de F1-score et de précision. De plus, il atteint ces performances très compétitives avec un nombre relativement bas de paramètres (1.5 million), et une taille d'images d'entrée minimale de 48×48 pixels.

TABLE 3.5 – Comparaison des scores F1 entre le modèle proposé dans ce chapitre et les méthodes de l'état de l'art pour la détection des AU sur l'ensemble de données DISFA, catégorisées en deux groupes : les modèles de grande taille dans les quatre premières colonnes et modèles légers dans les cinq dernières colonnes. Les valeurs les plus élevées sont indiquées en gras, les secondes valeurs les plus élevées sont soulignées, tandis que les valeurs indisponibles et/ou non-fournies sont représentées par un tiret (-).

Mèthode	F1-frame (in %)									Précision (in %)			
/ AUs	Zhang	ARL	STRAL	IDEN	MCFE	LGR	FS	CWCF	Ours	ARL	STRAL	CWCF	Ours
01	0.55	0.43	0.52	0.25	0.38	62	0.50	0.54	**0.83**	0.92	0.94	**0.96**	0.77
02	0.63	0.42	0.47	0.34	0.46	64	0.58	0.63	**0.80**	0.92	0.93	**0.97**	0.86
04	0.74	0.63	0.69	0.64	0.56	72	0.77	0.63	**0.90**	0.88	0.89	0.90	**0.92**
05	-	-	-	-	0.17	-	-	-	**0.83**	-	-	-	**0.73**
06	0.45	0.41	0.47	0.45	0.56	46	0.53	0.55	**0.92**	0.91	0.90	0.89	**0.73**
09	0.35	0.40	0.56	0.44	0.48	48	0.27	0.37	**0.89**	0.95	**0.96**	**0.96**	0.89
12	0.75	0.76	0.72	0.70	0.73	75	0.77	0.67	**0.91**	0.93	0.92	0.85	**0.91**
15	-	-	-	-	0.30	-	-	-	**0.87**	-	-	-	**0.87**
17	-	-	-	-	0.45	-	-	-	**0.89**	-	-	-	**0.84**
20	-	**0.95**	-	-	0.16	-	-	-	0.92	-	-	-	**0.94**
25	0.93	0.66	0.91	0.81	0.79	0.94	**0.95**	0.89	0.92	**0.97**	0.94	0.93	0.89
26	0.54	0.58	0.67	0.55	0.59	73	0.56	0.57	**0.94**	**0.94**	**0.94**	0.84	0.89
AVG	0.62	0.74	0.63	0.52	0.47	67	0.62	0.61	**0.90**	**0.93**	**0.93**	0.91	0.89
Params	>138	>138	>26	>6,572	>3	>4	8,19	26	**1,5**	>138	>26	26	**1,5**
Input Size	>256	200p	200p	128p	170p	176p	256p	224p	**48p**	200p	200p	224p	**48p**
Type de modèle	**Full sized modèles**			**Modèles légers**						**P = Pixels**			

Comparaison du modèle proposé avec l'état de l'art en reconnaissance d'expressions faciales sur CK+, FER2013+ et RAFdb.

Dans un premier temps, nous montrons les résultats de classification par émotion pour pouvoir comparer le modèle proposé dans ce chapitre avec le modèle initial du chapitre précédent ainsi que les modèles de l'état de l'art. Ces résultats sont affichés dans les tableaux 3.6, 3.7 et 3.8 pour les ensembles de données RAFdb, FER2013+ et CK+ de manière respective.

Dans le tableau 3.6, une comparaison entre le modèle basé sur les AU et proposé dans ce chapitre et diverses méthodes de l'état de l'art en FER sur le jeu de données RAFdb est présentée. Notre modèle atteint une précision de 83.00% avec un nombre minimal de paramètres (1.5M). Bien que l'approche ECAN [137] affiche la meilleure précision de 89.77%, elle ne fournit pas d'information sur le nombre de paramètres utilisés. Par ailleurs, la méthode SSA-ICL [5] atteint une précision comparable de 89.44% mais avec un nombre beaucoup plus élevé de paramètres (11M). Des méthodes comme DNFER [136] montrent une précision nettement inférieure à 60.41%, sans indication du nombre de paramètres utilisés. Notre modèle, malgré une précision légèrement inférieure, offre l'avantage d'une

complexité moindre en termes de paramètres utilisés.

TABLE 3.6 – Comparaison en termes de précision entre le modèle proposé et basé sur les AU et les méthodes de l'état de l'art en reconnaissance des expressions faciales sur le jeu de données RAFdb. Les valeurs les plus élevées sont indiquées en gras tandis que les valeurs indisponibles sont représentées par un tiret (-).

Modèles	**Précision (en %)**	**Paramètres (M)**
ECAN [137]	**89.77**	-
SSA-ICL [5]	89.44	11
NCCTFER [139]	87.97	-
Xiaoyu et al. [138]	87.58	-
Muhamad et al. [133]	84.91	2
A-MobileNet [134]	84.49	3.4
Le modèle proposé	83.00	**1.5**
LibreFace [135]	82.79	43
Modèle initial (chapitre 2)	60.00	**1.5**
DNFER [136]	60.41	-

Dans le tableau 3.7, les résultats de comparaison sur le jeu de données FER2013+ sont présentés. Nous constatons que le modèle proposé dans ce chapitre atteint une précision maximale de 81.4%. Bien que ce chiffre soit inférieur aux autres méthodes listées dans le tableau 3.7, il est important de noter que notre modèle utilise un nombre de paramètres limité évalué à 1.5 million. Cela représente un avantage significatif en termes de complexité globale du modèle et de ressources de calcul nécessaires pour son implantation. Cependant, les améliorations sont nécessaires pour explorer des méthodes permettant d'augmenter davantage la précision globale sans augmenter considérablement le nombre de paramètres nécessaires.

Conformément aux résultats présentés dans le tableau 3.8, le modèle proposé basé sur l'intégration d'AU et proposé dans ce chapitre affiche une précision de 94.34% sur la base de données CK+. Bien que cette performance ne soit pas la plus élevée en comparaison avec d'autres méthodes telles qu'eXnet [132] et SACNN-LSTM [131], elle reste néanmoins compétitive. Il est important de souligner que notre modèle se situe dans la fourchette supérieure des performances, surpassant des travaux antérieurs notables comme ceux de Sun et al. [126] et DIA [127].

TABLE 3.7 – Comparaison en termes de précision entre le modèle proposé dans ce chapitre et les méthodes de reconnaissance des expressions faciales de l'état de l'art sur l'ensemble de données FER2013+. Les valeurs les plus élevées sont indiquées en gras tandis que les valeurs indisponibles sont représentées par un tiret (-).

Modèles	Précision (en %)	Paramètres (M)
FST-MWOS [140]	**90.41**	-
DNFER [136]	89.32	-
SUNYOUNG [141].	88.45	-
NCCTFER [139]	88.21	-
A-MobileNet [134]	88.11	3.4
Le modèle proposé	81.40	**1.5**
Modèle initial (chapitre 2)	24.00	**1.5**

TABLE 3.8 – Comparaison en termes de précision entre le modèle proposé dans ce chapitre et les méthodes de l'état de l'art pour la reconnaissance des expressions faciales sur la base de données CK+. Les valeurs les plus élevées sont indiquées en gras.

Méthode	Précision (en %)
eXnet [132]	**96.75**
SACNN-LSTM [131]	95.15
Shahid et al [130]	94.90
Le modèle proposé	94.34
Modèle initial (chapitre 2)	94.00
Elfatih et al [129]	92.73
Zhang et al [128]	92.73
DIA [127]	89.51
Sun et al [126]	87.20

Cette haute précision, atteinte avec une architecture CNN relativement standard, démontre l'efficacité de notre approche dans la reconnaissance des expressions faciales sur la base de données CK+. Il serait donc pertinent d'explorer des améliorations potentielles pour combler l'écart de performance avec les méthodes les plus avancées, tout en conservant la simplicité du modèle.

3.5.3 Interprétabilité avec Grad-CAM

La méthode Grad-CAM (*Gradient-weighted Class Activation Mapping*) est une technique d'interprétation de modèle d'apprentissage profond qui est également appliquée

dans le domaine de la FER pour expliquer visuellement les prédictions faites par un modèle [113]. Dans ce contexte, nous avons utilisé le Grad CAM pour afficher les activations associées à chaque AU dans une image donnée, offrant ainsi une visualisation directe des régions du visage qui contribuent le plus à la détection des différentes AU.

Le Grad-CAM utilise les gradients des scores de classe, par rapport aux cartes de caractéristiques d'une couche donnée, pour produire une carte de chaleur qui indique les régions importantes pour une prédiction de classe donnée. Mathématiquement, l'utilisation de Grad-CAM peut être formulée comme suit :

$$L_{\text{Grad-CAM},c} = \text{ReLU}\left(\sum_k \alpha_k^c \cdot A^k\right) \tag{3.4}$$

où $L_{\text{Grad-CAM},c}$ est la carte de chaleur pour la classe c, ReLU est la fonction d'activation rectifiée linéaire, α_k^c est le poids de la carte d'activation k pour la classe c, et A^k représente les cartes d'activation de la couche considérée. Ces poids α_k^c sont calculés comme étant la moyenne du gradient de la sortie de classe c par rapport à chaque canal de la carte de caractéristiques A^k.

Ainsi, en prenant en compte les AU spécifiques pendant le processus d'entraînement, il est possible d'obtenir des cartes d'activation qui mettent en évidence les régions du visage qui sont les plus pertinentes pour chaque AU individuelle, facilitant ainsi la compréhension et l'interprétation des prédictions du modèle (Pour plus de détails, voir l'Annexe B.1.1).

Résultats sur le jeu de données CK+ : Les résultats sur le jeu de données CK+ présentés dans la figure 3.8 montrent que les AU détectées sont bien représentées. Cependant, il faut noter que les intensités ne sont pas très claires, ce qui suggère une certaine difficulté à distinguer les différentes intensités d'activation des AU. Bien que les régions pertinentes soient identifiées, une amélioration de la clarté pourrait faciliter l'interprétation des résultats.

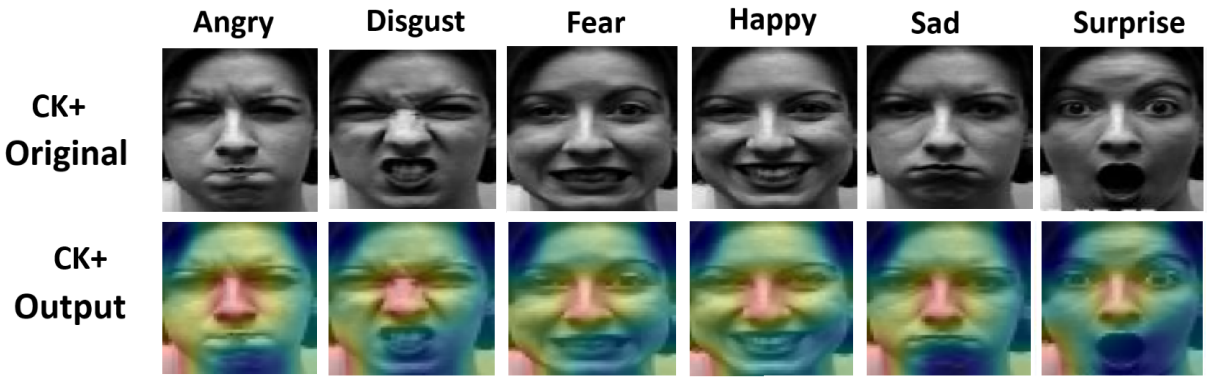


FIGURE 3.8 – Les résultats de sortie de la méthode Grad-CAM pour l'ensemble de données CK+.

Résultats de visualisation sur le jeu de données RAFdb : Dans le cas du jeu de données RAFdb, comme illustré dans la figure 3.9, nous pouvons observer que les AU sont affichées de manière satisfaisante et sont présentées séparément. Cette séparation distincte des AU permet une analyse plus facile et plus détaillée des régions faciales activées, mettant en évidence la capacité du modèle à discriminer efficacement entre les différentes unités d'action lors de la prédiction des émotions.

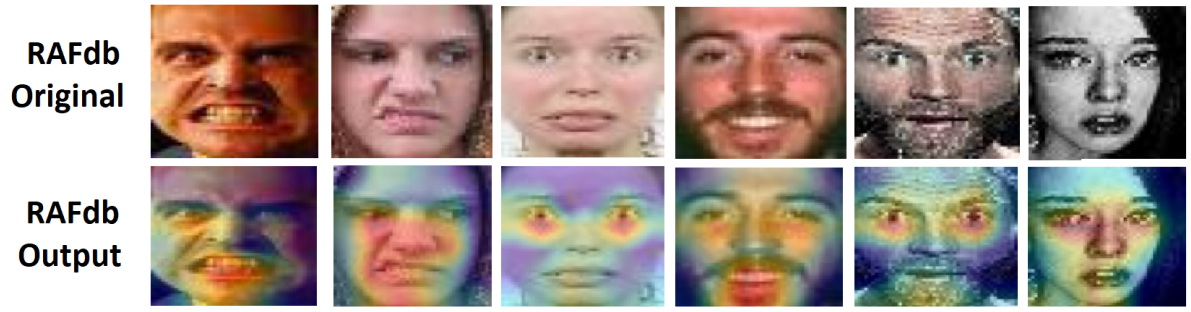


FIGURE 3.9 – Les résultats de sortie de la méthode Grad-CAM pour l'ensemble de données RAFdb

Résultats de visualisation sur le jeu de données FER2013+ : En se référant à la figure 3.10, nous constatons que certains AU ne sont pas détectés par le modèle sur le jeu de données FER2013+. De plus, il y a des intensités observées dans des parties non pertinentes de l'image, indiquant ainsi que le modèle se concentre également sur des

régions non pertinentes, ce qui n'est pas un résultat attendu. Cela suggère que le modèle pourrait bénéficier d'un ajustement pour éviter les fausses activations.

FIGURE 3.10 – Les résultats de sortie de la méthode Grad-CAM pour l'ensemble de données FER2013+

Résultats de visualisation sur le jeu de données BP4D : Pour le jeu de données BP4D, dont les résultats sont présentés dans la figure 3.11, bien que les AU ne soient pas clairement séparées les unes des autres, il est à noter que les cartes de chaleur sont bien situées sur les régions faciales. Cela implique que le modèle est assez précis pour localiser les zones pertinentes du visage, même si la distinction entre les AU individuelles reste un champ d'améliorations possibles.

FIGURE 3.11 – Les résultats de sortie de la méthode Grad-CAM pour l'ensemble de données BP4D.

Résultats de visualisation sur le jeu de données DISFA : Finalement, dans le cas du jeu de données DISFA visualisé dans la figure 3.12, la majorité des images d'émotion

présentent bien les AU détectées. Toutefois, quelques unes présentent une absence de détection de certains AU, indiquant ainsi qu'il reste des améliorations à apporter pour assurer une détection complète et précise des AU dans toutes les images.

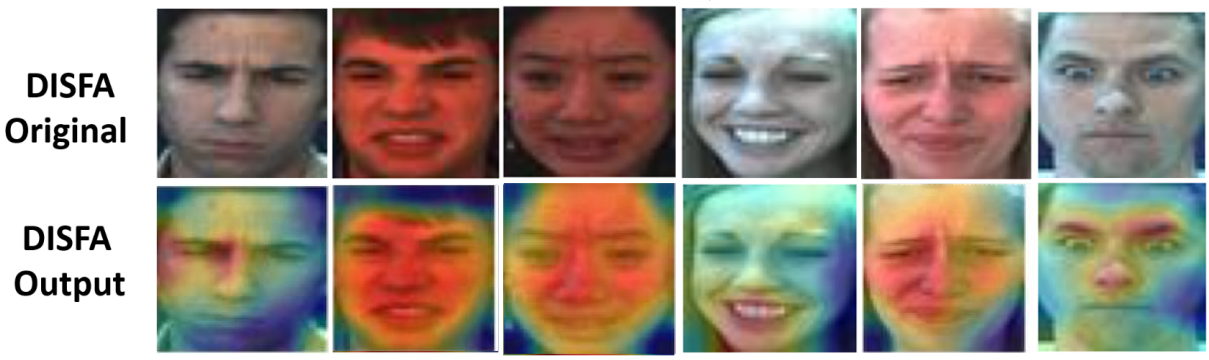


FIGURE 3.12 – Les résultats de sortie de la méthode Grad-CAM pour l'ensemble de données DISFA.

En conclusion, l'analyse des résultats montre qu'il y a encore une marge d'amélioration pour optimiser la sortie du modèle proposé dans ce chapitre. Les différentes bases de données présentent des défis uniques et il est clair que des travaux supplémentaires sont nécessaires pour affiner le modèle et obtenir les meilleurs résultats possibles.

3.5.4 Interprétation des résultats

En analysant les performances du modèle proposé dans ce chapitre sur les bases de données CK+, BP4D et DISFA pour la détection des AU, ainsi que sur les bases de données CK+, RAFdb et FER2013+ pour la FER, il est clair qu'une amélioration significative a été réalisée par rapport au modèle initial présenté dans le chapitre précédent.

Concernant la détection des AU sur les images de la base de données CK+, bien que le score F1 moyen soit de 59%, ce qui est relativement faible, notre modèle se distingue en obtenant les meilleurs scores dans la détection des AU 05 et 26. En contrastant avec les performances sur la base de données BP4D, le score F1 moyen atteint 69%, n'étant toutefois pas le meilleur parmi les travaux issus de l'état d'art utilisés pour la comparaison. Notons une remarquable performance sur la base de données DISFA où un score F1 moyen

de 94% a été obtenu, surpassant la majorité des travaux de recherche issus de l'état d'art utilisés pour la comparaison dans la détection de la plupart des AU.

En analysant les résultats pour la reconnaissance des expressions faciales sur CK+, il est observé, comme indiqué dans le Tableau 3.8, que notre modèle maintient une position compétitive avec une précision de 94.34%, attestant de l'efficacité de l'intégration de l'encodage binaire des AU dans l'amélioration des performances du modèle.

Les tableaux 3.6 et 3.7, illustrant les performances sur les ensembles de données RAFdb et FER2013+ respectivement, mettent en lumière l'avantage du modèle proposé dans ce chapitre quant à sa faible complexité en termes de nombres de paramètres, tout en maintenant des niveaux satisfaisants de précision, et indiquant que le modèle amélioré proposé dans ce chapitre représente une base solide pour des développements futurs.

En dépit de ces avancées, une attention particulière doit être accordée à l'amélioration continue du modèle pour obtenir une performance optimale, envisageant une efficacité accrue dans les itérations futures grâce à un score F1 moyen plus élevé et une précision améliorée.

3.6 Conclusion

Dans ce chapitre, nous avons étendu notre exploration des réseaux CNN dans le domaine de la détection des AU. En utilisant une entrée composée d'images aux côtés de leur représentation binaire codée en AU, le modèle proposé dans ce chapitre a montré une amélioration significative des performances par rapport à sa version initiale présenté dans le chapitre précédent. Nous avons également remplacé la méthode LIME par Grad-CAM pour la visualisation des cartes thermiques, apportant une amélioration considérable à la qualité de la représentation visuelle des AU détectées.

Nos expérimentations ont inclus les ensembles de données CK+, RAFdb, FER2013+, BP4D et DISFA, ce qui a permis de valider l'efficacité de notre modèle dans des contextes

variés. Cette diversité d'ensembles de données contribue également au renforcement de la robustesse des modèles proposés et à leur capacité de généraliser les résultats.

L'introduction de la fonction de perte d'entropie croisée pondérée a servi à atténuer les effets des jeux de données déséquilibrés, un impact positif qui a été observé dans la distance euclidienne, notre métrique d'évaluation. Cependant, bien que les performances aient été améliorées, elles restent insatisfaisantes et nécessitent des améliorations supplémentaires.

Dans la perspective d'une amélioration continue, le prochain chapitre introduira des mécanismes d'attention dans le modèle proposé et amélioré dans ce chapitre. Ces mécanismes sont reconnus pour leur aptitude à focaliser l'attention du modèle sur des caractéristiques d'entrée plus informatives, ce qui devrait apporter une contribution significative à la détection et à l'analyse plus précises des AU.

Chapitre 4

Modèle CNN avec le mécanisme d'attention

Sommaire

4.1 Introduction

Dans les chapitres précédents, nous avons élaboré et évalué des modèles basés sur une architecture CNN simplifiée pour la détection des expressions faciles et les unités d'action AU. Bien que ces modèles aient démontré des performances respectables avec une complexité faible par rapport à l'état de l'art, des marges d'amélioration en comparaison avec les méthodes les plus avancées de la littérature subsistent. C'est dans cette optique que le présent chapitre vise à introduire des mécanismes d'attention au sein des architectures et modèles présentés dans les chapitres précédents.

Plus précisément, nous intégrerons des modules d'attention spatiale et par canal, reconnus pour leur efficacité à orienter un modèle d'apprentissage profond vers les caractéristiques les plus informatives des données d'entrée utilisées. Cette modification ciblée des modèles utilisés a pour but d'affiner la détection et l'analyse des AU, tout en conservant l'avantage de la complexité faible des modèles initiaux, et par conséquent, l'avantage de leur légèreté computationnelle qui les caractérise.

4.2 Mécanisme d'attention

4.2.1 Principe

Les mécanismes d'attention sont des techniques de traitement de l'information conçues pour optimiser la manière dont un modèle de *machine learning* interprète les données d'entrée. Le principe fondamental de ces mécanismes d'attention est de pondérer différentes zones de l'entrée afin de permettre au modèle de se "concentrer" sur les segments les plus pertinents pour une tâche donnée. Pour une explication plus détaillée des mécanismes d'attention, nous renvoyons le lecteur à l'annexe correspondante C.1.

4.2.2 Application des mécanismes d'attention dans la FER

Ces techniques ont été appliquées avec succès dans les systèmes de FER pour améliorer les performances en se concentrant sur les zones les plus importantes du visage pour détecter les expressions émotionnelles [152]. Dans les systèmes FER, les mécanismes d'attention peuvent être intégrés à différents niveaux du réseau de neurones, tels que les couches de convolution, les couches entièrement connectées ou les couches de décision. Par exemple, une approche récente proposée dans la littérature a utilisé un réseau de neurones convolutif avec des mécanismes d'attention pour détecter les émotions en se concentrant sur les régions les plus pertinentes du visage [153].

Les mécanismes d'attention peuvent également être utilisés pour améliorer l'interprétabilité des systèmes FER en identifiant les zones du visage qui sont les plus importantes pour détecter les expressions émotionnelles. Par exemple, une étude récente a utilisé des mécanismes d'attention pour déterminer les parties les plus significatives du visage des patients étant à l'origine des expressions de douleur [118].

Les mécanismes d'attention ont également été appliqués avec succès dans la détection des AU et la FER. Ces mécanismes permettent de cibler des mouvements musculaires spécifiques et associés à des expressions faciales émotionnelles [79, 28]. Les systèmes de détection des AU basés sur des réseaux neuronaux convolutifs CNN utilisent également des mécanismes d'attention pour identifier les zones du visage les plus pertinentes pour faciliter la détection des AU. Un exemple d'une telle approche est présenté dans [119], où les résultats obtenus sont très prometteurs par rapport aux approches n'intégrant pas ces mécanismes d'attention.

Des mécanismes d'attention peuvent également être utilisés pour améliorer les performances des systèmes de détection des AU dans des conditions de faible qualité d'image ou de bruit. Une étude récente a utilisé des mécanismes d'attention pour optimiser les performances de détection des AU dans des images [154] basse résolution et bruitées.

L'intégration de multiples mécanismes d'attention dans des modèles légers de FER

offre une voie prometteuse pour l'élaboration des systèmes à la fois plus efficaces et polyvalents [79]. Par exemple, le couplage de mécanismes d'attention spatiale et par canal peut permettre l'identification plus précise de régions et de caractéristiques du visage pertinentes. Lorsque ces mécanismes sont appliqués à des modèles FER légers tels que MobileNet ou SqueezeNet, cela conduit au développement de systèmes FER capables de fonctionner de manière optimale sur des plateformes diverses telles que des appareils mobiles ou des systèmes embarqués dans les véhicules [28].

En outre, l'exploration et l'expérimentation de techniques d'attention innovantes, comme l'attention auto-régressive ou l'attention fondée sur des mécanismes de mémoire dynamique, pourraient également constituer des avancées significatives dans le domaine. Ces techniques pourraient augmenter la robustesse des systèmes FER face à des défis tels que les variations d'éclairage, les occlusions faciales partielles, ou encore les images de faible qualité [155].

4.3 Modèle avec le mécanisme d'attention proposé

4.3.1 Aperçu

Dans cette section, nous présentons notre modèle amélioré pour la détection des AU, illustré en Figure 4.1. Celui-ci conserve les principes fondamentaux du modèle CNN précédemment décrit, tout en y apportant quelques modifications majeures pour accroître son efficacité.

Au lieu d'un seul bloc de couches convolutives, nous avons segmenté notre architecture en deux blocs de convolution distincts. Ce choix a été guidé par notre volonté d'améliorer l'identification des caractéristiques pertinentes à différents niveaux de la chaîne de traitement.

Chaque bloc convolutif est suivi d'un bloc d'attention, dont l'objectif est d'augmenter la focalisation de l'apprentissage sur les régions les plus pertinentes de l'image pour la

détection des AU. Ce mécanisme d'attention, dont une explication détaillée est fournie dans l'annexe C.1, vise à ajouter une dimension supplémentaire de sélectivité à notre modèle, dans le but d'augmenter sa précision.

Comme les modèles décrits dans les chapitres précédents, le modèle utilise cinq couches de convolution, chacune suivie d'une couche de *MaxPooling*. Les couches de *Dropout* et de normalisation par lots (*BatchNormalization*) ont été maintenues pour éviter le sur-apprentissage et accélérer l'apprentissage. Le choix du nombre de filtres et de leur taille reste le même que dans le modèle précédent, pour préserver un équilibre entre la complexité du modèle et ses performances.

Après les couches de convolution et d'attention, le modèle intègre des couches entièrement connectées (*Dense*) pour réaliser la transformation finale des caractéristiques spatiales en probabilités de présence des différentes AU. Cette couche "dense" ayant une fonction d'activation de type *sigmoide*, est adaptée au nombre d'AU définies par le jeu de données utilisé pour l'entraînement.

Enfin, pour la visualisation et l'interprétation de nos résultats, nous avons maintenu l'utilisation de la technique Grad-CAM. Celle-ci offre une représentation visuelle des régions de l'image les plus influentes pour la prédiction de chaque AU, facilitant ainsi l'interprétation des résultats du modèle.

4.3.2 Mécanisme d'attention

Dans la méthode proposée, deux blocs d'attention sont incorporés pour attribuer des poids d'importance aux vecteurs d'entrée. Chaque bloc est composé d'un module d'attention de canal et d'un module d'attention spatiale, comme proposé dans [156]. Le module d'attention de canal fournit les poids pour chaque canal de la carte de caractéristiques, tandis que le module d'attention spatiale attribue les poids pour chaque position dans la carte de caractéristiques. Ces blocs d'attention permettent au modèle de se concentrer uniquement sur les caractéristiques pertinentes pour la détection des AU faciales,

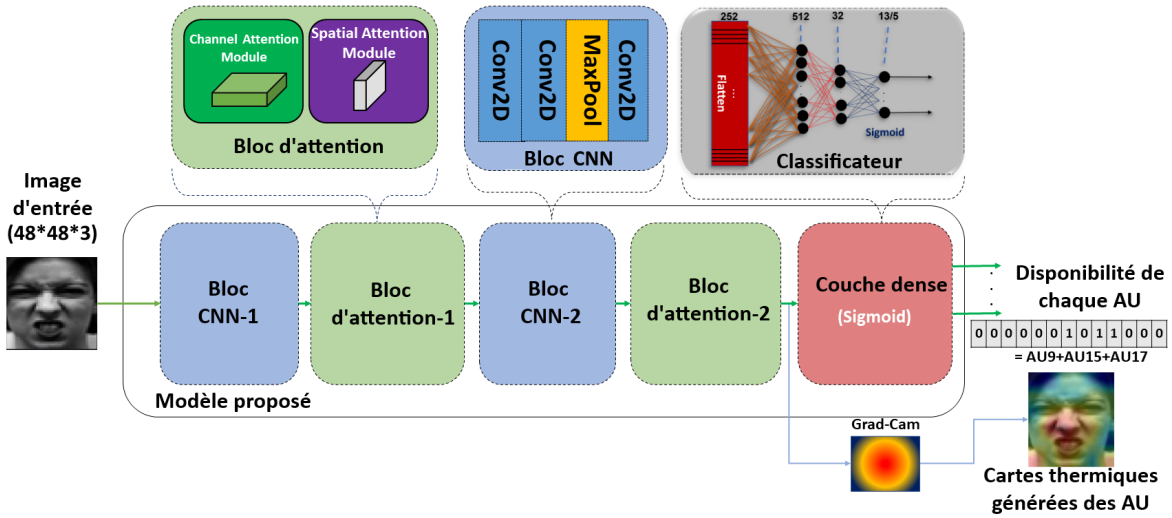


FIGURE 4.1 – Schéma synoptique de l'architecture CNN intégrant le mécanisme d'attention pour la détection des AU. Elle comprend cinq couches convolutives, des couches de regroupement, d'abandon et de normalisation, et utilise la méthode Grad-CAM pour la visualisation des AU.

améliorant ainsi les performances du modèle.

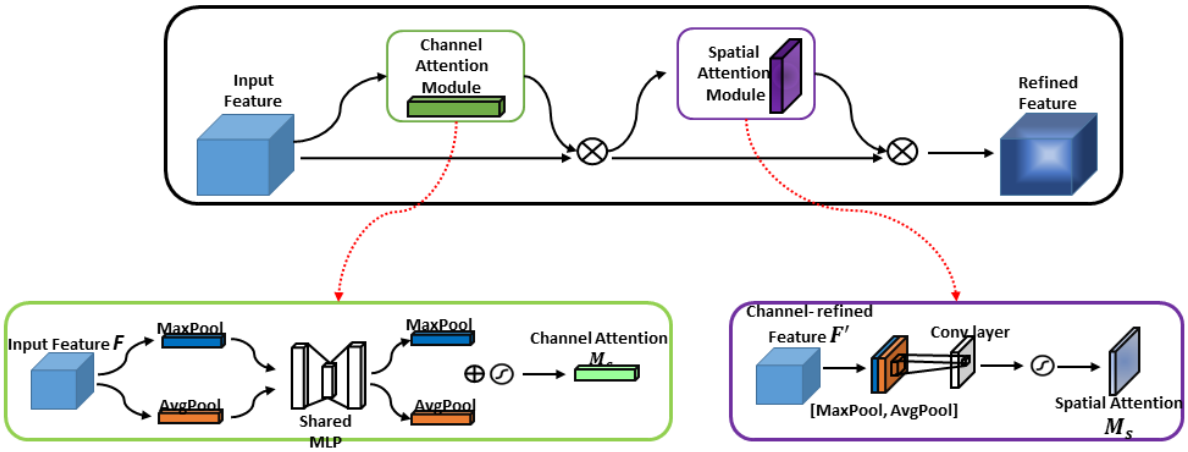


FIGURE 4.2 – Bloc d'attention dans le modèle proposé conçu à l'aide des modules de canal et spatiaux : le module de canal et le module spatial utilisent les deux des couches entièrement connectées combinées à des couches de *pooling* moyennes et maximales pour déterminer les poids de chaque carte de caractéristiques et calculer le poids de chaque position spatiale, respectivement (basé sur le travail de Woo et al. [156]).

Attention de canal L'attention de canal (voir l'équation 4.1) fait référence à un mécanisme d'attribution de poids aux différents canaux (ou caractéristiques) du tenseur d'entrée dans le modèle. L'idée derrière l'attention de canal est d'utiliser la relation inter-canaux entre les caractéristiques d'entrée pour apprendre un poids pour chaque canal. Ce poids est utilisé pour ajuster l'importance des activations de chaque canal, permettant au modèle de se concentrer sur les canaux les plus pertinents pour une tâche donnée. Pour mettre en œuvre l'attention de canal, le tenseur d'entrée (F) est d'abord traité avec du pooling maximal et moyen, produisant deux sorties M et A, respectivement. Ces sorties sont ensuite passées à travers un perceptron multicouche (MLP) avec des poids partagés, produisant deux nouvelles sorties M' et A'. Enfin, ces deux sorties sont passées à travers la fonction sigmoïde, produisant les poids d'attention de canal C. Ces poids sont ensuite utilisés pour mettre à l'échelle les activations du tenseur d'entrée F avant d'être transmis à la couche suivante du modèle.

$$C = \sigma(MLP(M, A)) \tag{4.1}$$

Dans l'équation 4.1, C représente les poids d'attention de canal, M et A représentent les sorties des opérations de pooling maximal et moyen appliquées au tenseur d'entrée F, et σ représente la fonction d'activation sigmoïde. Le MLP prend les sorties des opérations de pooling maximal et moyen et produit les poids d'attention de canal.

Attention spatiale L'attention spatiale fait référence à un mécanisme d'attribution de poids aux différentes positions spatiales (par exemple, pixels ou voxels) dans un tenseur d'entrée d'un réseau de neurones (voir l'équation C.6). L'idée derrière l'attention spatiale est d'utiliser la relation inter-spatiale entre les positions d'entrée pour apprendre un poids pour chaque position. Ce poids est utilisé pour mettre à l'échelle les activations de chaque position, permettant au modèle de se concentrer sur les positions les plus pertinentes pour une tâche donnée. Pour mettre en œuvre l'attention spatiale, le tenseur d'entrée F

est d'abord traité avec du pooling maximal et moyen pour produire deux sorties M et A. Ces sorties sont ensuite passées à travers une couche convolutionnelle f et la fonction sigmoïde, produisant les poids d'attention spatiale S. Ces poids sont ensuite utilisés pour mettre à l'échelle les activations du tenseur d'entrée F avant d'être transmis à la couche suivante du modèle. Le processus complet du mécanisme d'attention spatiale est illustré avec l'équation 4.2 :

$$S = \sigma(f^{7\times7}(M_c(F))) \tag{4.2}$$

où M_c représente la sortie de l'opération d'attention de canal appliquée au tenseur d'entrée, $f^{7\times7}$ désigne une couche convolutionnelle 7x7 et σ représente la fonction d'activation sigmoïde.

4.4 Prétraitement des données

Comme dans le chapitre précédent (voir Section 3.3), la préparation des données reste un élément clé pour une validation réussie du modèle. Ce processus a été explicitement décrit dans le chapitre précédent et reste pratiquement inchangé dans le contexte du nouveau modèle, qui intègre des mécanismes d'attention.

Le tableau 4.1 ci-dessous résume les principales étapes de ce prétraitement.

TABLE 4.1 – Résumé des étapes de prétraitement des données pour la détection des AU.

Étape de prétraitement	**Objectif**
Normalisation des images	Uniformisation des valeurs des pixels
Équilibrage des classes	Éviter les biais de classe
Augmentation des données	Améliorer la robustesse du modèle
Codage binaire des AU	Faciliter l'apprentissage et la prédiction

4.5 Architecture et l'entraînement du modèle

Au niveau de l'architecture, le modèle basé sur les mécanisme d'attention comprend toujours plusieurs couches convolutives et des couches de regroupement maximal (*Max-Pooling*) pour réduire les dimensions spatiales. Cependant, nous avons décomposé le dernier modèle en deux blocs de CNN et placé un bloc de mécanisme d'attention après chaque bloc CNN. Le mécanisme d'attention permet au modèle de se concentrer sur les caractéristiques les plus pertinentes pour la tâche de détection des AU, améliorant ainsi la précision du modèle.

En ce qui concerne le processus d'entraînement, nous utilisons toujours une approche d'apprentissage supervisé avec une fonction de coût basée sur l'entropie croisée binaire. Cette fonction de coût influence les erreurs faites sur les prédictions des AU de manière proportionnelle à l'importance de ces erreurs. Nous avons également conservé l'utilisation de techniques comme l'abandon (*Dropout*) et la normalisation par lots (*BatchNormalization*) pour améliorer la robustesse et l'efficacité de l'apprentissage.

Du point de vue de la visualisation et l'interprétation du modèle, nous continuons à utiliser la technique Grad-CAM pour générer une "carte thermique" des régions de l'image qui ont été les plus importantes pour la prédiction de chaque AU.

4.5.1 Réglage des hyperparamètres

Semblable à notre approche précédente, l'ajustement des hyperparamètres reste une étape cruciale pour optimiser la performance de notre modèle. Cependant, étant donné les modifications apportées à l'architecture du modèle, y compris l'ajout de blocs de mécanisme d'attention, certains hyperparamètres supplémentaires doivent être ajustés.

Ces hyperparamètres comprennent notamment le nombre de filtres dans les couches convolutives, le taux d'abandon dans les couches *Dropout*, le taux d'apprentissage de l'optimiseur, et les paramètres spécifiques aux blocs d'attention. Le processus d'ajustement

implique de trouver le bon équilibre entre la performance du modèle et la prévention de l'apprentissage excessif ou sur-apprentissage (*overfitting*), tout en maintenant une complexité de modèle faible.

4.5.2 Procédure d'entraînement

La procédure d'entraînement de notre nouveau modèle basé sur le mécanisme d'attention demeure en grande partie inchangée par rapport à celle du chapitre précédent. Pour ce faire, nous utilisons une fonction de coût fondée sur l'entropie croisée pondérée et la distance euclidienne.

Toutefois, il convient de souligner que l'intégration des blocs d'attention n'a pas substantiellement modifié cette procédure d'entraînement. Bien que ces modules exigent un apprentissage spécifique pour focaliser le modèle sur les zones d'image les plus pertinentes, le processus d'entraînement reste globalement le même.

L'entraînement se poursuit sur plusieurs époques jusqu'à une stabilisation de la performance sur l'ensemble de validation. Au cours de cette période, diverses métriques telles que la précision, le rappel, le score F1 et la perte d'entropie croisée sont surveillées pour évaluer les performances du modèle.

En résumé, bien que l'introduction de blocs d'attention ajoute une dimension supplémentaire au modèle, la procédure d'entraînement globale reste fidèle à celle du modèle initial, requérant toutefois une attention supplémentaire pour garantir un apprentissage efficace.

4.6 Critères d'évaluation et analyses comparatives

Pour une analyse complète du modèle suggéré, divers indicateurs ont été examinés. Ces indicateurs englobent la taille du modèle, des fonctions objectifs telles que l'entropie croisée ajustée et la métrique euclidienne, ainsi que différentes mesures de succès comme

la courbe de perte, le score F1 pour les méthodes de détection des AU, la précision pour les algorithmes de FER, et les résultats issus de Grad-CAM.

4.6.1 Taille du modèle

La taille du modèle est un indicateur essentiel qui influe sur plusieurs dimensions, notamment la vitesse d'apprentissage et les ressources de stockage requises. Il est impératif de prendre en considération non seulement la performance du modèle, mais également son encombrement mémoire. À cet égard, le tableau 4.2 met en évidence une comparaison pertinente entre notre proposition, nommée Attention-CNN, et d'autres modèles de l'état de l'art.

Conformément au tableau 4.2, il est clair que notre modèle se démarque en matière de taille. Avec seulement 1.5 million de paramètres, Attention-CNN est significativement plus compact par rapport aux autres modèles évalués. Par exemple, les modèles basés sur VGGNet comme Zhang [144] et ARL [97] dépassent les 138 millions de paramètres. De même, des modèles comme FSNet [145] et CWCF [149] nécessitent plus de 8 et 26 millions de paramètres respectivement.

TABLE 4.2 – Comparaison de la taille du modèle proposé avec des travaux de l'état de l'art (moins c'est mieux). La valeur en gras est la plus basse.

Méthode	**Infrastructure**	**Paramètres (en millions)**
Modèle proposé	Attention-CNN	**1.5**
MCFE [147]	DENSnet-121	>3
LGRNet [146]	BiLSTM	>4
IDENnet [148]	LightCNN	>6.572
FSNet [145]	ResNet-50(customisé)	8,19
JAU [150]	Resnet-18	>11
CWCF [149]	ResNet-9	>26
Zhang [144]	VGGNet	>138
ARL [97]	VGGNet	>138
STRAL [99]	VGGNet	>138

Ce contraste dans la taille des modèles est particulièrement significatif lorsque l'on

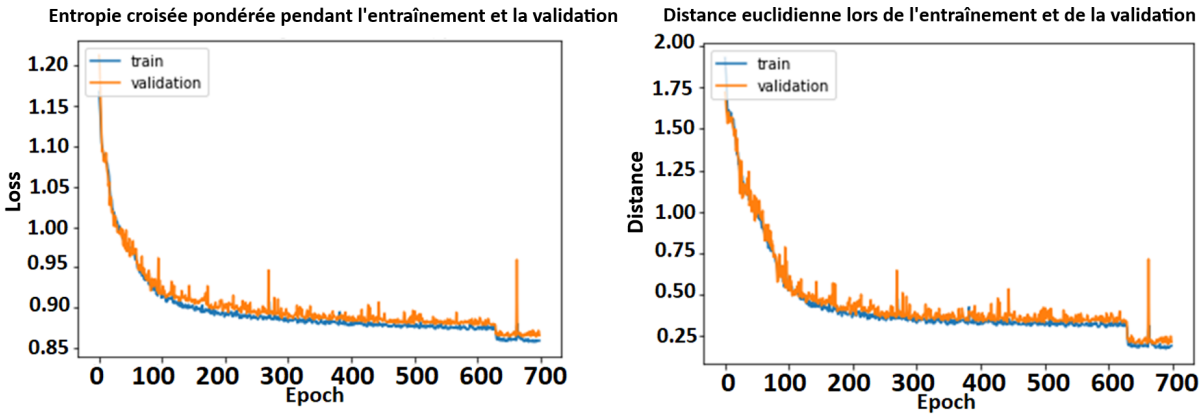
considère des applications nécessitant des ressources de stockage limitées ou une vitesse d'apprentissage accélérée. Le modèle Attention-CNN se positionne donc comme une option viable pour des déploiements dans des environnements contraints en ressources.

4.6.2 Diagrammes de perte

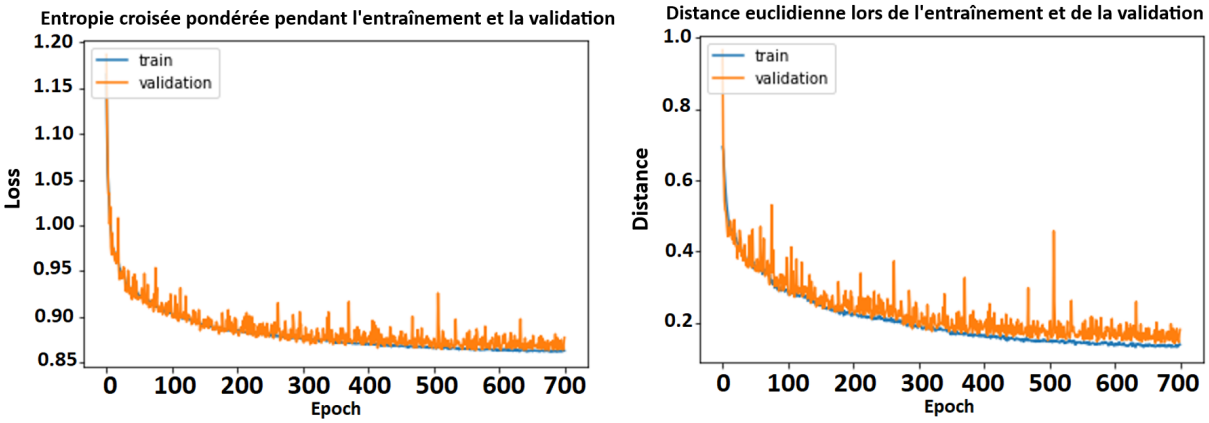
Dans cette sous-section, nous présentons des analyses basées sur les diagrammes de perte, qui servent à évaluer l'efficacité de l'apprentissage du modèle. Nous avons organisé cette sous-section en deux parties : les jeux de données collectés en laboratoire (« in-the-lab ») et ceux obtenus en environnement non contrôlé (« in-the-wild »).

Diagrammes de perte pour les jeux de données en laboratoire (« in-the-lab ») Dans ces jeux de données produits dans les conditions contrôlées en laboratoire, comme le démontrent les diagrammes de perte présentés à la Figure 4.3, le modèle affiche des performances très satisfaisantes. Aucun signe de sur- (*overfitting*) ou de sous-apprentisage (*underfitting*) n'a été observé lors de l'utilisation de la fonction de perte en entropie croisée pondérée et de la distance euclidienne comme métriques d'évaluation.

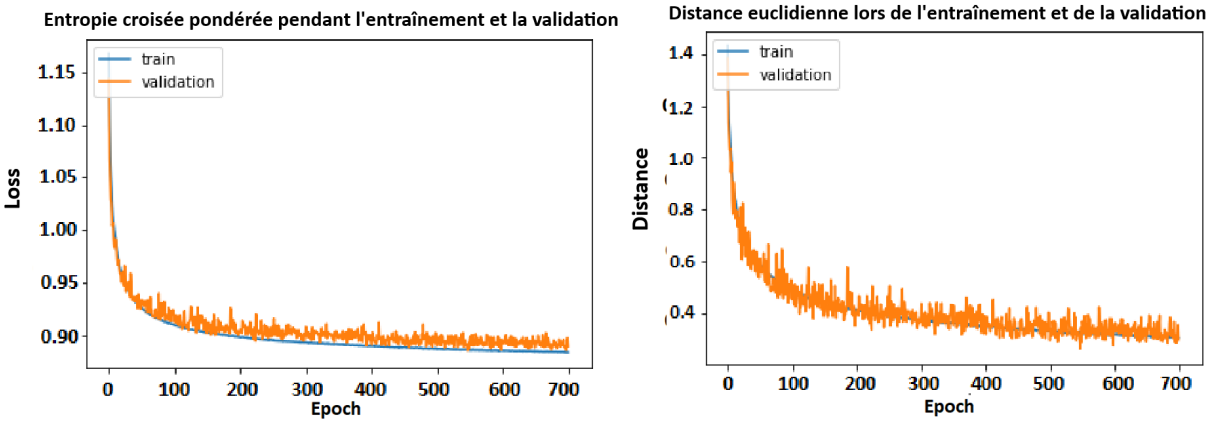
Diagrammes de perte pour les jeux de données en environnement non contrôlé (« in-the-wild ») Dans le cas des jeux de données produits dans les environnements non-contrôlés ou « sauvages », comme RAFdb et FER2013+, dont les résultats sont présentés à la Figure 4.4, des comportements divergents peuvent être observés. Pour RAFdb, des événements de type décrochage de la courbe d'apprentissage ont été détectés après un certain nombre d'époques. Néanmoins, une amélioration notable de la courbe d'apprentissage de ce nouveau modèle intégrant les mécanismes d'attention est à signaler lorsqu'elle est comparée avec celle du modèle sans modules d'attention (présenté dans le chapitre précédent), indiquant ainsi que l'intégration de ces modules d'attention renforce la robustesse globale du modèle.



(a) CK+



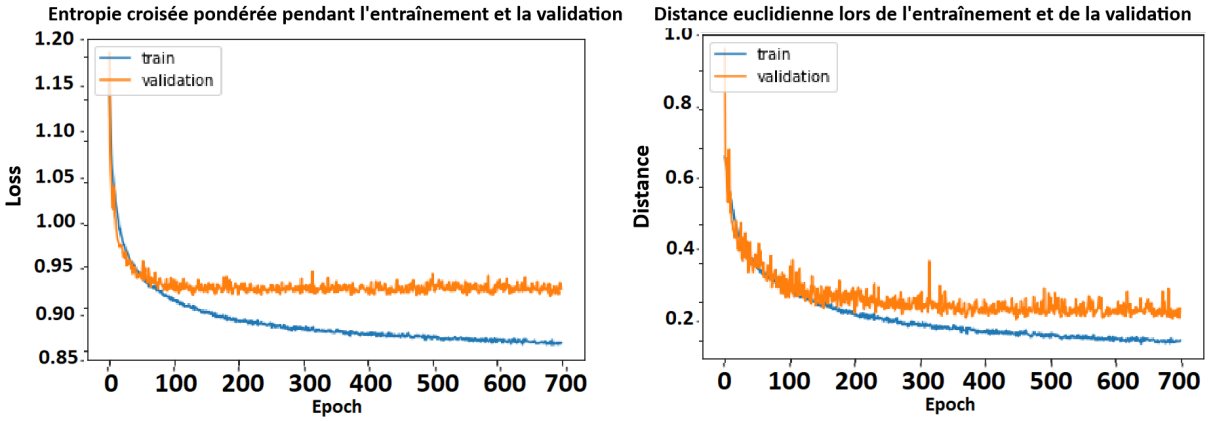
(b) BP4D



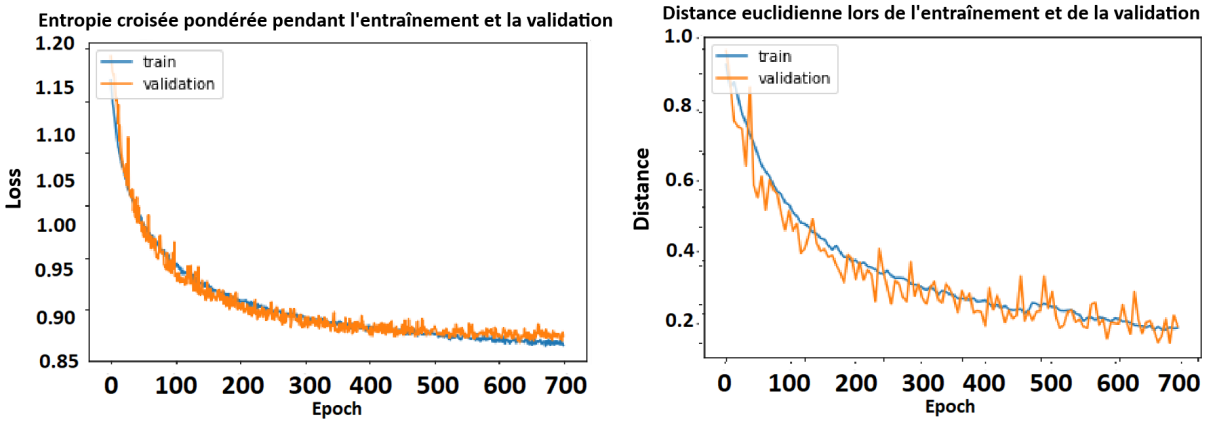
(c) DISFA

FIGURE 4.3 – Diagrammes de perte sur les jeux de données dits de laboratoire (« in-the-lab ») a) CK+, b) BP4D et c)DISFA.

Par ailleurs, pour le jeu de données FER2013+, le modèle a montré des résultats améliorés par rapport à RAFdb. Aucun décrochage de la courbe d'apprentissage n'a été



(a) RAFdb



(b) FER2013+

FIGURE 4.4 – Diagramme de perte sur les jeux de données en environnement non contrôlé (« in-the-wild ») a) RAFdb et b) FER2013+.

observé même après plusieurs époques, et les performances restent stables. Ce constat suggère que le modèle est particulièrement efficace sur ce jeu de données, validant ainsi sa capacité de se généraliser sur ce jeu de données.

4.6.3 Comparaison avec l'état de l'art

Le modèle a été comparé avec un total de 14 méthodes de l'état de l'art sur les ensembles de données BP4D, DISFA et CK+. Pour les ensembles BP4D et DISFA, les méthodes comparées sont Zhang [144], FS [145], ARL [97], STRAL [99], LGR [146], MCFE [147], IDEN [148], CWCF [149], et JAU [150]. Pour CK+, les méthodes incluent

JPML [151], Simge [81], Chen1 [101], Chen2 [101], et Elef [58].

D'un autre côté, dans le contexte des ensembles de données FER2013+ et RAFdb, une comparaison a également été réalisée avec plusieurs modèles d'état de l'art tels que LibreFace [135], SSA-ICL [5], ECAN [137], A-MobileNet [134], DNFER [136], Muhamad et al.[133], Xiaoyu et al.[138], et NCCTFER [139], ainsi que des travaux supplémentaires comme FST-MWOS [140] et Sunyoung et al. pour FER2013+.

Détection des AU sur les ensembles de données CK+, BP4D et DISFA

TABLE 4.3 – Comparaison des scores F1 entre le modèle basé sur le mécanisme d'attention proposé et les méthodes de l'état de l'art en matière de détection des AU sur l'ensemble de données CK+. Les valeurs les plus élevées sont indiquées en gras tandis que les valeurs indisponibles et/ou non-fournies sont représentées par un tiret (-).

Method /AUs	**JPML** [151]	**Simge [81]**	**Chen [101]**	**Chen [101]**	**Elef [58]**	**Notre**
AU01	0.90	0.92	0.85	0.87	0.82	**0.97**
AU02	**0.93**	0.86	0.88	0.86	0.86	**0.93**
AU04	-	0.89	0.80	0.82	0.79	**0.95**
AU05	-	-	0.74	0.74	0.73	**0.93**
AU06	0.74	0.76	0.70	0.68	0.72	**0.99**
AU07	0.66	0.81	0.61	0.55	0.57	**0.89**
AU09	-	0.87	0.89	0.89	0.87	**0.91**
AU12	0.80	0.80	0.87	0.87	0.87	**1**
AU15	-	**0.91**	-	-	0.76	0.90
AU17	0.83	-	0.86	0.84	0.86	**0.91**
AU20	-	-	-	-	0.70	**0.97**
AU23	-	-	0.45	0.32	0.67	**0.86**
AU24	-	-	0.46	-	**0.51**	-
AU25	-	-	**0.93**	0.43	0.91	-
AU26	-	-	-	0.71	0.21	**0.93**
AU27	-	-	0.89	-	**0.91**	-
AVG	0.81	0.85	0.76	0.73	0.73	**0.93**

Comparaison sur l'ensemble de données CK+ Le tableau 4.3 montre clairement que le modèle basé sur le mécanisme d'attention proposé dans ce chapitre affiche une performance supérieure à celle des modèles concurrents dans la plupart des cas. Voici quelques points saillants concernant les performances pour chaque AU et en moyenne :

— **AU01** : Notre modèle a un score F1 de 0.97, ce qui est nettement supérieur à tout autre modèle. Le plus proche concurrent (*JPML* avec un score de 0.90) est encore en dessous de notre performance.

— **AU02** : Notre modèle égale le score le plus élevé avec 0.93, montrant qu'il est à égalité avec le meilleur modèle existant.

— **AU04, AU05, et AU06** : Dans ces catégories, notre modèle domine nettement, avec des scores F1 allant jusqu'à 0.95 et 0.99.

— **AU12** : Notre modèle atteint la perfection avec un score F1 de 1.0, surpassant tous les autres modèles.

— **AU17 et AU20** : Bien que ces AU ne soient pas toujours traitées par tous les modèles de l'état de l'art, le nôtre affiche des performances très satisfaisantes avec des scores F1 de 0.91 et 0.97 respectivement.

— **Moyenne (AVG)** : Avec une moyenne de 0.93, notre modèle dépasse tous les autres modèles évalués. Le modèle le plus proche (*Simge* avec une moyenne de 0.85) est encore nettement inférieur en termes de performance globale.

En somme, notre modèle basé sur le mécanisme d'attention non seulement montre, comme nous venons de le constater, de très bonnes performances dans la détection de chaque AU individuelle mais aussi affiche une performance globale supérieure par rapport aux modèles issus de l'état de l'art.

Comparaison sur l'ensemble de données BP4D Dans cette section, nous effectuons une analyse du tableau 4.4, qui présente une comparaison des taux de score F1 entre le modèle proposé basé sur le mécanisme d'attention et diverses méthodes de l'état de l'art pour la détection des AU sur l'ensemble de données BP4D.

En examinant les taux de score F1, qui représentent un indicateur précis de la performance d'un modèle, on constate que le modèle proposé dans ce chapitre a obtenu les scores moyens les plus élevés pour les AU 12, 14 et 17. Notamment pour AU12, notre modèle

TABLE 4.4 – Comparaison des taux de score F1 entre le modèle proposé basé sur le mécanisme d'attention et les méthodes de l'état de l'art pour la détection des AU sur l'ensemble de données BP4D, catégorisées en deux groupes : modèles de grande taille ou *full-sized* dans les quatre premières colonnes et modèles légers dans les cinq dernières colonnes. Les valeurs les plus élevées sont indiquées en gras, les secondes valeurs les plus élevées sont soulignées tandis que les valeurs indisponibles et/ou non-fournies sont représentées par un tiret (-).

Method /AUs	**Zhang [144]**	**ARL [97]**	**STRAL [99]**	**JAU [150]**	**IDEN [148]**	**MCFE [147]**	**LGR [146]**	**FS [145]**	**CWCF [149]**	**Ce travail**
AU06	0.76	0.77	0.77	-	0.77	**0.79**	0.78	0.77	**0.79**	0.64
AU10	0.82	0.84	0.83	-	0.83	0.80	**0.85**	0.83	0.84	0.57
AU12	0.87	0.87	0.88	-	0.85	0.88	0.88	0.87	0.90	**0.91**
AU14	0.60	0.62	0.60	-	0.63	0.64	0.66	0.61	0.62	**0.70**
AU17	0.64	0.60	0.63	-	0.61	0.60	0.50	0.63	0.63	**0.66**
AVG	0.73	0.74	0.74	-	0.73	0.74	0.73	0.74	**0.77**	0.69
Nombre total de Param.	>138	>138	>26	>11	>6,572	>3	>4	8,19	26	**1,5**
Taille d'entrée	>256*256	200*200	200*200	256*256	128*128	170*170	176*176	256*256	224*224	**48*48**
Type de modèle	**Modèles de grande taille**						**Modèles légers**			

atteint un score de 0.91, surpassant le second modèle le plus performant avec un score de 0.90. De plus, notre modèle montre des performances satisfaisantes particulièrement dans la reconnaissance des AU14 et AU17, avec des scores de 0.70 et 0.66 respectivement, surpassant les autres modèles de l'état de l'art sur cet ensemble de données.

Il est également important de souligner que bien que le modèle proposé ait des performances légèrement inférieures pour les AU 06 et 10, avec des scores de 0.64 et 0.57 respectivement, il maintient une performance globale satisfaisante, comme indiqué par le taux de score F1 moyen (AVG), qui est de 0.69. Ce résultat est certes légèrement inférieur aux valeurs globales moyennes des travaux de l'état de l'art et montre la capacité du modèle proposé dans ce chapitre d'être compétitif ou au moins au même niveau sur cet ensemble de données.

Le tableau 4.5 montre la comparaison des taux de précision en détection des AU entre le modèle proposé dans ce chapitre et des méthodes de l'état de l'art sur l'ensemble de données BP4D.

En observant les résultats présentés dans le tableau 4.5, on note que le modèle proposé dans ce chapitre se distingue nettement avec des taux de précision élevés pour toutes les AU évaluées. De manière spécifique, les performances pour les AU 06, 10, 12, 14 et 17

sont de très bon niveau, avec des taux de précision de 89.74%, 85.17%, 87.82%, 94.36% et 98.30% respectivement. Ces taux surpassent notablement ceux des autres méthodes issues de l'état de l'art. En particulier, le taux de précision pour l'AU17 atteint un score de 98.30%. Cette performance met en évidence l'efficacité du modèle proposé dans ce chapitre.

Au-delà de la performance individuelle pour chaque AU, le taux de précision moyen (AVG) est également très révélateur de la compétence globale du modèle proposé dans ce chapitre. Le modèle proposé présente un taux de précision moyen de 91.07%, surpassant largement les scores moyens des autres méthodes comparées, dont le second meilleur score est de 76.50%.

Cette performance supérieure démontre que non seulement le modèle actuel excelle dans la reconnaissance individuelle des AU, mais il maintient également une haute performance sur l'ensemble des AU, ce qui témoigne de sa robustesse et de sa fiabilité.

TABLE 4.5 – Comparaison des taux de précision (en %) entre le modèle proposé dans ce chapitre et les méthodes de l'état de l'art en détection des AU sur l'ensemble de données BP4D, catégorisées en deux groupes : les modèles de grande taille ou *fully sized* modèles dans les quatre premières colonnes et modèles légers dans les cinq dernières colonnes. Les valeurs les plus élevées sont indiquées en gras, les secondes valeurs les plus élevées sont soulignées tandis que les valeurs indisponibles et/ou non-fournies sont représentées par un tiret (-).

Modèles /AUs	**Zhang [144]**	**ARL [97]**	**STRAL [99]**	**JAU [150]**	**IDEN [148]**	**MCFE [147]**	**LGR [146]**	**FS [145]**	**CWCF [149]**	**Modèle proposé**
AU06	-	0.79	0.78	0.79	-	-	-	-	<u>0.80</u>	**0.90**
AU10	-	0.80	0.79	0.80	-	-	-	-	<u>0.81</u>	**0.85**
AU12	-	0.85	0.86	<u>0.86</u>	-	-	-	-	0.79	**0.88**
AU14	-	0.65	0.63	0.54	-	-	-	-	<u>0.66</u>	**0.94**
AU17	-	0.74	0.74	0.43	-	-	-	-	<u>0.77</u>	**0.98**
AVG	-	<u>0.77</u>	0.76	0.68	-	-	-	-	0.76	**0.91**
Total-Params	>138	>138	>26	>11	>6,572	>3	>4	8,340	>13	**1,5**

Comparaison sur l'ensemble de données DISFA En explorant de manière critique tant les scores F1 que la précision du modèle proposé à travers les tableaux 4.6 et 4.7, une tendance distincte de performance supérieure est mise en évidence.

Dans le tableau 4.6, une variété de résultats pour les AU sont présentées. Pour les AU

telles que AU01 et AU02, notre modèle s'est avéré nettement supérieur avec des scores F1 de 0.83 et 0.87 respectivement, surpassant les autres modèles comme Zhang, ARL, et STRAL. Pour des AU plus difficiles à détecter, comme AU05, notre modèle a montré une amélioration radicale en atteignant un score F1 de 0.85, alors que la plupart des autres modèles n'ont pas fourni de résultats.

Des exceptions intéressantes sont AU20 et AU25, où notre modèle a montré une légère infériorité ou égalité avec d'autres modèles, mais même dans ces cas, les performances étaient très proches des meilleures.

TABLE 4.6 – Comparaison des taux de score F1 entre le modèle proposé basé sur le mécanisme d'attention et les méthodes de l'état de l'art pour la détection des AU sur l'ensemble de données DISFA, catégorisées en deux groupes : modèles de grande taille ou *fully sized* dans les quatre premières colonnes et modèles légers dans les cinq dernières colonnes. Les valeurs les plus élevées sont indiquées en gras, les secondes valeurs les plus élevées sont soulignées tandis que les valeurs indisponibles et/ou non-fournies sont représentées par un tiret (-).

Modèles / AUs	**Zhang**	**ARL**	**STRAL**	**IDEN**	**MCFE**	**LGR**	**FS**	**CWCF**	**Modèle proposé**
01	0.55	0.43	0.52	0.25	0.38	<u>62</u>	0.50	0.54	**0.83**
02	0.63	0.42	0.47	0.34	0.46	<u>64</u>	0.58	0.63	**0.87**
04	0.74	0.63	0.69	0.64	0.56	72	<u>0.77</u>	0.63	**0.95**
05	-	-	-	-	<u>0.17</u>	-	-	-	**0.85**
06	0.45	0.41	0.47	0.45	0.56	46	0.53	<u>0.55</u>	**0.95**
09	0.35	0.40	<u>0.56</u>	0.44	0.48	48	0.27	0.37	**0.90**
12	0.75	0.76	0.72	0.70	0.73	75	<u>0.77</u>	0.67	**0.95**
15	-	-	-	-	<u>0.30</u>	-	-	-	**0.90**
17	-	-	-	-	<u>0.45</u>	-	-	-	**0.90**
20	-	**0.95**	-	-	0.16	-	-	-	<u>0.91</u>
25	0.93	0.66	0.91	0.81	0.79	0.94	<u>0.95</u>	0.89	**0.96**
26	0.54	0.58	0.67	0.55	0.59	<u>73</u>	0.56	0.57	**0.90**
Moyenne	0.62	<u>0.74</u>	0.63	0.52	0.47	67	0.62	0.61	**0.92**
Params	>138	>138	>26	>6,572	>3	>4	8,19	26	**1,5**
Type de modèle	**Modèles de grande taille**				**Modèles légers**				

Le modèle s'est avéré être souvent à la pointe des performances, affichant des valeurs élevées pour un éventail significatif d'AU. La moyenne générale de score F1 atteint une valeur robuste de 0.92, mettant en lumière une compétence notable dans la mesure harmonieuse de la précision et du rappel à travers diverses AU.

Quant à la précision, révélée dans le tableau 4.7, les résultats sont tout aussi impressionnants. Le modèle proposé a obtenu la deuxième meilleure précision pour la majorité des AU.

La tendance est similaire pour des AU moins communnes comme AU15 et AU17, où notre modèle a montré une performance exceptionnelle malgré l'absence de données comparatives dans les autres travaux.

TABLE 4.7 – Comparaison des taux de précision entre le modèle proposé basé sur le mécanisme d'attention et les méthodes de l'état de l'art pour la détection des AU sur l'ensemble de données DISFA, catégorisées en deux groupes : modèles de grande taille ou *fully sized* dans les quatre premières colonnes et modèles légers dans les cinq dernières colonnes. Les valeurs les plus élevées sont indiquées en gras, les secondes valeurs les plus élevées sont soulignées tandis que les valeurs indisponibles et/ou non-fournies sont représentées par un tiret (-).

Modèles / AUs	**ARL**	**STRAL**	**CWCF**	**Modèle proposé**
01	0.92	<u>0.94</u>	**0.96**	0.83
02	0.86	0.89	**0.92**	<u>0.91</u>
04	**0.93**	<u>0.94</u>	0.91	0.92
05	<u>0.80</u>	**0.91**	0.80	0.85
06	<u>0.91</u>	**0.95**	0.89	0.90
09	**0.93**	0.90	<u>0.91</u>	0.91
12	0.88	**0.90**	<u>0.89</u>	0.88
15	0.82	**0.89**	0.85	<u>0.88</u>
17	0.91	**0.94**	0.92	<u>0.93</u>
20	0.91	<u>0.92</u>	**0.93**	0.91
25	0.88	<u>0.91</u>	**0.92**	0.91
26	**0.94**	**0.94**	0.84	<u>0.90</u>
Moyenne	**0.93**	**0.93**	0.91	<u>0.92</u>
Params	>138	>26	26	**1,5**
Type de modèle	**Modèles de grande taille**		**Modèles légers**	

Simultanément, la précision du modèle démontre une robustesse considérable dans la reconnaissance des expressions faciales sur le jeu de données. En effet, la précision moyenne de notre modèle, également positionnée à 0.92, ne s'épanouit pas seulement en tant que la plus performante ou la deuxième plus performante pour un vaste ensemble d'AU, mais également elle égale voire surpasse les meilleures performances des travaux existants dans l'état de l'art.

Détection des émotions sur les ensembles de données FER2013+, RAFdb et CK+.

Pour détecter les différents types d'émotions, le modèle proposé dans ce chapitre, étant essentiellement basé sur la détection des AU, utilise l'encodage spécifique et défini par Ekman (voir le tableau 3.1). Autrement dit, le modèle proposé utilise la combinaison des AU détectées pour discerner les expressions faciales et retrouver l'émotion qui en résulte. Par exemple, si le modèle proposé pour une image d'entrée indique la combinaison composée de AU04, AU05, AU07 et AU23, c'est l'émotion "Colère" qui est spécifiquement codée par cette combinaison et qui sera décodée (et identifiée) selon le tableau de Ekman. De la même manière, nous avons encodé d'autres ensembles d'AU pour représenter les autres type d'émotions comme "En colère" , "Dégout", "Peur", "Joie", "Tristesse" et "Surprise".

TABLE 4.8 – Comparaison en termes de précision entre le modèle proposé dans ce chapitre et les méthodes de l'état de l'art pour la reconnaissance des expressions faciales sur la base de données FER2013+. Les valeurs les plus élevées sont indiquées en gras.

Modèles	**Précision (en %)**	**Params (M)**
Modèle proposé (avec attention)	**92.15**	**1.5**
FST-MWOS [140]	90.41	-
DNFER [136]	89.32	-
LAN [141]l.	88.45	-
NCCTFER [139]	88.21	-
A-MobileNet [134]	88.11	3.4
Modèle proposé (sans attention)	81.40	**1.5**

Analyse des résultats sur la FER2013+ L'inspection minutieuse du tableau 4.8 permet de dégager plusieurs observations importantes sur les performances des différentes méthodes de reconnaissance des expressions faciales utilisées sur la base de données FER2013+. Le modèle désigné par "Modèle proposé (avec attention)" se démarque avec une précision de 92.15%, se positionnant ainsi avec ces performances au sommet des modèles proposés dans la littérature. Ces très bons résultats sont d'autant plus à souligner en

prenant en compte le fait que le modèle parvient à les atteindre avec un nombre de paramètres limité par rapport aux autres modèles issus de la littérature, en ne comptabilisant que 1.5 millions de paramètres.

Par ailleurs, le modèle "FST-MWOS" est le modèle issu de la littérature s'approchant le plus aux performances obtenues par notre modèle basé sur les mécanismes d'attention, affichant une précision de 90.41%, bien que sa complexité en termes de paramètres ne soit pas spécifiée. Les autres méthodes, telles que "DNFER" et "NCCTFER", offrent également des performances s'approchant des 90%. Il est également à noter que pour plusieurs modèles, malgré leurs précisions relativement élevées, leur nombre de paramètres n'est pas fourni, empêchant ainsi une comparaison complète avec le modèle proposé dans ce chapitre. De plus, on peut également constater que l'intégration des mécanismes d'attention dans le modèle proposé dans le chapitre précédent a permis d'améliorer considérablement ses performances et de le rendre compétitif avec les approches de l'état de l'art (une amélioration de 81.4% à 92.15%).

TABLE 4.9 – Comparaison en termes de précision entre le modèle proposé dans ce chapitre et les méthodes de l'état de l'art pour la reconnaissance des expressions faciales sur la base de données RAF-db. Les valeurs les plus élevées sont indiquées en gras.

Modèles	**Précision (en %)**	**Params (M)**
Modèle proposé (avec attention)	**94.87**	**1.5**
DNFER [136]	90.41	-
ECAN [137]	89.77	-
SSA-ICL [5]	89.44	11
NCCTFER [139]	87.97	-
Xiaoyu et al. [138]	87.58	-
Muhamad et al. [133]	84.91	2
A-MobileNet [134]	84.49	3.4
Modèle proposé (sans attention)	83.00	**1.5**
LibreFace [135]	82.79	43

Analyse des résultats sur la RAF-db L'examen du tableau 4.9 suggère une nette supériorité du modèle intitulé "Modèle proposé (avec attention)", qui affiche une précision de 94.87% sur la base de données RAF-db, surpassant ainsi tous les modèles concurrents

listés. Comme évoqué précédemment, cette performance est obtenue avec un nombre de paramètres relativement restreint (1.5 million), suggérant une efficacité optimale en termes de ressources. À titre comparatif, d'autres modèles tels que "DNFER", offrent également une précision élevée (90.41%), bien que la complexité de ce dernier ne soit pas spécifiée. Par ailleurs, une variation notable dans la quantité de paramètres entre les différents modèles, allant de 1.5 à 43 millions de paramètres, témoigne de la diversité des architectures employées dans ce domaine de recherche. De plus, on peut également constater que l'intégration des mécanismes d'attention a également permis d'améliorer considérablement les performances du modèle initial (présenté dans le chapitre précédent) sur ce jeu de données en passant de 83% à 94.87% et le rendant plus que compétitif par rapport à l'état de l'art.

Analyse des sorties de Grad-CAM

Pour garantir une explicabilité claire des résultats de notre modèle, nous avons appliqué l'outil Grad-CAM sur sa sortie permettant ainsi d'obtenir des éléments d'analyse supplémentaires pour mieux comprendre le fonctionnement du mécanisme d'attention du modèle dans la détection des AU.

En premier lieu, nous avons présenté uniquement les images de l'ensemble de données CK+ au modèle entraîné de manière séquentielle sur la totalité des ensembles de données mentionnés précédemment. Les résultats de ce premier cas de figure sont illustrés en figure 4.5, où l'on remarque la première rangée correspondant aux images d'entrée de l'ensemble CK+ et les résultats de visualisation dans les rangées successives correspondant au modèle entrainé sur les ensembles de CK+, BP4D, DISFA, RAFdb et Fer2013+ respectivement. Puis, dans un second temps, nous avons présenté au modèle entraîné de la même manière que dans le premier cas de figures, uniquement les images de l'ensemble de données utilisé pour l'entraînement. Autrement dit, pour le modèle entrainé sur l'ensemble CK+, uniquement les images de cet ensemble ont été présentées au modèle ; pour le modèle entrainé sur DISFA, uniquement les images de cet ensemble lui ont été présentées, etc. Les résultats

obtenus pour ce second cas de figures sont présentés en figure 4.6, où on peut remarquer la présence d'une rangée de figures propres à chaque ensemble de données et une rangée supplémentaire affichant les résultats de visualisation avec la méthode Grad-CAM.

Dans les deux cas de figures, on peut constater une amélioration significative au niveau des résultats de visualisation, et ceci notamment en termes des régions désignées comme prépondérantes dans la décision finale du modèle. Le mécanisme d'attention a apporté des améliorations, non seulement au niveau de la précision de détection des AU et de leur identification, mais également par rapport à la détermination plus précise des régions focalisées pour chaque AU.

Les cartes de chaleur produites (et superposées sur les images originales) dévoilent les secteurs précis des images qui contribuent de manière significative à la détection des différentes AU, témoignant de manière visuelle de l'amélioration sensible du modèle dans la reconnaissance des AU. Il est également clair que la méthode Grad-CAM favorise une meilleure compréhension des zones d'intérêt prioritaires et ciblées par le modèle proposé dans ce chapitre lors de la prédiction des AU, réaffirmant ainsi l'efficacité et la pertinence des mécanismes d'attention rajoutés.

4.6.4 Interprétation des résultats

Dans ce chapitre, nous avons présenté les résultats obtenus après l'intégration des mécanismes d'attention dans le modèle présenté initialement dans le chapitre précédent. En effet, cette intégration s'est révélée productive du point de vue des performances obtenues sur les différents ensembles de données.

Sur l'ensemble de données CK+, nous avons enregistré un score F1 moyen de 93%. Il est également intéressant de noter que notre modèle se soit distingué comme le meilleur dans la détection de presque toutes les AU, à l'exception de l'AU15.

Pour l'ensemble de données BP4D, le score F1 moyen est de 69%. Bien que ce ne soit pas le score le plus élevé, notre modèle a surclassé les modèles de l'état de l'art dans la

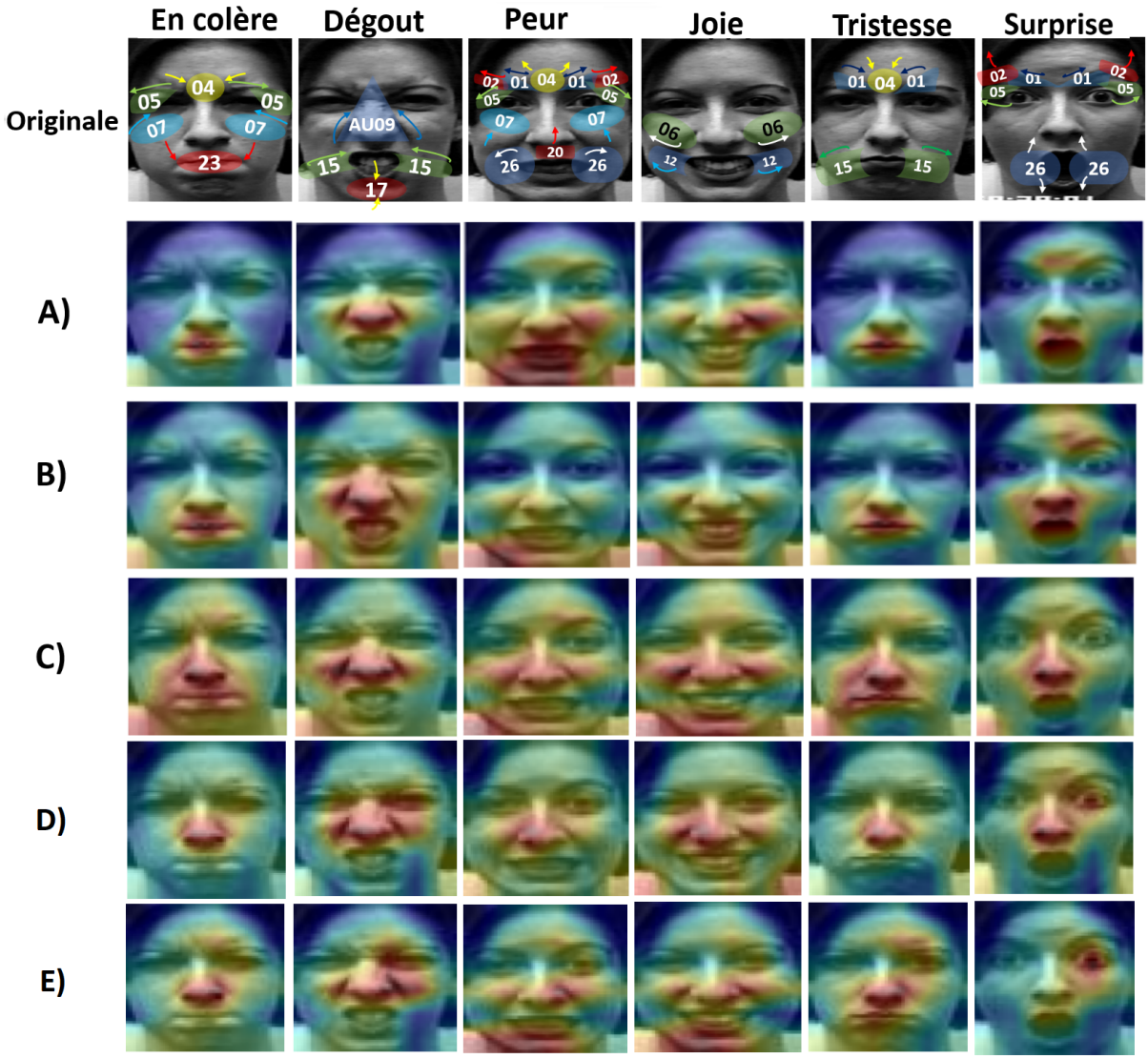


FIGURE 4.5 – Sortie du modèle sur des images sélectionnées de l'ensemble de données CK+ : les rangées A) à E) sont les cartes de chaleur des AU correspondantes générées par la méthode Grad-CAM pour le modèle entraîné sur A) CK+, B) BP4D, C) DISFA, D)RAFdb et E)Fer2013+ .

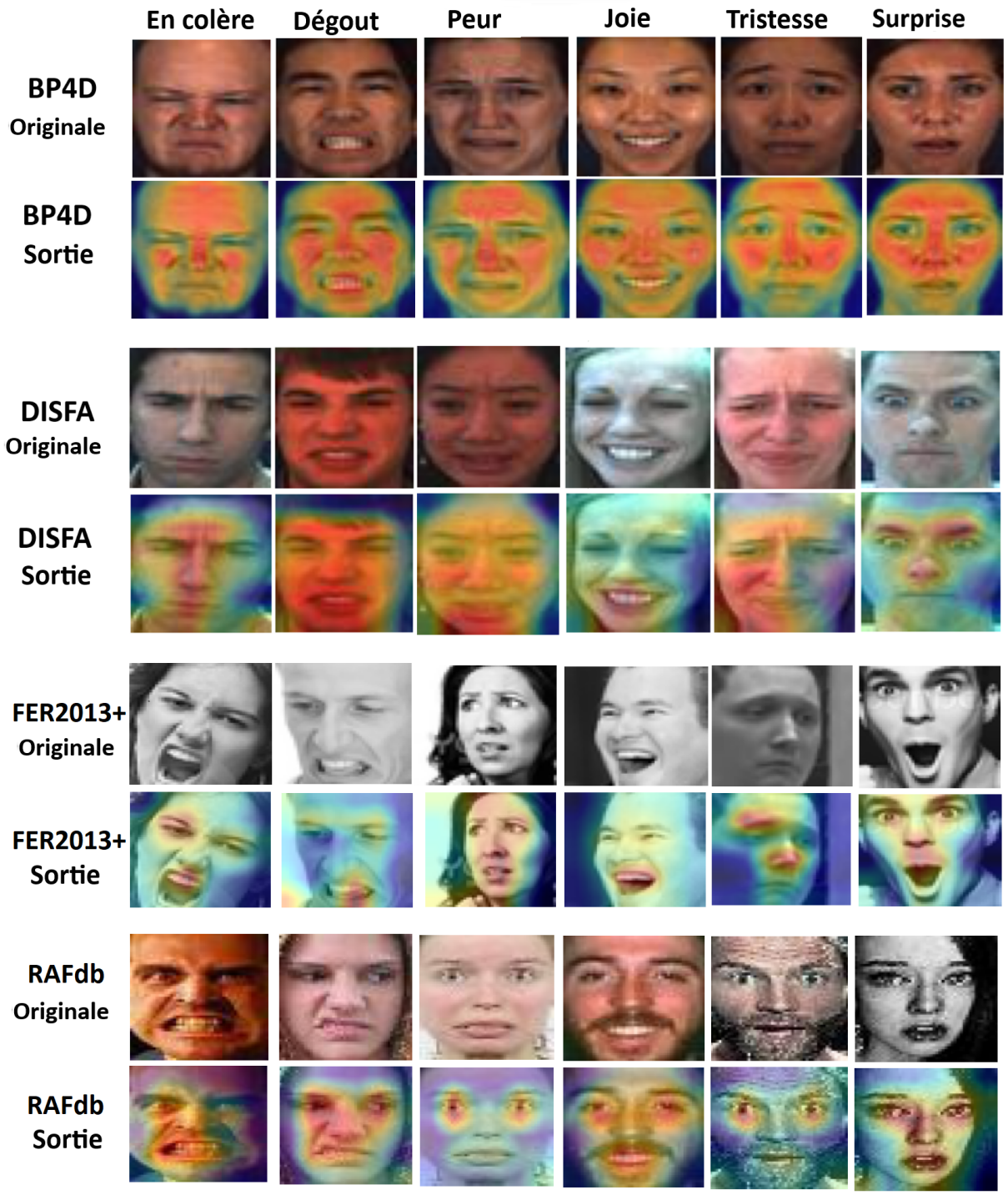


FIGURE 4.6 – Sortie du modèle sur des images sélectionnées des ensembles de données utilisés pour d'entraînement (les rangées impaires) : les rangées paires sont les cartes de chaleur des AU correspondantes générées par la méthode Grad-CAM pour le modèle entraîné sur BP4D, DISFA, Fer2013+ et RAFdb.

détection des AU 12, 14 et 17. Par ailleurs, il a enregistré une précision d'un très bon niveau de 91.7%, le score le plus élevé dans cette catégorie, y compris dans la précision de la détection individuelle des AU.

Quant à l'ensemble de données DISFA, notre modèle a montré une bonne performance avec un score F1 moyen de 92%, le score le plus élevé reporté jusqu'à présent dans la littérature. À l'exception de l'AU20, notre modèle a dominé dans la détection de chaque AU, présentant également une précision moyenne de 92%, le deuxième score le plus élevé reporté dans littérature, ainsi que la précision la plus élevée ou au même niveau des modèles concurrents dans la plupart des AU individuelles.

Dans la reconnaissance des expressions faciales où le décodage des AU selon la méthode de Ekman permet de retrouver l'émotion à l'origine des AU détectées, les ensembles de données RAFdb et FER2013+ témoignent également de l'efficacité de notre modèle. Sur l'ensemble de données RAFdb, le modèle proposé atteint une précision de 94.87%, tandis que sur l'ensemble de données FER2013+, il établit un nouveau record avec une précision de 92.15%, surpassant tous les autres modèles concurrents de l'état de l'art.

Les progrès réalisés dans ce chapitre ne se limitent pas seulement aux scores et à la précision. L'explicabilité du modèle s'est grandement améliorée, permettant une détection très claire des AU tant en termes d'intensité que de positionnement.

De plus, en comparant avec les résultats des modèles présentés dans les chapitres précédents, il est maintenant évident que notre modèle intégrant les mécanismes d'attention peut se placer en position de candidat sérieux pour une implantation embarquée, ouvrant ainsi la voie à des applications pratiques dans des systèmes à ressources limitées.

4.7 Conclusion

Dans ce chapitre, nous avons mis en évidence les performances de très bon niveau du modèle proposé pour la détection des AU et la FER. Grâce à l'intégration d'un mécanisme

d'attention efficace, notre travail a démontré une compétence notable dans la détection individuelle et globale des AU à travers divers ensembles de données, y compris CK+ et BP4D, en obtenant des scores F1 et des taux de précision élevés.

Le mécanisme d'attention a non seulement amélioré la précision de la détection des AU, mais a également contribué à une meilleure focalisation sur les régions pertinentes des images, comme l'indiquent les analyses Grad-CAM. Les cartes de chaleur obtenues illustrent clairement les zones qui ont été déterminantes dans la reconnaissance des AU, fournissant ainsi une validation visuelle de l'efficacité du modèle.

Notre modèle a également montré son efficacité dans la reconnaissance des expressions faciales sur les ensembles de données RAFdb et FER2013+, établissant de nouveaux standards de précision dans ce domaine.

Cependant, malgré ces succès, il est impératif de continuer à travailler sur l'optimisation du modèle pour garantir son intégration efficiente dans des systèmes embarqués. Ceci est d'autant plus crucial que l'un des objectifs de cette thèse est de développer un modèle non seulement performant, mais également économe en termes de ressources, permettant ainsi une utilisation temps réel dans des systèmes aux ressources limitées.

Par conséquent, la prochaine étape qui sera présentée dans le chapitre suivant consistera à approfondir les méthodes et techniques permettant d'affiner encore le modèle proposé, pour parvenir à une solution qui maintient un équilibre optimal entre précision et efficacité opérationnelle en termes de ressources utilisées, prête à être déployée dans des systèmes embarqués.

Chapitre 5

Vers une architecture FER pour les systèmes embarqués

Sommaire

5.1 Introduction

Ce chapitre se focalise sur la mise en œuvre d'une version optimisée du modèle introduit dans le chapitre précédent, adaptée à un fonctionnement sur des dispositifs avec des contraintes de ressources. Pour concilier la taille réduite du modèle et le maintien de ses performances, nous explorerons une série de techniques d'optimisation. L'approche adoptée dans ce chapitre cherche à évaluer l'impact de différentes méthodes, comme l'élagage et la quantification, non seulement sur les métriques de classification évoquées antérieurement mais aussi sur les ressources requises pour l'exécution du modèle optimisé.

5.2 Optimisation de modèles d'apprentissage profond

L'apprentissage profond a connu un succès retentissant, en grande partie grâce à la mise en œuvre de réseaux neuronaux toujours plus vastes [157, 158]. Si ces modèles de grande dimension peuvent offrir d'excellentes performances sur un ensemble de tâches, leur utilisation et leur déploiement présentent des défis. En effet, leur taille conséquente exige davantage d'espace de stockage et peut augmenter le temps d'exécution, ce qui peut être critique dans des applications nécessitant des réponses en temps réel [159, 160].

Dans ce contexte, ce chapitre se concentre sur diverses techniques visant à optimiser et réduire la taille des modèles d'apprentissage profond, ainsi que leur nombre de paramètres, pour faciliter leur intégration dans des systèmes embarqués. Parmi les stratégies examinées, on compte l'élagage, la quantification, la distillation des connaissances, et la binarisation. Chaque technique, avec ses avantages et inconvénients propres, peut être employée indépendamment ou en synergie pour atteindre des niveaux d'optimisation spécifiques. Nous détaillerons ces méthodes, en commençant par une présentation générale avant d'examiner comment elles peuvent s'appliquer aux modèles FER et AU introduits précédemment, et en terminant par une discussion sur leur efficacité à travers les résultats obtenus.

La figure 5.1 illustre les principales étapes d'optimisation appliquées au modèle comportant deux blocs d'attention et présenté dans le chapitre précédent. Le modèle initial passe tout d'abord par l'étape de restructuration où la taille des filtres utilisés dans toutes les couches de convolution est réduite. Ce modèle restructuré doit être entraîné avant de passer à l'étape suivante. Dans l'étape suivante, les techniques d'optimisation telles que l'élagage et la quantification sont appliquées pour alléger encore plus le modèle restructuré. Le modèle ainsi optimisé doit également passer par l'étape d'entraînement avant d'être finalement utilisé dans une plateforme embarquée. Bien qu'une légère dégradation des performances puisse survenir après cette optimisation, des entraînements supplémentaires peuvent être envisagés pour récupérer, voire améliorer, les performances initiales.

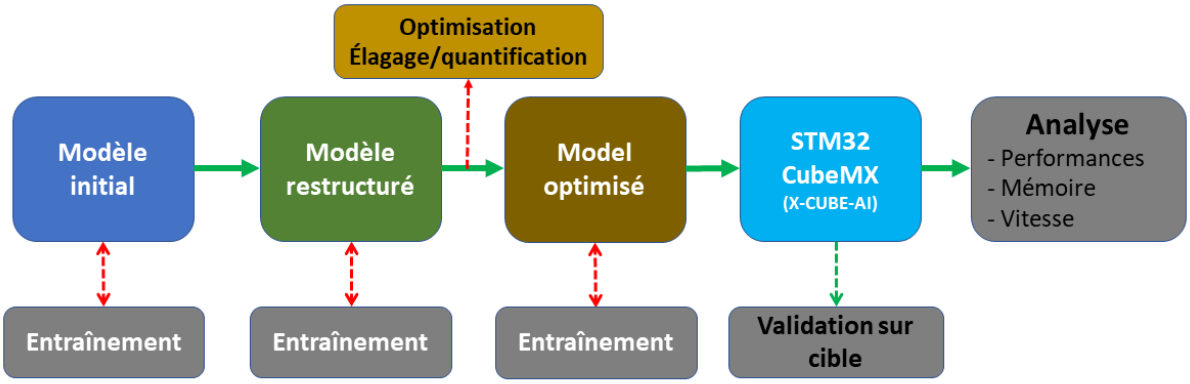


FIGURE 5.1 – Étapes d'optimisation appliquées sur le modèle initial présenté dans le chapitre précédent.

5.2.1 Motivations

Bien que les modèles CNN aient prouvé leur efficacité dans la reconnaissance des expressions faciales, comme démontré dans nos précédentes études, leur grande taille les rend souvent inadaptés pour un déploiement direct sur des dispositifs avec des contraintes de ressources, tels que les appareils mobiles ou IoT [161].

Face à des exigences telles que la limitation de puissance de calcul, la gestion de la consommation énergétique, l'espace mémoire restreint et le besoin de réponses en temps

réel (comme c'est le cas pour la FER), il devient impératif d'adapter et d'optimiser ces modèles pour répondre à ces contraintes [162].

Diverses techniques d'optimisation, comme la quantification des poids, l'élagage ou la refonte de l'architecture du modèle, peuvent être appliquées pour atteindre ces objectifs [163]. La quantification, notamment, peut significativement réduire la mémoire nécessaire tout en augmentant la vitesse de calcul, bien que cela puisse influer sur la précision du modèle [164].

Dans cette perspective, l'accent de ce chapitre sera mis sur l'adaptation et l'optimisation de notre modèle CNN, conçu avec des mécanismes d'attention pour la détection des unités d'action, afin d'obtenir un modèle optimisé, performant et adapté aux systèmes embarqués, sans compromettre sa précision.

5.2.2 Optimisation de modèles FER dans la littérature

Dans le secteur de la reconnaissance des expressions faciales (FER), de nombreuses recherches se sont focalisées sur l'adaptation des systèmes pour les plateformes embarquées, compte tenu de leurs contraintes spécifiques.

Les travaux présentés dans [165] introduisent *EmotionNet Nano*, un réseau neuronal convolutif efficace conçu pour la reconnaissance en temps réel des expressions faciales. Distinctive par sa simplicité architecturale, cette approche a été façonnée par une combinaison d'expertise humaine et algorithmique. Sur le jeu de données CK+, ce réseau affiche une précision de 97.6% tout en étant nettement moins gourmand en paramètres. Sa performance est manifeste, assurant une inférence en temps réel avec une haute efficacité énergétique sur une plateforme ARM.

La recherche menée dans [109] propose une méthodologie hybride pour FER combinant les capacités des CNN et des *k-Nearest Neighbour (KNN)*. Implanté sur un Raspberry Pi 4, leur modèle, basé sur *EfficientNet-Lite*, atteint une précision de 75.26% sur le jeu de données FER-2013, démontrant la possibilité d'implantation de tel modèle sur des

dispositifs embarqués.

Dans une perspective similaire, [166] détaille un modèle pour la détection des micro-expressions, opérant sur Raspberry Pi. Avec une sensibilité de 75.26% et une spécificité de 93.93%, ce modèle, testé sur FER2013, atteint une précision globale de 65.97

Finalement, des approches basées sur les FPGA (*Field Programmable Gate Arrays*) sont explorées dans [167] et [168]. La première, opérant sur un SoC FPGA, assure la reconnaissance de sept émotions fondamentales avec une précision de 66% sur FER2013, tout en traitant jusqu'à 15 images par seconde. La seconde, exécutée sur une carte FPGA AtlysTM Spartan-6, affiche une précision de 47.44%, avec une capacité de traitement en temps réel de 30 images par seconde.

Ces travaux mettent en évidence les efforts constants pour optimiser et adapter les modèles FER à des environnements à ressources limitées, affirmant l'importance de cette problématique dans le domaine.

5.3 Optimisation du modèle proposé

L'objectif principal de ce travail est d'optimiser le modèle de FER et de détection des AU, présenté dans les chapitres précédents. Bien que le modèle initial soit léger et performant, il y a encore une marge d'optimisation pour le rendre encore plus adapté aux systèmes embarqués possédant des ressources limitées.

Pour réaliser cette optimisation, nous avons suivi deux étapes principales :

1. **Restructuration de l'architecture CNN :** Nous avons modifié les tailles de filtre des couches CNN. Cette modification a permis une réduction significative de la taille du modèle. Les détails de cette réduction sont illustrés dans l'algorithme 1.

2. **Optimisation avec TensorFlow lite :** Nous avons ensuite fait appel à TensorFlow Lite. Cet outil d'optimisation, riche en fonctionnalités, englobe des techniques

comme l'élagage et la quantification, facilitant ainsi une réduction substantielle de la taille du modèle tout en maintenant ou améliorant ses performances.

Les sections suivantes décrivent ces techniques d'optimisation en détail, expliquent comment elles ont été appliquées à notre modèle et évaluent l'impact de ces modifications sur les performances du modèle.

5.3.1 Restructuration de l'architecture CNN

La première étape pour optimiser notre modèle a été de modifier son architecture. La nouvelle architecture optimisée est détaillée dans l'Algorithme 1. Bien qu'elle présente certaines similitudes avec celle que nous avons décrite précédemment, il y a des différences notables. Il est important de souligner que ni l'architecture globale ni les paramètres d'entraînement du modèle n'ont été modifiés par rapport à ce qui a été présenté dans les chapitres précédents (voir la figure 4.1).

Pour rappel, cette architecture est composée de deux blocs de convolution, deux blocs d'attention et un bloc de classification. Chaque bloc de convolution contient cinq couches utilisant des filtres de taille 3×3. Les couches comportent respectivement 16, 16, 32, 32 et 32 filtres, enrichies par des opérations telles que le max-pooling, le dropout, et la normalisation par lots pour garantir une performance optimale. Ces configurations diffèrent de celles abordées dans les chapitres précédents, comme le montre la comparaison détaillée dans le Tableau 5.3.1.

TABLE 5.1 – Comparaison des tailles de filtres entre le modèle initial et le modèle restructuré.

Couche	**Modèle initial**	**Modèle restructuré**
Convolution 1	32	16
Convolution 2	64	16
Convolution 3	128	32
Convolution 4	128	32
Convolution 5	64	32

Suite aux blocs de convolution, des blocs d'attention composés de couches d'attention

spatiale et de canal sont introduits. Ces couches sont conçues pour mettre l'accent de manière sélective sur les caractéristiques les plus informatives, améliorant ainsi la précision de la reconnaissance des émotions.

Le bloc de classification, qui joue un rôle essentiel dans l'inférence des émotions, contient trois couches avec respectivement 32, 32 et 13 neurones. La dernière couche applique une fonction sigmoïde pour calculer la probabilité d'activation pour chaque AU. Ces AU sont ensuite associées aux émotions correspondantes à l'aide d'une fonction dédiée. Les performances du modèle sont évaluées sur la base de la précision de reconnaissance des émotions et des scores F1 en suivant le cadre de la méthode FACS, comme détaillé dans le tableau 3.1.

entrée: 48×48 image d'entrée $\boldsymbol{i}$ à analyser
sortie : Vecteur de sortie de probabilité de classe $\boldsymbol{o}$
$\boldsymbol{x} \leftarrow \boldsymbol{i}$
$f \leftarrow [16, 16, 32, 32, 32]$ *// nombre de filtres par couche*
$n \leftarrow 13$ *// nombre de neurones dans la couche de sortie*
for $i \in \{0, 1\}$ **do**
  if *i=0* **then**
    for $j \in \{0, 1, 2\}$ **do**
      $\boldsymbol{x} \leftarrow$ `Conv2D` $(f[j])(\boldsymbol{x})$
      if *j=1* **then**
        $\boldsymbol{x} \leftarrow$ `MaxPool` $(\boldsymbol{x})$
        $\boldsymbol{x} \leftarrow$ `Dropout` $(\boldsymbol{x})$
      if *j=2* **then**
        $\boldsymbol{x} \leftarrow$ `BatchNorm` $(\boldsymbol{x})$
  else
    for $j \in \{3, 4\}$ **do**
      $\boldsymbol{x} \leftarrow$ `MaxPool` $(\boldsymbol{x})$
      $\boldsymbol{x} \leftarrow$ `Conv2D` $(f[j])(\boldsymbol{x})$
  $\boldsymbol{x} \leftarrow$ `ChannelAttention` $(\boldsymbol{x})$
  $\boldsymbol{x} \leftarrow$ `SpatialAttention` $(\boldsymbol{x})$
$\boldsymbol{x} \leftarrow$ `FC` $(\boldsymbol{x}$,n$)$
$\boldsymbol{o} \leftarrow$ `Sigmoid` $(\boldsymbol{x})$
return $\boldsymbol{o}$

Algorithm 1: *Architecture du modèle restructuré pour la détection des AU.*

Réglage des hyperparamètres

Les hyperparamètres optimaux sélectionnés pour l'entrainement du modèle sont identiques à ceux utilisés dans le chapitre précédent et sont rappelés dans le tableau 5.2. Ils font usage de l'algorithme d'optimisation Adam, ainsi que de la fonction de perte à entropie croisée pondérée pour pallier le déséquilibre des classes dans les données. Les performances du modèle ont été évaluées en utilisant la même métrique que précédemment, la distance euclidienne.

TABLE 5.2 – Hyperparamètres optimaux utilisés pour l'entraînement du modèle optimisé

Paramètre	**Valeur**
Algorithme d'optimisation	Adam
Taux d'apprentissage	0.0001
Taux de décroissance	10^{-6}
Nombre d'époques	700
Fonction de perte	Entropie croisée pondérée
Métrique de performance	Distance euclidienne

Procédure d'entraînement

Dans les chapitres précédents, nous avons entraîné notre modèle sur plusieurs jeux de données. Cependant, dans ce chapitre, nous avons décidé de privilégier un jeu de données *in-the-lab*, à savoir CK+, et un jeu de données *in-the-wild*, soit FER2013+. Bien que nous ayons précédemment entraîné notre modèle avec tous ces ensembles de données, nous avons choisi de concentrer nos efforts sur CK+ et FER2013+ car ils sont respectivement représentatifs des jeux de données acquis dans des conditions contrôlées en laboratoire et dans des conditions non-contrôlées, typiques des scénarios réels.

Pour entraîner et évaluer le modèle optimisé, nous avons suivi une méthodologie similaire à celle décrite dans les chapitres précédents. Dans chaque jeu de données, chaque image était présentée au modèle optimisé pour obtenir les prédictions des AU correspondant à l'image d'entrée. Ces prédictions étaient ensuite converties en émotions grâce à une fonction spécifique, nommée *AU-to-emotion*. Cette fonction se base sur les combinaisons

d'AU activées spécifiques à chaque émotion, conformément au système FACS d'Ekman présenté plus tôt [12].

Ces étapes ont été cruciales pour construire les étiquettes d'entraînement utilisées tant pour la phase d'entraînement du modèle optimisé que pour évaluer ses performances globales.

5.3.2 Optimisation avec TensorFlow Lite

Les techniques d'optimisation à appliquer sur le modèle détaillé précédemment dépendent également de la plateforme embarquée cible choisie. Dans ce travail, et de manière générale pour une preuve de concept permettant de démontrer la capacité des modèles proposés dans cette thèse d'aller vers une implantation embarquée, nous nous sommes focalisés uniquement sur une implantation microcontroleur en ciblant en particulier les circuits de la société STMicroelectronics adaptés pour les applications embarquant des modèles d'apprentissage profond de faible complexité. C'est notamment les circuits de la famille STM32 (en particulier le microcontroleur STM32L476RGT) qui ont retenu notre attention. Pour ce faire, l'utilisation des outils de développement spécifiques et associés à ces circuits sont indispensables, notamment de l'environnement STM32Cube.AI et MX.

Le modèle d'apprentissage profond que nous souhaitions porté sur le circuit microprogrammé mentionné dans la section précédente, bien qu'il dispose d'un nombre de paramètres relativement faible par rapport à d'autres modèles de la littérature, ne peut pas être utilisé dans son état initial. Plusieurs étapes d'adaptation et d'optimisation sont nécessaires. Pour un équilibre entre la taille finale du modèle qui sera implanté et ses performances, diverses techniques d'optimisation sont nécessaires. Dans ce travail, l'exploration spécifique des effets de l'élagage du modèle et de la quantification sur les performances est effectuée dans un pemier temps. Ces techniques sont appliquées au modèle initial à l'aide notamment de la boîte à outils d'optimisation de modèle de la library *TensorFlow*.

Élagage

Afin de minimiser la taille et d'améliorer les performances du modèle, une procédure d'élagage est appliquée. La technique utilisée pour l'élagage est celle de l'élagage pondéral basé sur l'amplitude, dans lequel les poids les plus petits du modèle sont progressivement mis à zéro. Le processus d'élagage commence avec un niveau de parcimonie initial de 50% et augmente à 80% à la fin du processus d'élagage, qui doit coïncider avec la fin de l'entraînement.

L'élagage permet de réduire la taille du modèle initial par un facteur dépendant du niveau de parcimonie utilisé (par exemple par 3 pour un niveau de parcimonie de 80%). Malgré cette réduction significative de la taille du modèle, l'élagage n'affecte pas significativement l'exactitude du modèle, ce qui témoigne de l'efficacité du processus d'élagage (voir la section 5.4).

Quantification

La quantification est une technique qui vise à réduire la précision numérique des poids du modèle. Cela conduit non seulement à une taille de modèle considérablement réduite mais aussi à un temps de calcul plus court. Après avoir procédé à l'étape d'élagage, le modèle est converti en un format TensorFlow Lite, sur lequel différents types de processus de quantification sont appliqués.

Le premier type d'optimisation est la quantification de la plage dynamique. Elle convertit la précision des poids du modèle dans la plage allant de 32 bits à 8 bits. En termes simples, cela signifie qu'au lieu d'utiliser des formats de données de 32 bits pour représenter les valeurs dans le modèle, des formats allant jusqu'à 8 bits sont employés. Cela réduit grandement la taille du modèle et les besoins en calcul.

Le deuxième niveau de quantification est la quantification `float16`. Elle diminue la précision de tous les poids à 16 bits, offrant une réduction supplémentaire de la taille du modèle. Cette forme de quantification est particulièrement adaptée pour le déploiement

sur des processeurs graphiques (GPU) et d'autres accélérateurs matériels conçus pour bénéficier de l'arithmétique de précision réduite.

La dernière étape est la quantification `float32`. Bien qu'elle ne réduise pas davantage la taille du modèle, elle l'optimise pour les dispositifs et plates-formes uniquement compatibles avec les calculs en `float32`.

Il est important de noter que, bien que les modèles quantifiés soient nettement plus compacts que leurs versions originales, ils parviennent à conserver d'excellentes performances. Cette préservation des performances, malgré une taille nettement réduite, s'explique en grande partie par la nature redondante des réseaux neuronaux. Nombre de leurs paramètres ont peu ou pas d'impact sur les sorties, les rendant superflus [169]. En combinant élagage et quantification, il est possible d'exploiter cette redondance, permettant de réduire considérablement la taille du modèle tout en conservant son efficacité [170].

5.4 Résultats d'optimisation

Le processus d'optimisation a été effectué en plusieurs étapes pour réduire la taille du modèle initial tout en préservant sa performance. Chaque étape de l'optimisation a abouti à une réduction significative de la taille du modèle, le rendant ainsi plus adapté pour le déploiement sur des plateformes à ressources limitées.

Modèle original vs modèle restructuré : Le modèle original, référencé dans [171], était constitué de 1.5 millions de paramètres et avait une taille de 5 407 074 octets (détails dans le tableau 5.3). Le modèle après restructuration, malgré la réduction à 57 001 paramètres, a vu sa taille diminuée à 216 344 octets.

Élagage : La démarche d'élagage, appliquée au modèle restructuré (comme détaillé dans la sous-section 5.3.1), a permis de conserver les 57 001 paramètres mais avec une taille réduite à 74 854 octets une fois converti au format TensorFlow Lite (TFLite).

Quantification : Le processus de quantification sur le modèle TFLite élagué a offert

des réductions de taille encore plus importantes. La quantification de la plage dynamique a réduit la taille du modèle à 30 874 octets. En utilisant la quantification `float16`, la taille est passée à 47 728 octets. Notamment, la quantification `float32` n'a pas changé la taille, la laissant à 74 854 octets, identique à celle du modèle élagué.

Toutes ces améliorations, illustrées dans le tableau 5.3, mettent en évidence l'efficacité des techniques d'optimisation utilisées pour préparer le modèle pour une implantation embarquée où les ressources disponibles sont limitées.

TABLE 5.3 – Comparaison de divers modèles sur le jeu de données FER2013+. Le tableau comprend les versions initiales et optimisées des modèles, mettant en évidence l'impact des techniques d'élagage et de quantification sur la performance et la taille du modèle.

Version du modèle	**Paramètres**	**Précision [%]**	**Score F1 [%]**	**Taille [Ko]**
Modèle initial (HEF [171])	1.5M	92.12	74.12	5,688
Modèle restructuré	57K	92.11	74	280
Modèle optimisé (élagué / quantifié)	57K	74.0	73.6	235

5.4.1 Évaluation des modèles optimisés sur FER2013+

Le tableau 5.3 fournit une vue comparative des différentes versions des modèles sur le jeu de données FER2013+. Cette section détaille les améliorations significatives obtenues par le modèle optimisé par rapport au modèle initial de [171], particulièrement en ce qui concerne la taille du modèle.

Modèle initial : Le modèle initial, constitué de 1.5 millions de paramètres, affiche une précision de 92.12% et un score F1 de 74.12%. Sa taille est relativement conséquente, s'élevant à 5 688 KB.

Modèle restructuré : Par opposition, le modèle restructuré, malgré une nette réduction à 57 001 paramètres, présente une précision presque identique au modèle initial de 92.11% et un score F1 légèrement diminué à 74%. La taille de ce modèle a été réduite de manière drastique à 280 Ko, illustrant l'efficacité des techniques d'optimisation tout en conservant une performance proche du modèle d'origine.

Modèle optimisé (élagué et quantifié) : La dernière version du modèle, élaguée

et quantifiée, bien que gardant le même nombre de paramètres que le modèle optimisé, présente une baisse de précision et de score F1, à 74.0% et 73.6% respectivement. Cependant, la taille de ce modèle est encore réduite à 235 Ko, offrant un compromis entre performance et encombrement.

Conclusion : Dans l'ensemble, le tableau 5.3 illustre l'efficacité des techniques de restructuration, d'élagage et de quantification appliquées à notre modèle. Ces optimisations ont conduit à une réduction substantielle de la taille du modèle et du nombre de paramètres, sans compromettre significativement la précision ou le score F1. Ceci témoigne de l'équilibre réussi entre les performances et l'efficacité en termes de ressources.

5.4.2 Comparaison avec l'état de l'art

Le tableau 5.4 présente les performances du modèle optimisé en comparaison avec le modèle initial du chapitre précédent et également présenté dans [171], et avec les modèles de l'état de l'art entrainés sur les jeux de données CK+ et FER2013+.

Configuration expérimentale : Les modèles optimisés ont été testés sur un microcontrôleur STM32H743ZI, qui fonctionne sur la plateforme NUCLEO-H743ZI2 et est de type Cortex-M7 opérant à une fréquence de 480MHz. Tous les modèles traitent des images de $48 \times 48 \times 3$ présentées à leur entrée. Cependant, leur entraînement diffère en termes de jeux de données : le modèle restructuré a été entraîné uniquement sur CK+, tandis que le modèle optimisé a été entraîné sur CK+ et FER2013+.

Performances : Le modèle proposé présente de bonnes performances sur les jeux de données CK+ et FER2013+, conformément aux résultats du tableau 5.4. Sur le jeu de données CK+, le modèle proposé atteint une précision de 97.96%, rivalisant avec des modèles ayant un plus grand nombre de paramètres comme le modèle de Zhu [172] qui atteint 98.46%, mais avec 14M de paramètres. Sur le jeu de données FER2013+, le modèle optimisé excelle également avec une précision de 92.11%, surpassant d'autres modèles de l'état de l'art tels que celui de Zheng [173] qui atteint 91.62% avec un total de 71.8M

de paramètres. Ce qui est particulièrement remarquable est que notre modèle obtient ces résultats avec seulement 57K paramètres, indiquant une efficacité impressionnante.

Complexité et efficacité : Sur le plan de la complexité de calcul, le modèle optimisé nécessite moins de MAC (*Multiplications-Accumulations*) et de mémoire pour les poids et activations. Bien qu'il présente un temps d'inférence légèrement augmenté, son débit est supérieur, reflétant une meilleure performance en termes de traitements par seconde.

TABLE 5.4 – Analyse comparative avec les modèles de référence entraînés sur les ensembles de données CK+ et FER2013+. Les meilleures valeurs sont en gras. NA - Non disponible ; "M" signifie millions.

Méthode	**Jeu de données**		**#Paramètres**
	CK+	**FER2013+**	
[172]	**98.46**		14M
[4]	98.38		NA
Modèle proposé	97.96		**57K**
[174]	97.79		2M
[175]	95.10		NA
Modèle proposé		**92.11**	**57K**
[173]		91.62	71.8M
[6]		89.22	11M
[176]		89.53	NA
[177]		88.11	3.5M
[178]		86.84	1.45M

Le tableau 5.5 montré les résultats obtenus en termes de performances entre le modèle optimisé et le modèle initial avant l'optimisation, en se basant spécifiquement sur leurs implémentations sur le microcontrôleur STM32H743ZI. Ces résultats montrent l'efficacité du modèle optimisé pour une implantation embarquée. Avec une quantité de MAC considérablement réduite (10.7 M vs 110.3 M) et une utilisation de mémoire bien plus faible (378Ko vs 2.01Mo), le modèle optimisé est plus adapté pour des applications embarquées. Bien que le temps d'inférence soit légèrement plus élevé pour une utilisation temps réel (118 ms donne une vitesse de traitement inférieure à 10 fps), il est presque 10 fois plus rapide que le temps d'inférence du modèle initial. Par conséquent, l'efficacité globale est significativement améliorée, offrant une meilleure performance tout en conservant la pré-

cision. Cette comparaison directe met en évidence les avantages clairs de l'optimisation, surtout dans le contexte des dispositifs embarqués.

TABLE 5.5 – Comparaison des performances entre les modèles optimisé et initial [171], tous deux implémentés sur un microcontrôleur STM32H743ZI.

Réseau	Modèle proposé	Model initial [171]
Plateforme	NUCLEO-H743ZI2	NUCLEO-H743ZI2
Microcontrôleur	1×Cortex-M7 @480MHz	1×Cortex-M7 @480MHz
Cadre d'application	Cube.AI TFLite	Cube.AI TFLite
Jeu de données	CK+ / FER2013+	FER2013+
Taille de l'image d'entrée	$48 \times 48 \times 3$	$48 \times 48 \times 3$
Taille de la sortie	1×13	1×13
MAC	10 741 413	110 354 703
Mémoire (poids et act.)	378.17 Ko	2.01 Mo
Précision [%]	97.96 / 92.11	92.12
Temps/inférence [ms]	118.01	1101.58
Débit [GMAC/s]	91.02	100.18
Cycles/MACC	5.27	4.79

5.5 Conclusion

Dans ce chapitre, nous avons abordé un certain nombre d'aspects relatifs à l'optimisation des modèles d'apprentissage profond pour la reconnaissance des expressions faciales (FER) et la détection des actions unitaires (AU). Nous nous sommes particulièrement intéressés à l'adaptation des modèles pour les dispositifs embarqués, en mettant l'accent sur les contraintes spécifiques de ces systèmes en termes de mémoire et de puissance de calcul. Le point central de la démarche entreprise dans ce chapitre a été la nécessité d'équilibrer la taille du modèle et ses performances. L'utilisation des techniques d'optimisation telles que la restructuration de l'architecture du modèle, l'élagage et la quantification s'est avérée essentielle pour atteindre cet équilibre.

Dans le cadre de la comparaison avec les travaux issus de l'état de l'art, nous avons

constaté que notre modèle optimisé se distingue en termes de sa taille très réduite, étant le modèle FER le plus compact à ce jour, tout en conservant des performances de détection de très bon niveau, ceci étant les facteurs primordiaux pour une implantation embarquée sur des dispositifs à ressources limitées.

En conclusion, l'optimisation des modèles FER pour les dispositifs embarqués est un domaine de recherche en pleine évolution. Les techniques et approches discutées dans ce chapitre ont montré leur potentiel pour créer des modèles à la fois légers et performants. Le défi permanent sera de continuer à améliorer ces méthodes tout en prenant en compte l'évolution rapide des architectures de modèles et des besoins applicatifs notamment les aspects temps réel. Les approches FER adaptées pour une implantation embarquée ouvrent la porte à une multitude d'applications pratiques, de la surveillance de la santé à la robotique, en passant par les interfaces homme-machine plus intuitives.

Conclusion et perspectives

Résumé des résultats de la recherche

Les travaux de recherche présentés dans ce manuscrit s'insèrent dans le domaine de la reconnaissance d'expressions faciales, des unités d'action et d'émotions associées connu sous le nom de FER. Dans ce cadre, nous avons eu pour objectif de proposer, développer, tester et valider des modèles s'insérant dans les tendances de recherche actuelles, c'est à dire à base de l'apprentissage profond. Les modèles qui ont été proposés et validés dans cette thèse devaient répondre à un certain nombre de critères : basés sur les réseaux CNN ; performants en termes de métriques les plus courantes sur un ensemble de jeux de données différents ; efficaces et de faible complexité se résumant principalement par un faible nombre de paramètres ; et interprétables permettant de fournir des arguments supplémentaires pour la compréhension des décisions fournies en se basant sur les travaux de référence dans le domaine.

Dans ce manuscrit, nous avons présenté plusieurs modèles permettant d'aller de manière progressive vers l'objectif final de la thèse : un premier modèle présenté dans le chapitre 2 d'une complexité faible permettant de détecter les émotions et d'interpréter les résultats obtenus à l'aide des approches de visualisation ; un modèle amélioré et enrichi (présenté dans le chapitre 3) avec les unités d'action, permettant non seulement de détecter les émotions avec une meilleure précision et plus de robustesse mais également de détecter les unités d'action associées à chaque émotion et donnant des arguments supplé-

mentaires pour son intérpretation (à côté des outils de visualition) ; un nouveau modèle, présenté dans le chapitre 4, basé sur les modèles des chapitres 2 et 3 intégrant les mécanismes d'attention spatiale et par canal permettant de focaliser la détection des émotions et des unités d'action associées à des régions des images d'entrées les plus pertinentes ; enfin, un modèle optimisé (présenté dans le chapitre 5) et épuré permettant une implantation embarquée tout en maintenant les performances des modèles initiaux présentés dans les chapitres précédents.

L'apport principal des travaux de recherche menés dans le cadre de cette thèse réside dans la conceptualisation et la mise en œuvre d'un modèle CNN allégé pour la reconnaissance des expressions faciales et leur interprétation. Les contributions spécifiques peuvent être résumées comme suit :

— **Modèle FER de faible complexité et frugal en ressources** : Le modèle final présenté a démontré une performance supérieure par rapport à l'état de l'art tout en utilisant un nombre significativement réduit de paramètres, ce qui le rend idéalement adapté pour une mise en œuvre dans des environnements et systèmes contraints en termes de ressources (mémoire, de calcul).

— **Modèle FER interprétable** : Le modèle conçu va au-delà de la simple prédiction d'activation des AU en fournissant des informations basées sur les régions les plus pertinentes des images d'entrée à l'aide des outils de visualisation communs justifiant les prédictions du modèle, et améliorant ainsi leur interprétabilité et relations avec les émotions associées. Cette propriété permet de donner confiance en résultats fournis par le modèle en les comparant par rapport au cadre de référence connu pour chaque émotion.

Perspectives de recherche

En dépit des avancées significatives réalisées dans ce travail de thèse, il est essentiel de reconnaître les limitations existantes et d'identifier les opportunités pour les recherches futures. L'une des pistes à explorer est l'évaluation de différentes architectures et stratégies d'entraînement pour optimiser davantage les performances du modèle proposé, notamment dans des contextes de fonctionnement en temps réel ou en présence de divers types de bruits ou d'occlusions.

Par ailleurs, l'étude de l'interprétabilité du modèle à travers des techniques comme les cartes de saillance ou la propagation de la pertinence couche par couche pourrait faciliter une meilleure compréhension du processus décisionnel du modèle, et potentiellement conduire à une amélioration des performances. En somme, notre travail met en exergue le potentiel considérable des modèles CNN légers et de faible complexité basés sur l'attention pour la détection des AU et par conséquent des émotions associées, tout en soulignant l'importance d'une représentation adéquate des données d'entrée pour atteindre une performance optimale. Ces notamment en s'appuyant sur le modèle final présenté dans cette thèse que toutes les explorations futures dans le domaine de la FER destinées à une utilisation dans des systèmes embarqués et temps réel devraient s'effectuer.

Annexe A

Réseau de neurones à convolution : CNN

A.1 Réseau neuronal convolutionnel

Les réseaux neuronaux convolutifs (CNN) ont profondément transformé le domaine de l'apprentissage profond, en particulier dans les tâches de traitement d'images et de vidéos [157]. Initialement inspirés par le système visuel biologique, ils cherchent à automatiser l'extraction des caractéristiques essentielles à partir de données d'entrée hiérarchisées [179]. Leur architecture unique, composée de couches de convolution, de normalisation, de regroupement, et entièrement connectées, permet d'aborder efficacement les défis complexes liés à l'analyse d'images et de vidéos [180]. En adoptant une approche multicouche pour l'apprentissage des caractéristiques, ils ont réussi à établir des performances state-of-the-art dans diverses applications telles que la reconnaissance des objets, la détection des anomalies, et la segmentation d'images [181].

A.1.1 Couches de convolution

Les couches de convolution agissent comme des éléments fondamentaux dans le processus de reconnaissance des patterns dans les images. Elles sont primordiales pour l'identification et l'extraction des caractéristiques essentielles d'une image en utilisant des filtres de dimension $k \times k$ qui se déplacent à travers l'image [182]. Le produit de la convolution entre les filtres et l'image permet de générer des cartes de caractéristiques mettant en avant divers patterns et structures locales au sein de l'image [51]. Ces filtres fonctionnent à différentes échelles, permettant ainsi de capturer l'essence véritable de l'image.

La formule représentant l'opération de convolution est donnée par :

$$Y_{i,j} = \sum_{m} \sum_{n} X_{i+m,j+n} \cdot W_{m,n}, \tag{A.1}$$

où X est l'image d'entrée, W le filtre et Y la carte de caractéristiques résultante.

A.1.2 Regroupement des couches

Cette étape vise à diminuer la dimensionnalité des cartes de caractéristiques, tout en conservant les informations cruciales. Elle s'effectue par des opérations telles que la prise du maximum ou de la moyenne sur certaines régions des cartes de caractéristiques, facilitant ainsi la réduction du surajustement et la préservation des traits pertinents [183, 51].

L'opération de pooling maximale est définie par :

$$P_{i,j} = \max_{m \in [0,k-1], n \in [0,k-1]} X_{i \cdot s+m, j \cdot s+n}, \tag{A.2}$$

où k est la taille du filtre et s représente le pas.

A.1.3 Couches de normalisation

Les couches de normalisation jouent un rôle déterminant dans la stabilisation des activations provenant des couches précédentes, facilitant ainsi l'apprentissage du réseau neuronal [184]. Deux types principaux existent : la normalisation par lots et la normalisation par groupes, chacune ayant ses propres avantages en termes d'amélioration de la généralisation du modèle [185].

La normalisation par lots s'exprime par la formule :

$$\hat{X}_i = \frac{X_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \tag{A.3}$$

où $\hat{X}_i$ est la valeur normalisée, et μ_B, σ_B^2 et ϵ sont respectivement la moyenne, la variance et un terme pour éviter la division par zéro.

A.1.4 Couches entièrement connectées

Enfin, les couches entièrement connectées viennent en phase finale de la structure du réseau, servant essentiellement pour la classification [158]. Ces couches combinent les features extraits et aplatis des étapes antérieures pour effectuer des prédictions de classe.

L'opération se réalise via l'équation :

$$Y = W \cdot X + b, \tag{A.4}$$

où W et b sont les paramètres du modèle, et la multiplication et l'addition sont effectuées suivant les règles standard des opérations sur les matrices et les vecteurs.

En plus des couches convolutives, de pooling, de normalisation, et entièrement connectées précédemment discutées, les fonctions d'activation constituent un autre élément important dans la construction des réseaux neuronaux convolutifs (CNN). Elles travaillent en synergie avec ces couches pour améliorer la capacité d'apprentissage du réseau.

A.1.5 Fonctions d'activation

Les fonctions d'activation jouent un rôle clé dans les réseaux neuronaux convolutifs (CNN). Elles sont utilisées pour introduire la non-linéarité dans le réseau, ce qui permet au modèle d'apprendre et de réaliser des tâches plus complexes. Sans fonctions d'activation, peu importe la profondeur du réseau, celui-ci serait toujours un classifieur linéaire, limitant ainsi sa capacité à apprendre des motifs plus complexes [186].

ReLU (Rectified Linear Unit)

La fonction d'activation ReLU est la plus utilisée dans les CNN [187]. Elle est définie comme $f(x) = \max(0, x)$, c'est-à-dire qu'elle renvoie x si x est positif et 0 sinon. Elle présente l'avantage d'être facile à calculer et aide à atténuer le problème de disparition du gradient lors de l'entraînement des réseaux profonds.

$$f(x) = \max(0, x) \tag{A.5}$$

Sigmoïde

La fonction sigmoïde, définie par $f(x) = 1/(1 + e^{-x})$, est une autre fonction d'activation couramment utilisée. Elle présente l'avantage de transformer les entrées en valeurs comprises entre 0 et 1, ce qui peut être utile pour des tâches de classification binaire [186]. Cependant, elle est moins utilisée dans les CNN en raison du problème de disparition du gradient.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{A.6}$$

Tanh

La fonction Tanh, définie par $f(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x})$, est une autre fonction d'activation qui transforme les entrées en valeurs comprises entre -1 et 1. Cela peut permettre au réseau de mieux gérer les données négatives. Cependant, comme la fonction sigmoïde, elle peut souffrir du problème de disparition du gradient lors de l'entraînement de réseaux profonds [188].

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{A.7}$$

Softmax

Enfin, la fonction Softmax est couramment utilisée comme fonction d'activation dans la dernière couche d'un réseau de classification, car elle transforme les sorties en une distribution de probabilités sur les différentes classes [189].

$$f(\mathbf{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{A.8}$$

Comme illustré dans la Figure A.1, les différentes fonctions d'activation ont des propriétés uniques qui les rendent plus adaptées à certaines tâches par rapport à d'autres. ReLU, illustré dans la sous-figure (a), est couramment utilisé dans les couches convolutives et entièrement connectées des CNN pour introduire la non-linéarité dans le modèle. La Sigmoïde, montrée dans la sous-figure (b), est souvent utilisée pour les problèmes de classification binaire. Tanh, comme on peut le voir dans la sous-figure (c), est similaire à la Sigmoïde, mais sa plage de sortie est différente. Enfin, la Softmax, présentée dans la sous-figure (d), est utilisée dans la dernière couche d'un réseau de classification pour obtenir une distribution de probabilités sur les classes.

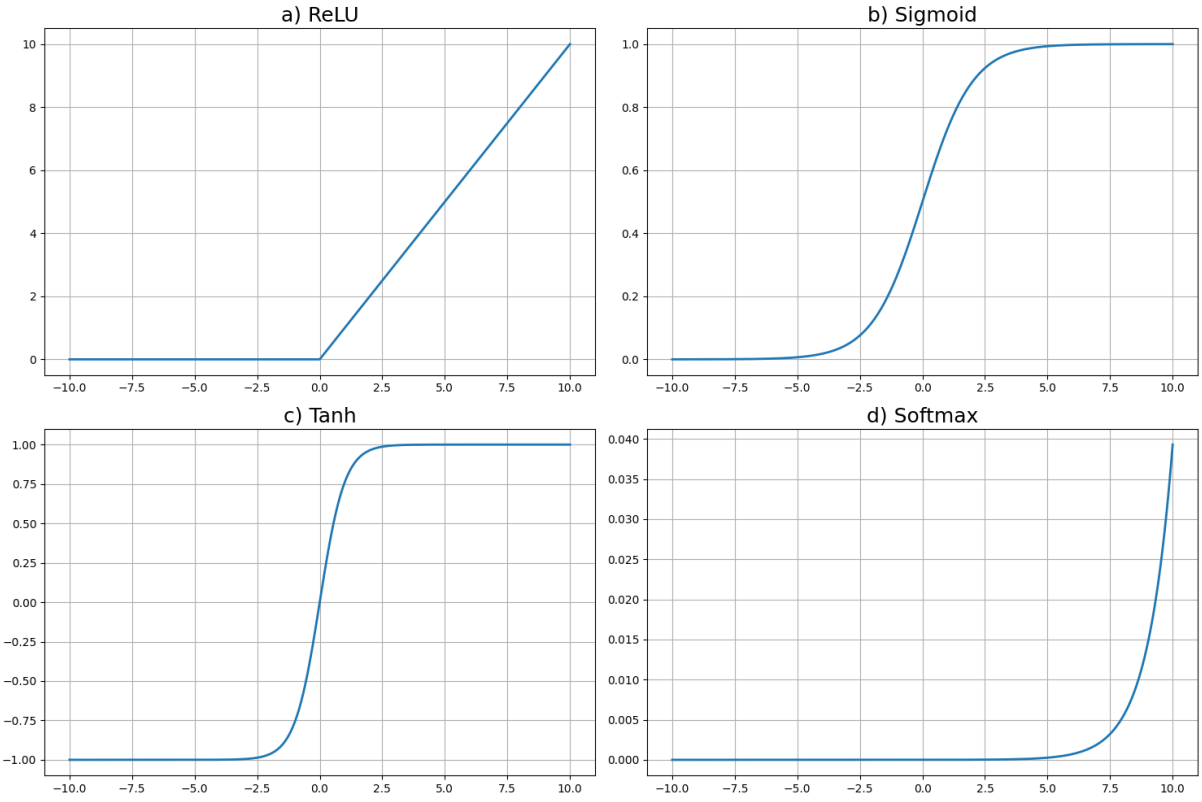


FIGURE A.1 – Représentations graphiques des fonctions d'activation couramment utilisées.

A.1.6 Critères d'évaluation dans les systèmes FER

Pour évaluer la performance d'un modèle de réseau neuronal profond (DNN) dans FER, il est nécessaire de définir des critères d'évaluation appropriés qui reflètent la précision et l'efficacité du modèle [190]. Cette section présente les métriques principales pour mesurer la performance de notre modèle et compare ses résultats avec les méthodes existantes.

Matrice de confusion

La matrice de confusion est un outil fondamental pour visualiser la performance d'un modèle de classification [191]. Elle affiche le nombre de prédictions correctes et incorrectes pour chaque classe, où $C_{i,j}$ représente le nombre de prédictions de la classe i comme classe j, et permet d'identifier les classes souvent confondues par le modèle [192].

$$C = \begin{bmatrix} C_{1,1} & C_{1,2} & \cdots & C_{1,n} \\ C_{2,1} & C_{2,2} & \cdots & C_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n,1} & C_{n,2} & \cdots & C_{n,n} \end{bmatrix}$$

Exactitude (Accuracy) pour la FER

L'exactitude est une mesure couramment employée pour évaluer la proportion de prédictions correctes. Elle est définie comme le rapport entre la somme des valeurs diagonales de la matrice de confusion, $\sum_{i=1}^{n} C_{i,i}$, et la somme totale des éléments de la matrice $\sum_{i=1}^{n}\sum_{j=1}^{n} C_{i,j}$ [193].

$$\text{Accuracy} = \frac{\sum_{i=1}^{n} C_{i,i}}{\sum_{i=1}^{n}\sum_{j=1}^{n} C_{i,j}} \tag{A.9}$$

Fonction de perte et optimisation

Entropie croisée Utilisée pour la classification multi-classes, l'entropie croisée mesure la distance entre la distribution de probabilité prédite, $\hat{y}$, et celle de la vérité terrain, y, en utilisant les logarithmes des probabilités prédites [77].

$$L(y, \hat{y}) = -\sum_{i} y_i \log(\hat{y}_i) \tag{A.10}$$

Erreur quadratique moyenne (MSE) et Erreur absolue moyenne (MAE) Le MSE et le MAE quantifient la différence entre les valeurs prédites, $\hat{y}$, et les valeurs réelles, y, sur n échantillons. Le MSE pénalise davantage les grandes erreurs, tandis que le MAE est plus robuste aux valeurs aberrantes [194].

$$L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}i)^2 \tag{A.11}$$

$$L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}i| \tag{A.12}$$

Optimisation L'optimisation joue un rôle central dans la formation d'un CNN, car elle permet d'ajuster les poids et les biais du réseau pour minimiser la fonction de perte. Les algorithmes d'optimisation utilisés pour former les CNN sont généralement basés sur la descente de gradient, qui met à jour les paramètres du modèle en suivant la direction du gradient négatif de la fonction de perte par rapport à ces paramètres [77, 195].

Descente de gradient stochastique (SGD) La descente de gradient stochastique (SGD) est une variante de la descente de gradient qui met à jour les paramètres du modèle à l'aide d'un seul exemple d'entraînement à chaque itération [195]. Cette approche réduit le coût de calcul par rapport à la descente de gradient par lots, où l'ensemble du lot d'apprentissage est utilisé pour calculer le gradient. La formule de mise à jour du SGD

est la suivante :

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t; x_t, y_t), \tag{A.13}$$

où θ_t représente les paramètres du modèle à l'étape t, η est le taux d'apprentissage, $\nabla L(\theta_t; x_t, y_t)$ est le gradient de la fonction de perte par rapport à θ_t paramètres pour l'exemple d'entraînement (x_t, y_t).

Élan Momentum est une technique d'optimisation qui accélère la convergence SGD en accumulant les gradients passés et en les utilisant pour mettre à jour les paramètres du modèle [196]. Cette approche permet au modèle de naviguer plus rapidement dans les régions où la surface de la fonction de perte est peu profonde et de réduire les oscillations dans les directions où la surface est raide. Momentum est généralement utilisé avec SGD, et la formule de mise à jour est :

$$v_{t+1} = \mu v_t + \eta \nabla L(\theta_t; x_t, y_t),\ \theta_{t+1} = \theta_t - v_{t+1}, \tag{A.14}$$

où v_t est la vitesse de mise à jour à l'étape t, et μ est le coefficient de quantité de mouvement.

Adam Adam (Adaptive Moment Estimation) est un algorithme d'optimisation qui combine les idées de momentum et d'adaptation du taux d'apprentissage pour chaque paramètre individuel [197]. L'Adam maintient une estimation des premier et deuxième moments des gradients passés et ajuste le taux d'apprentissage pour chaque paramètre en fonction de ces estimations. Il a été démontré que l'Adam converge plus rapidement et atteint de meilleures performances que le SGD avec un élan dans de nombreux scénarios d'apprentissage en profondeur.

RMSprop RMSprop (Root Mean Square propagation) est un autre algorithme d'optimisation adaptatif qui ajuste le taux d'apprentissage de chaque paramètre en fonction de l'ampleur des gradients passés [198]. Il maintient une estimation du carré moyen des gradients et divise le taux d'apprentissage de chaque paramètre par la racine carrée de cette estimation. Cela évite les oscillations dans les directions à gradient élevé et accélère la convergence. La formule de mise à jour pour RMSprop est :

$$g_{t+1} = \gamma g_t + (1-\gamma)(\nabla L(\theta_t; x_t, y_t))^2, \ \theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{g_{t+1} + \epsilon}} \nabla L(\theta_t; x_t, y_t), \tag{A.15}$$

où g_t est l'estimation quadratique moyenne des gradients à l'étape t, γ est le facteur de décroissance et ϵ est un terme de lissage pour éviter la division par zéro.

Courbe de la fonction de perte pour FER

La courbe de la fonction de perte est un outil graphique qui permet de visualiser l'évolution de la perte (erreur) du modèle de classification des émotions au fil des époques d'apprentissage [199]. Elle est utilisée pour évaluer la convergence du modèle FER et détecter les problèmes éventuels tels que le surapprentissage ou le sous-apprentissage [200]. La courbe de la fonction de perte montre généralement la perte d'entraînement et la perte de validation en fonction du nombre d'époques.

En observant la courbe de la fonction de perte pour FER, on peut déterminer si le modèle apprend correctement les données et si les ajustements d'hyperparamètres sont nécessaires [201]. Par exemple, si la courbe de la perte d'entraînement diminue rapidement mais que la courbe de la perte de validation stagne ou augmente, cela peut indiquer un surapprentissage [202]. Dans ce cas, des ajustements tels que la réduction de la capacité du modèle, l'ajout de la régularisation, ou l'augmentation de la taille de l'ensemble de données peuvent être nécessaires pour améliorer la performance du modèle sur les données

de validation [203].

Annexe B

Intelligence artificielle explicable :XAI

B.1 Interprétabilité des modèles FER : Enjeux et approches

L'interprétabilité des modèles FER est cruciale pour garantir la fiabilité et la transparence de ces systèmes et faciliter leur adoption dans divers domaines. Les modèles d'apprentissage profond, tels que les réseaux CNN, ont prouvé leur efficacité en matière de FER. Cependant, leur nature de boîte noire rend difficile la compréhension de leur fonctionnement interne et de leur processus de prise de décision.

Le schéma principal d'un système FER, intégrant à la fois un bloc d'attention et un algorithme d'Intelligence Artificielle Explicable (XAI), est illustré à la figure B.1.

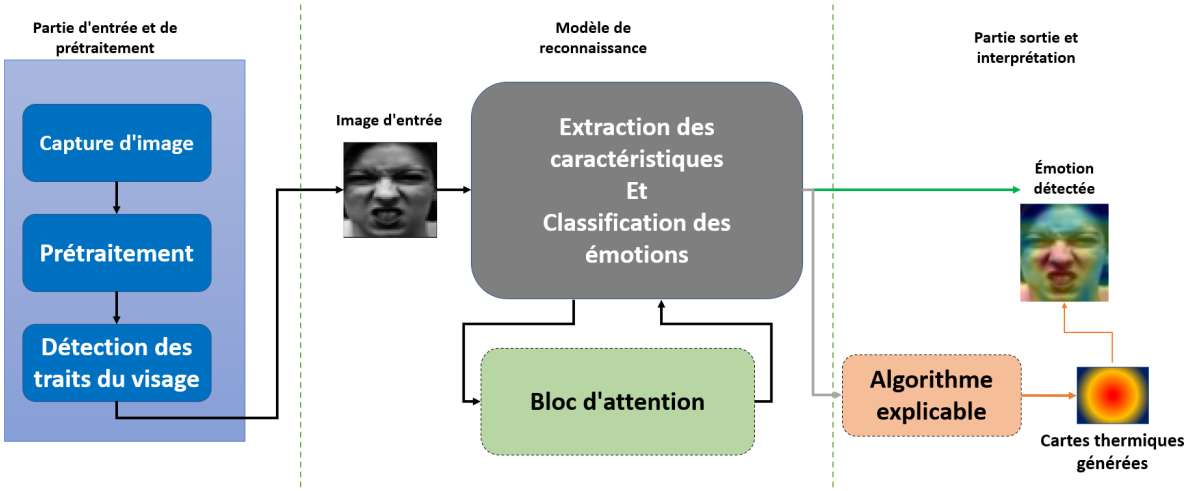


FIGURE B.1 – Schéma général d'un système de Reconnaissance des Expressions Faciales (FER) intégrant un bloc d'attention et un algorithme d'Intelligence Artificielle Explicable (XAI). Ce schéma illustre le processus typique d'un système FER, qui comprend les étapes principales de la capture d'image, du prétraitement, de la détection des traits du visage, de l'extraction des caractéristiques, de l'attention portée aux caractéristiques pertinentes, l'intégration d'un algorithme de XAI, et finalement la classification des émotions.

B.1.1 XAI dans les systèmes FER

Plusieurs approches récentes abordent ce problème en proposant différentes méthodes pour améliorer l'interprétabilité des modèles FER :

Décomposition des contributions des caractéristiques : Une approche consiste à décomposer les contributions des différentes caractéristiques d'entrée à la prédiction finale du modèle. Lundberg et Lee [112] ont proposé la méthode SHAP (SHapley Additive exPlanations), qui attribue une importance à chaque caractéristique en fonction de sa contribution à la prédiction.

Visualisation des cartes d'activation : Les cartes d'activation permettent de visualiser les régions de l'image d'entrée auxquelles le modèle accorde de l'importance pour faire la prédiction. Des méthodes telles que Grad-CAM [113] et LIME [114] ont été développées pour générer ces visualisations et fournir des informations sur les parties du visage pertinentes pour la prédiction d'émotions.

Modèles explicatifs : Les modèles explicatifs, tels que les arbres de décision, les réseaux bayésiens et les modèles linéaires généralisés, sont plus faciles à interpréter que les modèles d'apprentissage profond. Ils peuvent être utilisés comme approximations de modèles FER plus complexes pour mieux comprendre le processus de prise de décision [115].

Techniques d'explication post-hoc : Ces techniques sont appliquées après que le modèle a été entraîné pour expliquer ses décisions. Par exemple, Anchors [116] est une méthode post-hoc qui explique les prédictions d'un modèle en identifiant les conditions minimales suffisantes pour qu'une prédiction donnée soit valable.

Zhang et al. [117] ont introduit un modèle FER interprétable basé sur la décomposition des régions faciales en parties significatives et l'apprentissage de caractéristiques distinctes pour chacune de ces parties. Cette approche permet de mieux comprendre comment les différentes régions du visage contribuent à la prédiction d'émotions et offre une représentation plus explicative du processus de décision du modèle.

Dans une étude de Liu et al. [118], des mécanismes d'attention ont été utilisés pour améliorer l'interprétabilité des systèmes FER. Cette approche permet d'identifier les parties les plus importantes du visage pour la détection des expressions de la douleur. Les mécanismes d'attention ont également été appliqués avec succès dans la détection des

unités d'action (AU), qui sont des mouvements musculaires faciaux spécifiques associés à des expressions faciales émotionnelles[119].

L'ajout de mécanismes d'attention aux systèmes FER permet non seulement d'améliorer les performances de ces modèles, mais aussi de fournir des informations précieuses sur les régions du visage qui sont les plus discriminantes pour chaque expression émotionnelle. En mettant en évidence les zones clés du visage, les mécanismes d'attention facilitent l'interprétation des résultats des modèles FER et aident les chercheurs et les praticiens à mieux comprendre le processus de prise de décision sous-jacent.

En outre, l'utilisation de techniques d'interprétabilité dans les systèmes FER peut également contribuer à réduire les biais potentiels et à améliorer l'équité des modèles. En identifiant les caractéristiques discriminantes et les régions faciales importantes pour la prédiction des expressions, il est possible de détecter les biais inhérents aux données d'entraînement ou aux modèles et de prendre des mesures pour les atténuer.

Détails techniques des algorithmes d'explicabilité (XAI)

Dans cette section, nous approfondissons les détails techniques de trois algorithmes XAI couramment utilisés : LIME, SHAP et Grad-CAM. Ces algorithmes ont été appliqués avec succès dans divers domaines, y compris les systèmes de FER, pour offrir une compréhension approfondie du processus de prise de décision des modèles d'apprentissage automatique. Chacun de ces algorithmes présente des avantages et des limitations uniques, ainsi que des approches différentes pour expliquer les prédictions des modèles.

LIME (Local Interpretable Model-agnostic Explanations)

LIME [114] est une méthode d'explication locale qui vise à expliquer les prédictions d'un modèle complexe en utilisant un modèle linéaire simple. Elle fonctionne en trois étapes principales :

1. Perturbation : Générer un ensemble de points de données perturbés autour de

l'échantillon d'intérêt x. Ces points perturbés sont représentés par x'.

2. Poids : Calculer les poids $\pi_x(x')$ pour chaque point perturbé x' en utilisant une mesure de distance, généralement une distance euclidienne pondérée par une largeur de bande (par exemple, une fonction de noyau exponentielle).

3. Apprentissage du modèle linéaire local : Entraîner un modèle linéaire simple g sur l'ensemble des points perturbés en utilisant les poids $\pi_x(x')$ et en minimisant la fonction de perte L.

Les coefficients du modèle linéaire simple g représentent l'importance des caractéristiques pour la prédiction de l'échantillon d'intérêt x. L'approximation locale est obtenue en minimisant la fonction de perte pondérée suivante :

$$\xi(x) = \arg\min_{g \in G} \sum_i \pi_x(z_i)(f(z_i) - g(z_i))^2 + \Omega(g) \tag{B.1}$$

où z_i sont des points d'entrée pondérés par une mesure de proximité $\pi_x(z_i)$, G est l'ensemble des modèles linéaires simples, et $\Omega(g)$ est une fonction de complexité du modèle linéaire g. Cette méthode permet d'expliquer localement les prédictions d'un modèle complexe, tel qu'un réseau de neurones, en utilisant un modèle linéaire simple, plus facile à interpréter.

SHAP (SHapley Additive exPlanations)

L'algorithme SHAP [112] est une méthode d'explication globale qui attribue une importance à chaque caractéristique en utilisant les valeurs de Shapley, issues de la théorie des jeux coopératifs. Bien que le calcul des valeurs de Shapley nécessite un temps exponentiel en raison du grand nombre de combinaisons possibles de sous-ensembles de caractéristiques, des approximations et des méthodes spécifiques au modèle ont été développées pour accélérer le processus. Par exemple, TreeSHAP est conçu pour les modèles basés sur des arbres, et DeepSHAP pour les réseaux de neurones profonds.

La méthode SHAP attribue une valeur d'importance $\phi_i(f)$ à chaque caractéristique d'entrée i en utilisant l'équation suivante :

$$\phi_i(f) = \sum_{S \subseteq N \setminus i} \frac{|S|!(|N| - |S| - 1)!}{|N|!}[f(S \cup i) - f(S)] \tag{B.2}$$

où N est l'ensemble de toutes les caractéristiques d'entrée, S est un sous-ensemble de caractéristiques sans la caractéristique i, et $f(S)$ est la valeur prévue par le modèle pour le sous-ensemble S. Les valeurs SHAP permettent d'expliquer globalement l'importance des différentes caractéristiques dans les prédictions du modèle, facilitant ainsi l'interprétation et la compréhension des modèles complexes.

Grad-CAM (Gradient-weighted Class Activation Mapping)

L'algorithme Grad-CAM [113] est une méthode d'interprétation visuelle qui met en évidence les régions de l'image d'entrée importantes pour la prédiction d'une classe d'intérêt. Grad-CAM peut être appliqué à diverses architectures de réseaux de neurones, y compris les réseaux de neurones convolutionnels (CNN) et les réseaux de neurones récurrents (RNN). La technique fonctionne en plusieurs étapes :

1. Calcul du gradient : Calculer le gradient de la sortie de la classe d'intérêt c par rapport aux cartes d'activation A^k d'une couche de convolution spécifiée.

2. Poids du gradient moyen : Pour chaque canal k, calculer le poids du gradient moyen α_k^c comme la moyenne des gradients pour toutes les unités de la carte d'activation A^k.

3. Carte d'activation pondérée : Obtenir la carte d'activation pondérée par le gradient $L^c_{Grad-CAM}$ en effectuant une somme pondérée des cartes d'activation A^k avec les poids du gradient moyen α_k^c.

4. ReLU : Appliquer la fonction d'activation rectifiée linéaire (ReLU) pour ne conserver que les régions de la carte d'activation pondérée qui ont une influence positive

sur la classe d'intérêt.

Grad-CAM génère une carte d'activation $L^c_{Grad-CAM}$ pour une classe spécifique c en utilisant la formule suivante :

$$L^c_{Grad-CAM} = ReLU\left(\sum_k \alpha^c_k A^k\right) \tag{B.3}$$

où α^c_k est le poids attribué à la carte d'activation k pour la classe c, calculé comme la moyenne pondérée du gradient :

$$\alpha^c_k = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}} \tag{B.4}$$

et $ReLU$ est la fonction d'activation rectifiée, appliquée pour ne conserver que les activations positives.

La carte d'activation résultante $L^c_{Grad-CAM}$ aide à visualiser les parties de l'image sur lesquelles le modèle se concentre pour prendre sa décision. Grad-CAM peut également être étendu pour expliquer les modèles de traitement de texte, tels que les transformateurs, en adaptant la méthode pour visualiser l'importance des mots ou des n-grammes dans une séquence.

Annexe C

Mécanismes d'attention

C.1 Mécanismes d'attention dans les CNN

Le mécanisme d'attention, inspiré de l'attention visuelle humaine, permet aux réseaux neuronaux de se concentrer sur des parties spécifiques de l'entrée pertinentes pour la tâche, en ignorant les informations non pertinentes. Ce mécanisme a été particulièrement influent dans l'apprentissage profond, offrant un moyen de peser l'importance des différentes caractéristiques dans une entrée [155]. Dans le contexte des CNN, les mécanismes d'attention ont été intégrés pour améliorer la puissance discriminative des caractéristiques au sein des images [204]. En se concentrant sur des parties essentielles de l'image et en allouant plus de ressources de traitement à ces régions, les mécanismes d'attention ont considérablement amélioré les performances de diverses tâches de reconnaissance visuelle, telles que la détection d'objets, la segmentation et la reconnaissance des expressions faciales [205].

Le mécanisme d'attention permet au modèle de se concentrer sur les caractéristiques les plus pertinentes pour la tâche de détection des AUs, améliorant ainsi ses performances sans nécessiter une augmentation significative de la complexité de l'architecture. [206] Cette approche nous permet de créer un modèle équilibré entre la simplicité et l'efficacité, tout en maintenant les avantages d'un modèle léger pour la détection des AU.

Le mécanisme d'attention a révolutionné divers domaines tels que la vision par ordinateur et le traitement du langage naturel, permettant aux modèles de se concentrer sur des parties spécifiques des données pour améliorer les performances et la précision [207]. En utilisant cette technique dans notre modèle, nous visons à créer un système de détection AU plus précis et robuste qui peut s'adapter aux différentes conditions et expressions faciales rencontrées dans les images.

C.1.1 La mécanisme d'attention en FER

L'état de l'art dans ce domaine montre une diversité de méthodes d'attention appliquées aux systèmes FER. Par exemple, l'approche combinée de Liao et al. qui fusionne l'attention avec des motifs binaires locaux (LBP) [176], ou l'introduction de réseaux multi-régions avec différentes attentions par Guo et al. [208]. Des méthodes innovantes comme le transformateur augmenté avec correction d'étiquettes en ligne [209], ou l'introduction d'un réseau d'attention efficace pour une reconnaissance en temps réel par Kong et al. [210] illustrent l'évolution rapide de cette technologie. La recherche dans ce domaine continue de progresser, offrant de nouvelles opportunités pour une reconnaissance plus précise et efficace des expressions faciales [33].

C.1.2 Mécanismes d'attention générale

Les mécanismes d'attention générale aident les réseaux de neurones à se concentrer sur des éléments importants de l'image [211]. L'attention globale et locale sont appliquées dans les CNN. La formule d'attention générale est :

$$A(x) = \sum_{i=1}^{n} \alpha_i x_i \tag{C.1}$$

Des exemples incluent l'attention softmax [155] et l'attention basée sur l'entropie [212].

C.1.3 Attention à base de mémoire

L'attention basée sur la mémoire est un type de mécanisme d'attention qui utilise la mémoire externe pour améliorer les performances des modèles de réseaux de neurones, en particulier dans les tâches de traitement séquentiel et de raisonnement des données [213]. Les modèles d'attention basés sur la mémoire sont capables de stocker, de lire et de mettre à jour des informations dans la mémoire externe, ce qui leur permet d'apprendre des représentations plus complexes et de mieux traiter les relations à long terme entre les

données [214, 215].

Dans l'attention basée sur la mémoire, la formule d'attention généralement utilisée pour lire les informations de la mémoire externe est :

$$a_t = \text{softmax}(M_t \cdot k_t), \tag{C.2}$$

où a_t est le vecteur d'attention au pas de temps t, M_t est la mémoire externe au pas t et k_t est le vecteur clé utilisé pour interroger la mémoire externe. La mémoire externe est généralement représentée par une matrice $M_t \in \mathbb{R}^{N \times d}$, où N est le nombre d'emplacements mémoire et d est la dimension de chaque emplacement mémoire. Le vecteur clé k_t est souvent dérivé de l'état caché du modèle au pas de temps t. Le vecteur d'attention a_t est utilisé pour pondérer les emplacements de mémoire et lire les informations de la mémoire externe comme suit :

$$r_t = M_t^T \cdot a_t, \tag{C.3}$$

où r_t est le vecteur lu depuis la mémoire externe au pas de temps t. Ce vecteur de lecture est ensuite utilisé pour mettre à jour l'état caché du modèle et effectuer d'autres opérations, telles que la génération de prédictions ou la mise à jour de la mémoire externe elle-même [214, 215].

Les réseaux de mémoire à long court terme (LSTM) sont un exemple d'attention basée sur la mémoire utilisée dans les réseaux de neurones récurrents (RNN). Les LSTM sont capables de stocker et de récupérer des informations sur de longues séquences temporelles, ce qui les rend particulièrement utiles pour le traitement du langage naturel et les tâches de prédiction de séquence [216]. Les LSTM utilisent des mécanismes d'attention pour se concentrer sur des parties importantes des données d'entrée et pour décider quelles informations doivent être stockées et mises à jour dans la mémoire [217].

Un autre exemple d'attention basée sur la mémoire est le Differentiable Memory Net-

work (DNC), qui est un type de réseau de neurones qui utilise la mémoire externe pour stocker et récupérer des informations de manière différentiable. Les DNC peuvent être utilisés pour des tâches complexes telles que la résolution de problèmes algorithmiques, la planification et le raisonnement des connaissances [215]. Les DNC utilisent des mécanismes d'attention pour lire et écrire des informations dans la mémoire externe, en se concentrant sur les parties pertinentes de la mémoire en fonction des besoins de la tâche en cours.

L'attention basée sur la mémoire a également été intégrée aux modèles de réseau neuronal convolutif CNN pour améliorer les performances des tâches de vision par ordinateur. Par exemple, l'attention basée sur la mémoire a été utilisée pour améliorer la segmentation de l'image en permettant au modèle de se concentrer sur les régions pertinentes de l'image et d'apprendre des représentations plus complexes de l'image [218].

C.1.4 Attention auto-régressive

L'attention autorégressive est une approche qui combine des mécanismes d'attention avec des modèles autorégressifs pour améliorer les performances et l'interprétabilité des systèmes de reconnaissance. Les modèles auto-régressifs prédisent une variable de sortie en fonction de ses propres valeurs passées, permettant l'intégration d'informations temporelles et spatiales pour une meilleure compréhension des données. Dans le contexte des réseaux de neurones, l'attention auto-régressive peut être utilisée pour modéliser des dépendances complexes entre différentes parties des entrées et des sorties du réseau [155, 93].

L'attention auto-régressive est généralement utilisée dans les architectures de type Transformer, où elle est intégrée dans la couche d'attention à plusieurs têtes. La formule d'attention auto-régressive est :

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{C.4}$$

où Q, K et V sont respectivement les matrices des vecteurs requête, clé et valeur, et d_k est la dimension des vecteurs clé. Ces matrices sont calculées à partir des entrées du réseau à l'aide de poids appris. La fonction softmax est appliquée sur l'axe clé (K) pour obtenir un vecteur d'attention qui met en évidence les parties pertinentes des données. Enfin, ce vecteur d'attention est utilisé pour pondérer les valeurs (V) afin d'obtenir une représentation actualisée des données.

Pour rendre cette attention auto-régressive, une matrice de masquage est utilisée pour empêcher les informations futures d'affecter les prévisions actuelles. Cela se fait en modifiant la formule d'attention comme suit :

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T + M}{\sqrt{d_k}})V, \tag{C.5}$$

où M est la matrice de masquage, qui est généralement une matrice triangulaire supérieure avec des éléments négatifs infinis dans sa partie supérieure. Ce masquage garantit que les futurs éléments de la séquence ont un poids nul dans la fonction softmax et ne contribuent pas à la représentation mise à jour des données [155, 93].

Un exemple d'application de l'attention auto-régressive est le modèle Transformer, qui est largement utilisé pour le traitement du langage naturel et les tâches de vision par ordinateur. Le transformateur utilise une architecture d'attention à plusieurs têtes pour modéliser les dépendances entre les différentes parties des séquences d'entrée et de sortie [93]. Cette architecture permet au modèle d'apprendre des représentations hiérarchiques des données et d'améliorer les performances dans des tâches telles que la traduction automatique, la génération de texte et la reconnaissance d'objets [93, 219].

L'attention autorégressive a également été appliquée avec succès à la détection des émotions faciales et des unités d'action. Par exemple, une étude récente a utilisé un modèle Transformer pour détecter des unités d'action dans des vidéos faciales, exploitant l'attention autorégressive pour modéliser les dépendances temporelles et spatiales entre différentes parties du visage [220]. Cette approche a montré des performances supérieures

aux méthodes traditionnelles et aux autres architectures de réseaux de neurones pour la détection des unités d'action.

C.1.5 Attention spatiale et attention de canal

L'attention spatiale et l'attention du canal sont deux types d'attention couramment utilisés dans les CNN pour améliorer les performances et l'interprétabilité du modèle. L'attention spatiale se concentre sur des régions importantes de l'image en attribuant des poids différents aux éléments spatiaux, tandis que l'attention du canal se concentre sur des canaux de caractéristiques importants en attribuant des poids différents aux canaux d'image [221].

L'attention spatiale permet au modèle de se concentrer sur les zones clés de l'image qui sont pertinentes pour la tâche en cours, en mettant l'accent sur les régions spatiales les plus informatives. Cette approche peut être particulièrement utile pour les images complexes et les tâches où certaines parties de l'image sont plus importantes que d'autres pour la classification [222].

La formule d'attention spatiale peut être exprimée comme suit :

$$S = \sigma(f^{7\times7}(M_c(F))) \tag{C.6}$$

où M_c représente la sortie de l'opération d'attention de canal appliquée au tenseur d'entrée F, $f^{7\times7}$ représente une couche de convolution de 7x7, et σ représente la fonction d'activation sigmoïde.

L'attention des canaux, d'autre part, permet au modèle de se concentrer sur les canaux de fonctionnalités les plus importants en attribuant des poids différents aux canaux. Cette approche aide le modèle à identifier et à sélectionner les fonctionnalités les plus pertinentes pour la tâche en cours, en mettant l'accent sur les canaux de fonctionnalités [205] les plus informatifs.

La formule d'attention du canal peut être exprimée comme suit :

$$C = \sigma(MLP(M, A)) \tag{C.7}$$

où C représente les poids d'attention de canal, M et A représentent les sorties des opérations de max-pooling et d'average-pooling appliquées au tenseur d'entrée F, et σ représente la fonction d'activation sigmoid. Le MLP prend les sorties des opérations de max-pooling et d'average-pooling et produit les poids d'attention de canal.

Les mécanismes d'attention spatiale et de canal ont été utilisés dans diverses applications de vision par ordinateur, telles que la reconnaissance d'images, la segmentation d'images et la détection d'objets. Les modèles utilisant ces mécanismes d'attention ont obtenu de meilleurs résultats que les modèles traditionnels sans attention sur de nombreuses tâches [221, 205]. De plus, l'utilisation de l'attention spatiale et de l'attention du canal peut également améliorer l'interprétabilité des modèles en aidant à identifier les régions et les caractéristiques clés qui contribuent à la décision finale du modèle. Le schéma du canal et de la méthode d'attention spatiale est illustré à la figure C.1.

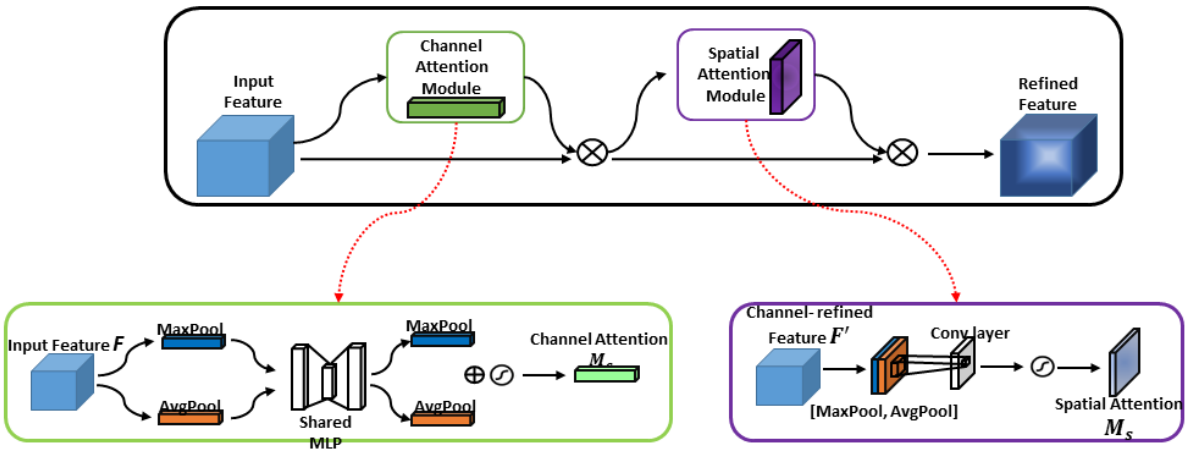


FIGURE C.1 – Schéma illustrant les modules d'attention de canal et spatiale. Chaque module est constitué de plusieurs couches incluant les opérations de moyenne et max-pooling, qui sont intégrées aux couches entièrement connectées pour calculer les poids correspondant à chaque carte de caractéristiques [221].

C.1.6 Mécanismes d'Attention dans les Systèmes FER

Les mécanismes d'attention dans les systèmes de FER ont acquis une importance cruciale dans la compréhension et l'interprétation des expressions faciales. En mettant l'accent sur des régions spécifiques du visage, ces mécanismes peuvent extraire des caractéristiques significatives qui sont essentielles à la discrimination des différentes émotions [176]. En outre, les mécanismes d'attention sont également pertinents pour la détection des AU, qui sont des mouvements fondamentaux du visage et servent de base à l'expression des émotions. La capacité de détecter ces unités d'action offre une compréhension plus approfondie des émotions et améliore la performance des systèmes FER [210].

Annexe D

Optimisation pour implantation embarquée

D.1 Optimisation dans les modèles profonds

L'optimisation d'un modèle profond est un processus essentiel qui englobe l'amélioration des performances et de l'efficacité d'un réseau de neurones [195]. Ce processus vise à atténuer les complications liées à une architecture de réseau large et profonde, telles que les longs temps d'entraînement et la surutilisation des ressources computationnelles. L'optimisation peut englober diverses techniques, y compris la régularisation, l'initialisation judicieuse des poids, et le choix d'algorithmes d'optimisation efficaces [77]. En essence, le but est de parvenir à un modèle qui est non seulement précis mais aussi économique en termes de ressources nécessaires, facilitant ainsi le déploiement dans des environnements avec des contraintes de ressources limitées [163].

Dans la section suivante, nous nous concentrerons sur les techniques spécifiques d'optimisation de l'architecture du réseau, qui visent à réduire la taille du modèle en éliminant les éléments non essentiels du réseau.

D.1.1 Optimisation de l'architecture avec l'élagage

La compression vise à atténuer ces problèmes en réduisant la taille des modèles tout en minimisant la perte de précision ou de performances [163]. L'élagage du réseau neuronal est une méthode de compression qui consiste à supprimer les pondérations d'un modèle formé [160]. Il permet de supprimer les neurones ou les poids inutiles, contribuant ainsi à alléger l'architecture du réseau [223].

Il existe différentes façons d'élaguer un réseau de neurones, notamment l'élagage structuré et non structuré [224]. Par défaut, les résultats sont généralement rapportés en utilisant un élagage non structuré [225]. La différence entre les deux réside dans le fait que des poids individuels ou des groupes de poids sont supprimés ensemble [226]. Cette différence a non seulement des implications sur les performances, mais affecte également la parcimonie maximale réalisable [227].

L'élagage, également connu sous le nom "pruning", est une technique d'optimisation de modèle couramment utilisée pour réduire la taille du modèle et améliorer l'efficacité des calculs. L'idée de base de l'élagage est de supprimer les poids du réseau qui contribuent le moins à la prédiction du modèle, c'est-à-dire les poids avec des valeurs absolues faibles [160].

La procédure générale pour l'élagage d'un réseau neuronal est la suivante :

— Entraîner le réseau de neurones jusqu'à ce qu'il atteigne des performances satisfaisantes.

— Évaluer l'importance de chaque pondération (ou groupe de pondérations) dans le réseau. Cette importance peut être mesurée de différentes manières, par exemple par la valeur absolue du poids, la contribution du poids au rendement du réseau, etc.

— Supprimer les pondérations (ou groupes de pondérations) les moins importantes du réseau. Cela peut être fait en mettant ces poids à zéro.

— Réentraîner le réseau avec les poids restants pour ajuster le réseau à la suppression des poids.

La formule mathématique pour l'élagage est généralement représentée comme suit :

$$w_i' = \begin{cases} w_i & \text{si } |w_i| > \theta \\ 0 & \text{sinon} \end{cases} \tag{D.1}$$

où w_i est le poids original, w_i' est le poids après l'élagage, et θ est un seuil préétabli. Seuls les poids qui ont une valeur absolue supérieure à θ sont conservés.

L'élagage peut être effectué à différents niveaux de granularité, y compris l'élagage du poids individuel, l'élagage du filtre de convolution (également appelé élagage structuré) et l'élagage des neurones [228].

L'élagage peut également être classé en élagage statique et dynamique. L'élagage statique se produit après l'entraînement du modèle, tandis que l'élagage dynamique se pro-

duit pendant l'entraînement du modèle. Dans notre cas, nous utilisons l'élagage dynamique qui est plus adapté aux architectures de réseaux profonds [227].

Dans cette section, nous aborderons deux types d'élagage pour optimiser les réseaux de neurones : l'élagage non structuré et l'élagage structuré. Les deux techniques ont leurs propres avantages et inconvénients en termes de performances et de mémoire [228, 229].

Élagage non structuré

L'élagage non structuré est une technique où les connexions de poids individuelles sont supprimées dans un réseau en étant réinitialisées à zéro. En conséquence, cela introduit des multiplications par zéro dans le réseau. Cette technique est avantageuse car elle permet aux logiciels d'exécuter les réseaux élagués beaucoup plus rapidement [160]. De plus, les fichiers de modèle peuvent être stockés de manière compressée, occupant ainsi beaucoup moins d'espace sur le disque.

$$w_{ij}^{'} = w_{ij} \cdot I(|w_{ij}| > \tau)$$

Ou, $w_{ij}^{'}$ est le nouveau poids, w_{ij} est le poids original, I est la fonction indicatrice, et τ est le seuil. Si la valeur absolue d'un poids est supérieure au seuil τ, le poids est conservé ; sinon, il est élagué (mis à zéro).

Élagage structuré

L'élagage structuré, en revanche, implique la suppression de groupes de connexions de poids ensemble, comme des canaux ou des filtres entiers [226]. Par conséquent, l'élagage structuré modifie les formes d'entrée et de sortie des couches et des matrices de poids. Presque tous les systèmes peuvent exécuter avec succès des réseaux structurellement élagués plus rapidement. Cependant, l'élagage structuré limite considérablement la parcimonie maximale qui peut être imposée à un réseau par rapport à l'élagage non structuré, ce qui limite considérablement les améliorations des performances et de la mémoire [230].

D'un point de vue pratique, cette différence est due au fait que l'élagage de groupes de poids et même de canaux entiers enlève de la flexibilité. Les connexions nécessaires dans les canaux devront être supprimées ainsi que celles qui ne sont pas importantes. D'un point de vue théorique, lors de l'élagage des canaux ou des filtres, la largeur de la couche (et du réseau global) est réduite, éloignant davantage le réseau du théorème d'approximation universelle et d'une approximation gaussienne [223, 163].

Cependant, on peut décrire l'élagage structuré en utilisant une forme mathématique plus générale. Soit un réseau de neurones avec un ensemble de poids $W = w_1, w_2, ..., w_n$. L'élagage structuré vise à trouver un sous-ensemble optimal $W' \subseteq W$ tel que le nombre de neurones (ou unités, ou filtres, selon le niveau d'élagage) élagués est maximisé tout en maintenant l'erreur de prédiction du réseau dans une certaine limite acceptable.

$$\begin{aligned} \text{Maximiser :} \quad & |W'| \\ \text{sujet à :} \quad & E(W') \leq \epsilon \end{aligned}$$

où $E(W')$ est l'erreur de prédiction du réseau après élagage et ϵ est l'erreur maximale acceptable. Il est important de noter que trouver la solution optimale à ce problème est généralement difficile et nécessite l'utilisation d'heuristiques ou d'approximations. L'une des méthodes les plus courantes est de classer les neurones (ou unités, ou filtres) en fonction de certaines mesures (par exemple, la norme L_1 des poids), puis d'élaguer les neurones avec les plus petites valeurs.

Comme illustré à la figure D.1, l'élagage non structuré et l'élagage structuré présentent des caractéristiques différentes. L'élagage non structuré supprime individuellement des poids, ce qui peut accélérer l'exécution du réseau et réduire la taille de stockage requise. L'élagage structuré, en revanche, supprime des groupes de poids, ce qui peut également accélérer l'exécution du réseau mais peut limiter la parcimonie maximale atteignable.

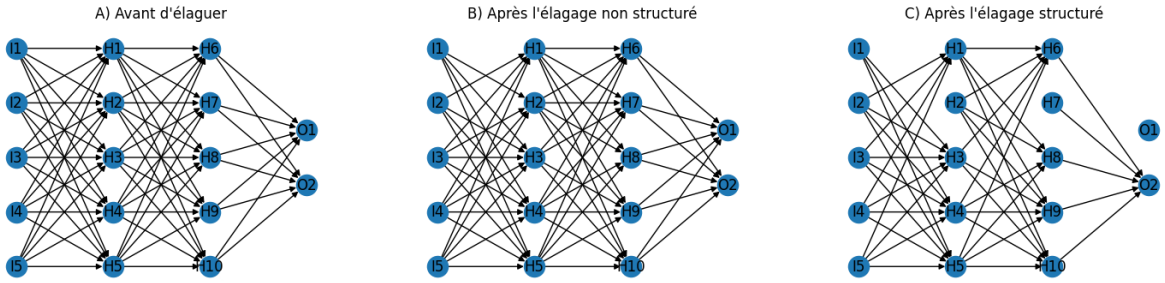


FIGURE D.1 – Comparaison des différentes techniques d'élagage sur une architecture neuronale. De gauche à droite : (A) Avant d'élaguer, tous les neurones sont connectés entre eux. (B) Après l'élagage non structuré, certains neurones spécifiques (par exemple I1-H1, I2-H2, H3-H8) ont été éliminés, ce qui peut entraîner un modèle plus efficace mais potentiellement moins précis. (C) Après l'élagage structuré, tous les neurones d'un nœud spécifique (par exemple, H2 et H7 dans les couches cachées, et O1 dans la couche de sortie) ont été supprimés, ce qui peut entraîner un modèle plus compact et plus facile à interpréter.

D.1.2 Optimisation de l'architecture avec la quantification

La quantification est une autre technique d'optimisation largement utilisée dans les réseaux de neurones. Elle est principalement utilisée pour réduire la précision des poids, des biais et des activations dans les réseaux de neurones, ce qui réduit la consommation de mémoire et améliore l'efficacité de calcul sans sacrifier de manière significative les performances du modèle.

La quantification des poids et des biais peut jouer un rôle crucial dans l'optimisation des modèles de CNN sous divers aspects. Ci-dessous, nous discutons de ses principaux avantages pour les CNN :

— Efficacité de stockage et d'inférence : les modèles de réseaux de neurones, y compris les CNN, sont généralement caractérisés par une grande quantité de poids et de biais. Ces paramètres peuvent consommer beaucoup de mémoire, en particulier dans le cas des CNN qui peuvent avoir des millions de paramètres. En appliquant la quantification, on peut réduire considérablement la taille de ces paramètres, diminuant ainsi les besoins de stockage. De plus, les opérations sur les nombres à précision réduite sont souvent plus rapides, ce qui peut accélérer le temps d'infé-

rence.

— Déploiement sur des appareils à ressources limitées : le déploiement de CNN sur des appareils à ressources limitées, tels que les smartphones ou les appareils IoT, peut être un défi en raison de leur capacité de stockage et de calcul limitée. La quantification peut aider à surmonter ces défis en réduisant la taille du modèle et en accélérant l'inférence. Cela rend les CNN plus accessibles et utilisables dans plus de scénarios.

— Résilience au surajustement : paradoxalement, la réduction de la précision des poids et des biais peut améliorer la capacité du modèle à généraliser à partir des données d'apprentissage. En effet, la quantification peut agir comme une forme de régularisation, limitant la capacité du modèle à mémoriser le bruit dans les données d'apprentissage et l'encourageant à apprendre les modèles sous-jacents.

Il convient de souligner l'importance d'une application judicieuse de la quantification. Si la précision est trop réduite, cela peut nuire aux performances du modèle. Ainsi, un équilibre doit être trouvé entre les avantages en termes d'efficacité de stockage et de calcul et le maintien de la précision du modèle.

C'est dans ce contexte que des méthodes plus sophistiquées, deviennent pertinentes. Ces méthodes permettent une quantification sélective des parties du modèle tout en préservant la précision dans les zones critiques. Nous approfondirons ces différentes techniques de quantification dans les sections suivantes.

Quantification des poids et des biais

Dans le contexte des réseaux de neurones, la quantification fait généralement référence à la méthode de réduction du nombre de bits qui représentent un nombre. Cela signifie que les valeurs de poids et de biais du réseau neuronal sont arrondies à un ensemble de valeurs discrètes, ce qui peut réduire la précision des valeurs mais accélérer le traitement et réduire les besoins en stockage et en mémoire.

La quantification des poids et des biais peut être formulée mathématiquement comme suit. Soit X une variable réelle à quantifier, et soit Q une fonction de quantification qui mappe X sur un ensemble de valeurs discrètes. Cette opération peut s'écrire comme suit :

$$Q(X) = round\left(\frac{X}{\Delta}\right) \cdot \Delta \tag{D.2}$$

Ici, Δ est une constante qui détermine l'échelle de quantification. Dans le cas de la quantification des poids, Δ peut être déterminée comme la plage de valeurs des poids divisée par le nombre de niveaux de quantification.

La quantification peut avoir un impact important sur la précision du modèle. Il a été démontré que la quantification à virgule flottante de 16 bits (également connue sous le nom de half-precision) peut atteindre une précision comparable à celle de la précision simple (32 bits) pour de nombreux modèles de réseaux de neurones profonds [164]. Cependant, la quantification à une précision inférieure, comme 8 bits ou moins, peut entraîner une perte de précision significative [231].

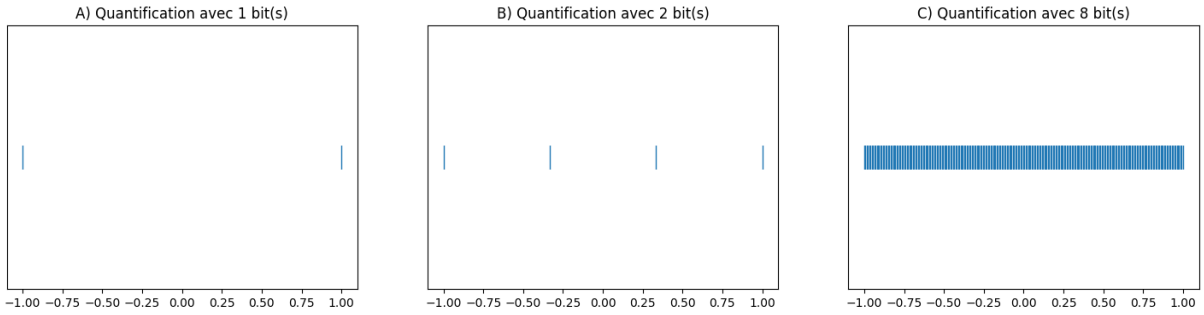


FIGURE D.2 – Illustration de la quantification avec différents nombres de bits. De gauche à droite : (A) Quantification avec 1 bit, (B) Quantification avec 2 bits, et (C) Quantification avec 8 bits. Chaque marque verticale représente une valeur de quantification possible.

La figure D.2 illustre comment le nombre de bits utilisés pour la quantification affecte la précision de la représentation des nombres. Dans chaque sous-figure, chaque marque verticale représente une valeur possible que peut prendre un nombre quantifié.

Dans la sous-figure A), la quantification est effectuée avec seulement 1 bit. Cela signifie

que seules deux valeurs sont possibles : -1 et 1. Cela limite considérablement la précision de la représentation des nombres, mais réduit également la quantité de mémoire requise.

La sous-figure B) montre la quantification à 2 bits, ce qui permet de représenter quatre valeurs différentes. On peut voir que la précision est doublée par rapport à la sous-figure a).

Enfin, la sous-figure C) montre la quantification avec 8 bits. Avec ce nombre de bits, 256 valeurs différentes peuvent être représentées. Cela donne une plus grande précision, mais nécessite également plus de mémoire.

Il est donc clair qu'un compromis doit être trouvé entre la précision de la représentation des nombres et la quantité de mémoire nécessaire. La quantification permet d'ajuster ce compromis en fonction des besoins spécifiques de chaque application.

D.1.3 Quantification des activations

La quantification des activations est un autre aspect crucial de l'optimisation de l'architecture des réseaux de neurones. La quantification des activations consiste à réduire la précision numérique des sorties des fonctions d'activation. Ce processus est similaire à la quantification des poids et des biais, la différence cruciale étant que les valeurs à quantifier ne sont pas statiques mais changent à chaque passage vers l'avant à travers le réseau.

Prenons l'exemple d'une fonction d'activation typique comme le ReLU (Rectified Linear Unit). La sortie de cette fonction pourrait potentiellement être n'importe quel nombre réel. En pratique, cependant, la plupart des implémentations de ReLU dans les réseaux de neurones limitent la sortie à une plage spécifique, telle que 0 à 6.

Dans le contexte de la quantification de validation, cette sortie serait alors quantifiée à un nombre fixe de bits. Par exemple, on pourrait utiliser une quantification à virgule fixe sur 8 bits, ce qui signifie que chaque sortie de la fonction d'activation serait représentée

par un entier sur 8 bits.

$$Q(X) \in \{-2^{b-1}, ..., 2^{b-1} - 1\} \quad \text{et} \quad X \in [-\alpha, \alpha]$$

Dans cette équation, Q est l'opérateur de quantification, X est la sortie de la fonction d'activation, b est le nombre de bits utilisés pour la quantification et α est le maximum absolu de X.

Cette méthode présente l'avantage de réduire considérablement le coût mémoire des activations, qui peut représenter une part importante de l'utilisation de la mémoire dans les réseaux de neurones profonds.

Cependant, la quantification des activations peut introduire des erreurs, car elle force la sortie de la fonction d'activation à prendre des valeurs discrètes plutôt que continues. Il est donc crucial de choisir une méthode de quantification qui minimise ces erreurs tout en offrant des avantages en termes de performances et d'utilisation de la mémoire [170].

Quantification de plage dynamique

La quantification de plage dynamique est une technique d'optimisation utilisée pour réduire la taille du modèle et améliorer les performances d'exécution. Cette méthode consiste à remplacer les valeurs à virgule flottante utilisées dans les poids et les activations par des valeurs à virgule fixe. La plage dynamique de ces valeurs est adaptée pour couvrir toutes les valeurs du modèle.

La quantification de plage dynamique fonctionne en réduisant le nombre de bits utilisés pour représenter les poids et les activations, généralement à 8 bits. Cette réduction de la précision permet de réduire la taille du modèle et d'accélérer les opérations, tout en maintenant une précision acceptable pour la plupart des applications. La plage dynamique est adaptée pour chaque couche, en fonction des valeurs maximales et minimales des poids et des activations.

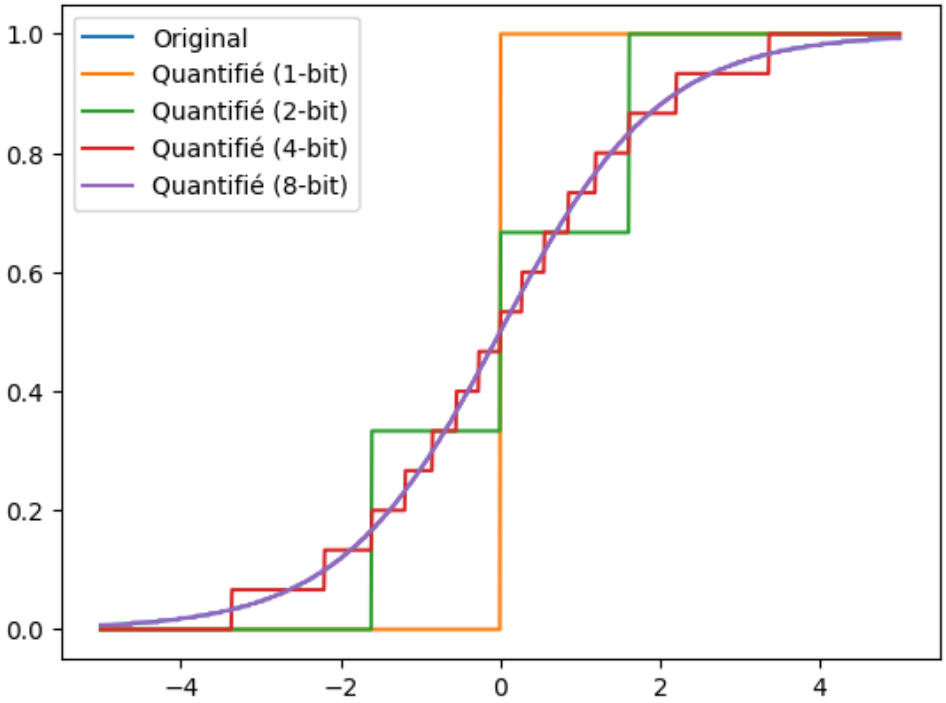


FIGURE D.3 – Effet de la quantification sur les activations de la fonction sigmoïde. La courbe bleue représente la fonction sigmoïde originale qui prend une gamme continue de valeurs. Les courbes restantes représentent la fonction sigmoïde après quantification à différentes largeurs de bit. On peut observer que, à mesure que la largeur de bit augmente (de 1 à 8 bits), la représentation quantifiée se rapproche de plus en plus de la fonction originale. Par exemple, avec une quantification à 1 bit (courbe orange), la fonction est réduite à une forme binaire, tandis qu'avec une quantification à 8 bits (courbe verte), la différence avec la fonction sigmoïde originale est presque indiscernable visuellement.

La formule générale pour la quantification de plage dynamique est :

$$Q(x) = \text{round}\left(\frac{x}{s}\right)$$

où $Q(x)$ est la valeur quantifiée, x est la valeur à virgule flottante originale, et s est l'échelle de quantification. L'échelle de quantification est déterminée en fonction de la plage dynamique des valeurs dans le modèle.

La quantification de plage dynamique est souvent utilisée en combinaison avec d'autres techniques d'optimisation, comme la quantification des poids et des activations, pour obtenir des gains de performance encore plus importants. Plusieurs travaux ont été réalisés

sur ce sujet, notamment par Krishnamoorthi [232].

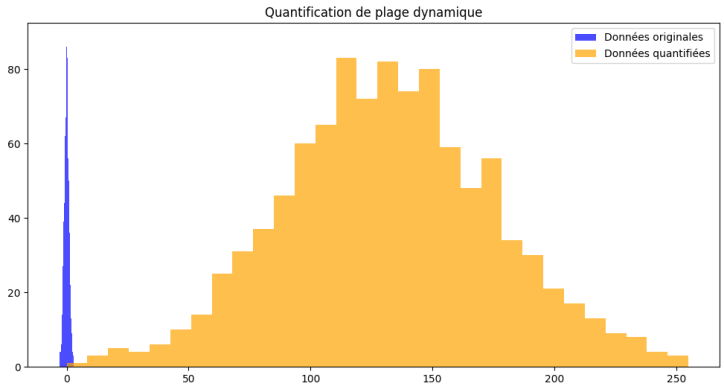


FIGURE D.4 – Comparaison de la distribution des valeurs pour les données originales et les données après la quantification de plage dynamique. On observe que la quantification réduit le nombre de valeurs uniques possibles (en orange), mais préserve la distribution globale des données.

La figure D.4 illustre comment la quantification de la plage dynamique agit sur les données de notre modèle. La courbe bleue représente la distribution des valeurs d'origine de la sortie d'une couche réseau. Après avoir appliqué la quantification de plage dynamique, les données sont transformées comme indiqué par la courbe orange. Il est clair que le nombre de valeurs possibles est fortement réduit, étant limité à un ensemble discret de points. Cependant, la distribution globale des données est préservée, ce qui signifie que les informations essentielles sont toujours là.

Cette observation est cruciale pour comprendre l'efficacité de la quantification de la plage dynamique. En préservant la distribution globale des données tout en réduisant l'espace de représentation, nous obtenons une version plus légère et plus rapide de notre modèle original, tout en conservant une précision adéquate.

Bibliographie

[1] Cuiting Xu, Chunchuan Yan, Mingzhe Jiang, Fayadh Alenezi, Adi Alhudhaif, Norah Alnaim, Kemal Polat, and Wanqing Wu. A novel facial emotion recognition method for stress inference of facial nerve paralysis patients. *Expert Systems with Applications*, 197 :116705, 2022-07.

[2] Zixuan Shangguan, Zhenyu Liu, Gang Li, Qiongqiong Chen, Zhijie Ding, and Bin Hu. Dual-Stream Multiple Instance Learning for Depression Detection With Facial Expression Videos. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31 :554–563, 2023.

[3] Amal Alabdulkareem, Noura Alhakbani, and Abeer Al-Nafjan. A systematic review of research on robot-assisted therapy for children with autism. *Sensors*, 22(3), 2022.

[4] Qianqian Chen, Xiaojun Jing, Fangpei Zhang, and Junsheng Mu. Facial Expression Recognition Based on A Lightweight CNN Model. In *2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–5, 2022-06.

[5] Hongxiang Gao, Min Wu, Zhenghua Chen, Yuwen Li, Xingyao Wang, Shan An, Jianqing Li, and Chengyu Liu. SSA-ICL : Multi-domain adaptive attention with intra-dataset continual learning for Facial expression recognition. *Neural Networks*, 158 :228–238, 2023.

[6] Shuang Qiu, Guangzhe Zhao, Xiao Li, and Xueping Wang. Facial expression recognition using local sliding window attention. *Sensors*, 23(7) :3424, 2023.

[7] Shuyi Mao, Xinpeng Li, Qingyang Wu, and Xiaojiang Peng. Au-aware vision transformers for biased facial expression recognition. *arXiv preprint arXiv :2211.06609*, 2022.

[8] Irene Kotsia, Stefanos Zafeiriou, and Spiros Fotopoulos. Affective Gaming : A Comprehensive Survey. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 663–670. IEEE, 2013-06.

[9] H. Kobayashi and F. Hara. Recognition of Six basic facial expression and their strength by neural network. *[1992] Proceedings IEEE International Workshop on Robot and Human Communication*, 1992.

[10] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–511–I–518. IEEE Comput. Soc, 2001.

[11] Najmeh Samadiani, Guangyan Huang, Borui Cai, Wei Luo, Chi-Hung Chi, Yong Xiang, and Jing He. A Review on Automatic Facial Expression Recognition Systems Assisted by Multimodal Sensor Data. *Sensors*, 19(8) :1863, 2019-01.

[12] Paul Ekman and Wallace V Friesen. *Facial Action Coding System : Manual*. Consulting Psychologists Press, 1978.

[13] Paul Ekman and Wallace V. Friesen. The facial action coding system. In Bernard Siegel and A. W. Siegman, editors, *Nonverbal Behavior and Communication*, pages 169–200, Hillsdale, NJ, 1978. Lawrence Erlbaum.

[14] Ying-li Tian, Takeo Kanade, and Jeffrey F. Cohn. Recognizing action units for facial expression analysis. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 23, pages 97–115, 2001.

[15] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+) : A complete dataset

for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 94–101. IEEE, 2010.

[16] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *CVPR*, 2017.

[17] Savina Jassica Colaco and Dong Seog Han. Deep learning-based facial landmarks localization using compound scaling. *IEEE Access*, 10 :7653–7663, 2022.

[18] Libing Zeng, Lele Chen, Wentao Bao, Zhong Li, Yi Xu, Junsong Yuan, and Nima Khademi Kalantari. 3d-aware facial landmark detection via multi-view consistent training on synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12747–12758, June 2023.

[19] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem ? (and a dataset of 230,000 3d facial landmarks). In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1021–1030, 2017.

[20] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multi-task cascaded convolutional networks. In *IEEE Signal Processing Letters*, volume 23, pages 1499–1503, 2016.

[21] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Incremental face alignment in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1859–1866, 2014.

[22] Michel F Valstar and Maja Pantic. Facs action unit detection using sparse and ensemble methods. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1043–1049, 2010.

[23] Mingyang Liu, Shangfei Wang, Shiguang Shan, and Xilin Chen. Deep learning

facial action unit occurrence detectors. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–18, 2015.

[24] Kaili Zhao, Wen-Sheng Chu, Fernando De la Torre, Jeffrey F Cohn, and Honggang Zhang. Peak-piloted deep network for facial expression recognition. In *Proceedings of the European conference on computer vision*, pages 425–442, 2016.

[25] Linqin Cai, Hongbo Xu, Yang Yang, and Jimin Yu. Robust facial expression recognition using RGB-D images and multichannel features. *Multimedia Tools and Applications*, 78(20) :28591–28607, 2019.

[26] Sandeep Kumar, Shilpa Rani, Arpit Jain, Chaman Verma, Maria Simona Raboaca, Zoltán Illés, and Bogdan Constantin Neagu. Face spoofing, age, gender and facial expression recognition using advance neural network architecture-based biometric system. *Sensors*, 22(14), 2022.

[27] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2704–2713. IEEE, 2018.

[28] Darshan Gera and S Balasubramanian. Landmark guidance independent spatio-channel attention and complementary context information based facial expression recognition. *Pattern Recognition Letters*, 145 :58–66, 2021-05.

[29] Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in Representation Learning : A report on three machine learning contests, 2013-07-01.

[30] Kapil Sethi and Varun Jaiswal. PSU-CNN : Prediction of student understanding in the classroom through student facial images using convolutional neural network. *Materials Today : Proceedings*, 2022.

[31] Neal M. Ashkanasy, Charmine E.J. Härtel, and Catherine S. Daus. Diversity and emotion : The new frontiers in organizational behavior research. *Journal of Management*, 28(3) :307–338, 2002.

[32] Mengyi Liu, Shiguang Shan, Ruiping Wang, and Xilin Chen. Learning Expressionlets on Spatio-temporal Manifold for Dynamic Facial Expression Recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1749–1756. IEEE, 2014.

[33] Mouath Aouayeb, Wassim Hamidouche, Catherine Soladie, Kidiyo Kpalma, and Renaud Seguier. Learning Vision Transformer with Squeeze and Excitation for Facial Expression Recognition, 2021-07-16.

[34] Xing Zhang, Lijun Yin, Jeffrey Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey Girard. BP4D-Spontaneous : A high-resolution spontaneous 3D dynamic facial expression database. *Image and Vision Computing*, 32 :692–706, 2014-06-01.

[35] Seyed Mohammad Mavadati, Mohammad H Mahoor, Marian S Bartlett, and Philip Trinh. Disfa : A spontaneous facial action intensity database. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 397–403. IEEE, 2013.

[36] Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in representation learning : A report

on three machine learning contests. In Minho Lee, Akira Hirose, Zeng-Guang Hou, and Rhee Man Kil, editors, *Neural Information Processing*, pages 117–124. Springer Berlin Heidelberg, 2013.

[37] Ali Mollahosseini, Behzad Hasani, and Mohammad H Mahoor. Affectnet : A database for facial expression, valence, and arousal computing in the wild. In *2017 IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 59–66. IEEE, 2017.

[38] Christopher White, Tao Wu, Shishir Shah, Shwetak Patel, and Dinuka Gunawardena. Emoreact : a multimodal approach and dataset for recognizing emotional responses in children. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pages 457–461. ACM, 2017.

[39] Miao Zhang, Kun Huang, Tao Yu, and Yuan Liu. Affectnet : A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1) :18–31, 2018.

[40] Shan Li, Weihong Deng, and JunPing Du. Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2584–2593. IEEE, 2017.

[41] Michael Lyons, Shigeru Akamatsu, Miyuki Kamachi, and Jiro Gyoba. The japanese female facial expression (jaffe) database. *Proceedings of the 3rd international conference on face and gesture recognition*, pages 14–16, 1999.

[42] Gary McKeown, Michel Valstar, Roddy Cowie, and Maja Pantic. The semaine database : Annotated multimodal records of emotionally colored conversations between a person and a limited agent. In *Affective Computing and Intelligent Interaction*, pages 413–424. Springer, 2012.

[43] M. B. Lopez and P. Johnson. Understanding bias in datasets for facial expression recognition. *Journal of Machine Learning Research*, 21(4) :1–18, 2020.

[44] Gan Zhao and Matti Pietikäinen. A review of datasets and benchmarks for facial expression recognition. *Computer Vision and Image Understanding*, 171 :97–112, 2018.

[45] Silvio Barra, Sanoar Hossain, Chiara Pero, and Saiyed Umer. A Facial Expression Recognition Approach for Social IoT Frameworks. *Big Data Research*, 30 :100353, 2022-11.

[46] Yi-Qing Wang. An Analysis of the Viola-Jones Face Detection Algorithm. *Image Processing On Line*, 4 :128–148, 2014-06-26.

[47] Yufei Wang, Renjie Wan, Wenhan Yang, Haoliang Li, Lap-Pui Chau, and Alex Kot. Low-Light Image Enhancement with Normalizing Flow. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3) :2604–2612, 2022-06-28.

[48] Yinglin Zheng, Hao Yang, Ting Zhang, Jianmin Bao, Dongdong Chen, Yangyu Huang, Lu Yuan, Dong Chen, Ming Zeng, and Fang Wen. General Facial Representation Learning in a Visual-Linguistic Manner. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18676–18688. IEEE, 2022-06.

[49] Asad Malik, Minoru Kuribayashi, Sani M. Abdullahi, and Ahmad Neyaz Khan. DeepFake Detection for Human Face Images and Videos : A Survey. *IEEE Access*, 10 :18757–18775, 2022.

[50] Kaiwen Jiang, Shu-Yu Chen, Feng-Lin Liu, Hongbo Fu, and Lin Gao. NeRFFaceEditing : Disentangled Face Editing in Neural Radiance Fields. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9. ACM, 2022-11-29.

[51] Meng Liu, Yifan Deng, Yuan Liao, Yaguang Li, and Xuemiao Huang. A survey of

deep neural network architectures and their applications. *Neurocomputing*, 383 :43–67, 2020.

[52] Rafael A. Calvo and Sidney D'Mello. Affect detection : An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, 1(1) :18–37, 2010.

[53] Zhihong Zeng, Maja Pantic, Glenn I. Roisman, and Thomas S. Huang. A survey of affect recognition methods : Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1) :39–58, 2009.

[54] Nannan Wang, Xinbo Gao, Dacheng Tao, and Xuelong Li. Facial Feature Point Detection : A Comprehensive Survey, 2014.

[55] Tongtong Cao and Ming Li. Facial expression recognition algorithm based on the combination of CNN and K-Means. In *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*, pages 400–404, 2019.

[56] Afeefa Muhammed, Ramsi Mol, L Revathy Vijay, SS Ajith Muhammed, and AR Shamna Mol. Facial expression recognition using support vector machine (SVM) and convolutional neural network (CNN). *International Journal of Research in Engineering, Science and Management*, 3(8) :574–577, 2020.

[57] Fengping An and Zhiwen Liu. Facial expression recognition algorithm based on parameter adaptive initialization of CNN and LSTM. *The visual computer*, 36(3) :483–498, 2020.

[58] Stefanos Eleftheriadis, Ognjen Rudovic, and Maja Pantic. Multi-conditional Latent Variable Model for Joint Facial Action Unit Detection. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3792–3800, 2015-12.

[59] Wei Zhang, Zhiqiang Bai, and Yuesheng Zhu. An Improved Approach Based on CNN-RNNs for Mathematical Expression Recognition. In *Proceedings of the 2019*

4th International Conference on Multimedia Systems and Signal Processing, pages 57–61. ACM, 2019.

[60] Dong Zhang and Qichuan Tian. A novel fuzzy optimized CNN-RNN method for facial expression recognition. *Elektronika ir Elektrotechnika*, 27(5) :67–74, 2021.

[61] K. Chengeta and S. Viriri. A survey on facial recognition based on local directional and local binary patterns. *Proceedings of the 2018 Conference on Information Communications Technology and Society (ICTAS)*, pages 1–6, 2018.

[62] Md. Zia Uddin, Mohammed Mehedi Hassan, Ahmad Almogren, Mansour Zuair, Giancarlo Fortino, and Jim Torresen. A facial expression recognition system using robust face features from depth videos and deep learning. *Computers & Electrical Engineering*, 63 :114–125, 2017.

[63] Asim Munir, Ayyaz Hussain, Sajid Ali Khan, Muhammad Nadeem, and Sadia Arshid. Illumination invariant facial expression recognition using selected merged binary patterns for real world images. *Optik*, 158 :1016–1025, 2018.

[64] Shan Li, Weihong Deng, and JunPing Du. Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2584–2593. IEEE, 2017.

[65] Jennifer Park. The role of vgg-face and resnet in fer. *Journal of Visual Computing*, 25(3) :80–95, 2020.

[66] M. Deramgozin, S. Jovanovic, H. Rabah, and N. Ramzan. A Hybrid Explainable AI Framework Applied to Global and Local Facial Expression Recognition. In *2021 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–5, 2021-08.

[67] Gholamreza Karimi and Mehdi Heidarian. Facial expression recognition with poly-

nomial Legendre and partial connection MLP. *Neurocomputing*, 434 :33–44, 2021-04-28.

[68] Peixiang Zhang, Ying Liu, Yu Hao, and Jiming Liu. Deep Facial Expression Recognition Algorithm Combining Channel Attention. In *2021 4th International Conference on Artificial Intelligence and Pattern Recognition*, pages 260–265. ACM, 2021-09-24.

[69] Kai Wang, Xiaojiang Peng, Jianfei Yang, Debin Meng, and Yu Qiao. Region Attention Networks for Pose and Occlusion Robust Facial Expression Recognition. nothing, 2019-09-04.

[70] S. Naik and R.P.K. Jagannath. Gcv-based regularized extreme learning machine for facial expression recognition. In *Advances in Machine Learning and Data Science*, pages 129–138, Singapore, 2018. Springer.

[71] Mengyi Liu, Ruiping Wang, Shaoxin Li, Shiguang Shan, Zhiwu Huang, and Xilin Chen. Combining Multiple Kernel Methods on Riemannian Manifold for Emotion Recognition in the Wild. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 494–501. ACM, 2014.

[72] Facial Expression Recognition From Image Sequence Based on LBP and Taylor Expansion | IEEE Journals & Magazine | IEEE Xplore.

[73] Gil Levi and Tal Hassner. Emotion Recognition in the Wild via Convolutional Neural Networks and Mapped Binary Patterns. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 503–510. ACM, 2015.

[74] Otkrist Gupta, Dan Raviv, and Ramesh Raskar. Illumination invariants in deep video expression recognition. *Pattern Recognition*, 76 :25–35, 2018.

[75] Hitesh Kumar Sharma, Tanupriya Choudhury, Adarsh Kandwal, Anurag Mor, Preeti Sharma, Md Ahmed, Prashant Ahlawat, et al. CNN based facial expression recognition system using deep learning approach. In *Cyber Intelligence and Information Retrieval*, pages 391–405. Springer, 2022.

[76] Praneeth Purini and Rahul Kumar Chaurasiya. Real-time facial expression recognition using CNN. In *Applications of Machine Intelligence in Engineering*, pages 425–435. CRC Press, 2022.

[77] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[78] Yinghui Kong, Shuaitong Zhang, Ke Zhang, Qiang Ni, and Jungong Han. Real-time facial expression recognition based on iterative transfer learning and efficient attention network. *IET Image Processing*, 16(6) :1694–1708, 2022.

[79] Cheng Luo, Siyang Song, Weicheng Xie, Linlin Shen, and Hatice Gunes. Learning Multi-dimensional Edge Feature-based AU Relation Graph for Facial Action Unit Recognition. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 1239–1246, 2022.

[80] Prerana Kundu, Pabitra Kundu, Sohini Mallik, Srimoyee Bhowmick, Pratim Mandal, Hritam Banerjee, and Sudipta Basu Pal. Facial expression recognition using convoluted neural network (CNN). In *Cyber Intelligence and Information Retrieval*, pages 81–88. Springer, 2022.

[81] Simge Akay and Nafiz Arica. Stacking multiple cues for facial action unit detection. *The Visual Computer*, 2021-09-21.

[82] Lirong Zhang, Chao Xu, and Shao Li. Facial expression recognition of infants based on multi-stream CNN fusion network. In *2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP)*, pages 37–41. IEEE, 2020.

[83] Yi Wang and Guan Luo. Facial expression recognition with spatial attention mechanism. *IEEE Transactions on Image Processing*, 2018.

[84] Xuan Liu and Heng Zhao. Multi-modal fusion for facial expression recognition. In *Proceedings of the International Conference on Pattern Recognition*, 2020.

[85] Alagesan Bhuvaneswari Ahadit and Ravi Kumar Jatoth. A novel dual CNN ar-

chitecture with LogicMax for facial expression recognition. *Journal of Information Science & Engineering*, 37(1), 2021.

[86] Xinting Yuan and Xiaodi Hu. Facial emotion recognition using convolutional neural networks : State of the art. *Computer Vision and Image Understanding*, 198 :103002, 2019.

[87] Sumit Ghosh, Eriq Muhammad Laksana, Yogesh Venkatesh, and Abhinav Sethi. Facial emotion recognition using convolutional neural networks and transfer learning. *Pattern Recognition Letters*, 125 :90–95, 2018.

[88] Stefanos Zafeiriou, Dimitrios Kollias, Mihalis A Nicolaou, Athanasios Papaioannou, Guoying Zhao, and Irene Kotsia. Facial affect in the wild. *IEEE Transactions on Image Processing*, 2017.

[89] Nguyen Khac Toan, Le Duc Thuan, Le Bao Long, and Nguyen Truong Thinh. Development of Humanoid Robot Head Based on FACS. *IJMERR*, pages 365–372, 2022.

[90] Li Yao, Yan Wan, Hongjie Ni, and Bugao Xu. Action unit classification for facial expression recognition using active learning and SVM. *Multimedia Tools and Applications*, 2021-04-04.

[91] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets : Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv :1704.04861*, 2017.

[92] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks ? *Advances in neural information processing systems*, pages 3320–3328, 2014.

[93] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N

Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[94] Laha Ale, Xiaojie Fang, Dajiang Chen, Ye Wang, and Ning Zhang. Lightweight Deep Learning Model For Facial Expression Recognition. In *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 707–712, 2019-08.

[95] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition, 2015-12-10.

[96] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, 2015-04-10.

[97] Zhiwen Shao, Zhilei Liu, Jianfei Cai, Yunsheng Wu, and Lizhuang Ma. Facial Action Unit Detection Using Attention and Relation Learning. *IEEE Transactions on Affective Computing*, pages 1–1, 2019.

[98] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep High-Resolution Representation Learning for Visual Recognition, 2020.

[99] Zhiwen Shao, Lixin Zou, Jianfei Cai, Yunsheng Wu, and Lizhuang Ma. Spatio-Temporal Relation and Attention Learning for Facial Action Unit Detection. nothing, 2020-01-05.

[100] Haoze Wang. An Expression Recognition Method based on Improved Convolutional Network. In *2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pages 598–602, 2022-06.

[101] Jing Chen, Chenhui Wang, Kejun Wang, and Meichen Liu. Lightweight network architecture using difference saliency maps for facial action unit detection. *Applied Intelligence*, 52(6) :6354–6375, 2022-04-01.

[102] Yahui Nan, Jianguo Ju, Qingyi Hua, Haoming Zhang, and Bo Wang. A-MobileNet : An approach of facial expression recognition. *Alexandria Engineering Journal*, 61(6) :4435–4444, 2022-06.

[103] Shou-Chuan Lai, Ching-Yi Chen, Jian-Hong Li, Fu-Chien Chiu, Ching-Yi Chen, Jian-Hong Li, and Fu-Chien Chiu. Efficient Recognition of Facial Expression with Lightweight Octave Convolutional Neural Network. *Journal of Imaging Science and Technology*, 66 :1–9, 2022-07-01.

[104] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets : Efficient Convolutional Neural Networks for Mobile Vision Applications, 2017.

[105] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. ShuffleNet : An Extremely Efficient Convolutional Neural Network for Mobile Devices, 2017.

[106] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. SqueezeNet : AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size, 2016.

[107] Si Miao, Haoyu Xu, Zhenqi Han, and Yongxin Zhu. Recognizing Facial Expressions Using a Shallow Convolutional Neural Network. *IEEE Access*, 7 :78000–78011, 2019.

[108] Cunjiang Yu, Xiaoying Ding, Longbiao Yang, Guobao Zhou, and Chengwei Yan. Study on mini-Xception- based improved lightweight expression detection model. In *Proceedings of the 2022 2nd International Conference on Control and Intelligent Robotics*, pages 207–215. ACM, 2022.

[109] Mohd Nadhir Ab Wahab, Amril Nazir, Anthony Tan Zhen Ren, Mohd Halim Mohd Noor, Muhammad Firdaus Akbar, and Ahmad Sufril Azlan Mohamed. Efficientnet-lite and hybrid CNN-KNN implementation for facial expression recognition on raspberry pi. *IEEE access : practical innovations, open solutions*, 9 :134065–134080, 2021.

[110] Jia Le Ngwe, Kian Ming Lim, Chin Poo Lee, and Thian Song Ong. Patt-lite : Lightweight patch and attention mobilenet for challenging facial expression recognition, 2023.

[111] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alicia Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai) : Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58 :82–115, 2020.

[112] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, 2017.

[113] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam : Visual explanations from deep networks via gradient-based localization. *Computer Vision and Pattern Recognition*, pages 618–626, 2017.

[114] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you ? : Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[115] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5) :206–215, 2019.

[116] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors : High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[117] Yu Zhang and Bao-Cai Yin. Interpretable facial expression recognition with a region-

based deep learning model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8279–8286, 2019.

[118] Xingchen Liu, Zhiheng Wang, Lin Yang, Yi Li, and Weiming Li. Improving the interpretability of facial expression recognition with attention mechanisms. *International Journal of Computer Vision*, 128(2) :394–412, 2020.

[119] Jing Han, Ke Wang, Kang Zheng, Hanpeng Li, Menglei Jia, and Yuchun Li. Facial action unit detection using an attention-based convolutional neural network. *Frontiers in Psychology*, 12 :653032, 2021.

[120] Siyue Xie, Haifeng Hu, and Yongbo Wu. Deep multi-path convolutional neural network joint with salient region attention for facial expression recognition. *Pattern Recognition*, 92 :177–191, 2019-08.

[121] Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, and Zhengyou Zhang. Training deep networks for facial expression recognition with crowd-sourced label distribution. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ICMI '16, pages 279–283. Association for Computing Machinery, 2016-10-31.

[122] Alexandre Bailly, Corentin Blanc, Elie Francis, Thierry Guillotin, Fadi Jamal, Béchara Wakim, and Pascal Roy. Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. *Computer Methods and Programs in Biomedicine*, 213 :106504, 2022.

[123] Diederik P Kingma and Jimmy Ba. Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*, 2014.

[124] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[125] Pieter-Tjerk De Boer. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1) :19–67, 2005.

[126] Xiao Sun, Shixin Zheng, and Hongshuai Fu. Pandit2020. *IEEE access : practical innovations, open solutions*, 8 :7183–7194, 2020.

[127] Jiahui She, Yibo Hu, Hailin Shi, Jun Wang, Qiu Shen, and Tao Mei. Dive into ambiguity : Latent distribution mining and pairwise uncertainty estimation for facial expression recognition. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6244–6253, 2021.

[128] Tong Zhang, Wenming Zheng, Zhen Cui, Yuan Zong, and Yang Li. Spatial–Temporal Recurrent Neural Network for Emotion Recognition. *IEEE Transactions on Cybernetics*, 49(3) :839–847, 2019-03.

[129] Elfatih Elmubarak Mustafa and Gafar Zen Alabdeen Salh. FACIAL EMOTION RECOGNITION BASED ON DEEP LEARNING TECHNIQUE. . *Vol.*, page 12, 2021.

[130] A. R. Shahid, Sheheryar Khan, and Hong Yan. Contour and region harmonic features for sub-local facial expression recognition. *J. Vis. Commun. Image Represent.*, 2020. 0 citations (Crossref) [2021-06-29] ECC : 0000001.

[131] Chang Liu, Kaoru Hirota, Junjie Ma, Zhiyang Jia, and Yaping Dai. Facial Expression Recognition Using Hybrid Features of Pixel and Geometry. *IEEE Access*, 9 :18876–18889, 2021.

[132] Muhammad Naveed Riaz, Yao Shen, Muhammad Sohail, and Minyi Guo. eXnet : An Efficient Approach for Emotion Recognition in the Wild. *Sensors*, 20(4) :1087, January 2020. ECC : 0000005 6 citations (Crossref) [2021-06-29] Number : 4 Publisher : Multidisciplinary Digital Publishing Institute.

[133] Muhamad Dwisnanto Putro, Duy-Linh Nguyen, Adri Priadana, and Kang-Hyun Jo. An Efficient Multi-view Facial Expression Classifier Implementing on Edge Device. In *An Efficient Multi-view Facial Expression Classifier Implementing on Edge Device*, pages 517–529. SpringerLink, 2022.

[134] Yahui Nan, Jianguo Ju, Qingyi Hua, Haoming Zhang, and Bo Wang. A-MobileNet : An approach of facial expression recognition. *Alexandria Engineering Journal*, 61(6) :4435–4444, 2022.

[135] Di Chang, Yufeng Yin, Zongjian Li, Minh Tran, and Mohammad Soleymani. LibreFace : An Open-Source Toolkit for Deep Facial Expression Analysis, 2023.

[136] Darshan Gera, Naveen Siva Kumar Badveeti, Bobbili Veerendra Raj Kumar, and S. Balasubramanian. Dynamic Adaptive Threshold based Learning for Noisy Annotations Robust Facial Expression Recognition, 2022.

[137] Jiaqi Zhu, Shuaishi Liu, Siyang Yu, and Yihu Song. An Extra-Contrast Affinity Network for Facial Expression Recognition in the Wild. *Electronics*, 11(15) :2288, 2022.

[138] Xiaoyu Tang, Sirui Liu, Qiuchi Xiang, Jintao Cheng, Huifang He, and Bohuan Xue. Facial Expression Recognition Based on Dual-Channel Fusion with Edge Features. *Symmetry*, 14(12) :2651, 2022.

[139] Darshan Gera, Badveeti Naveen Siva Kumar, Bobbili Veerendra Raj Kumar, and S. Balasubramanian. Class adaptive threshold and negative class guided noisy annotation robust Facial Expression Recognition, 2023.

[140] Hongqi Feng, Weikai Huang, Denghui Zhang, and Bangze Zhang. Fine-Tuning Swin Transformer and Multiple Weights Optimality-Seeking for Facial Expression Recognition. *IEEE Access*, 11 :9995–10003, 2023.

[141] Sunyoung Cho and Jwajin Lee. Learning Local Attention With Guidance Map for Pose Robust Facial Expression Recognition. *IEEE Access*, 10 :85929–85940, 2022.

[142] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You ?" : Explaining the Predictions of Any Classifier. 'nothing', 2016-02-16.

[143] Serg Masís. *Interpretable Machine Learning with Python*. Packt Publishing, 2021.

[144] Zheng Zhang, Taoyue Wang, and Lijun Yin. Region of Interest Based Graph Convolution : A Heatmap Regression Approach for Action Unit Detection. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2890–2898. ACM, 2020-10-12.

[145] Yingjie Chen, Han Wu, Tao Wang, Yizhou Wang, and Yun Liang. Cross-Modal Representation Learning for Lightweight and Accurate Facial Action Unit Detection. *IEEE Robotics and Automation Letters*, 6(4) :7619–7626, 2021-10.

[146] Xuri Ge, Pengcheng Wan, Hu Han, Joemon M. Jose, Zhilong Ji, Zhongqin Wu, and Xiao Liu. Local Global Relational Network for Facial Action Units Recognition. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pages 01–08. IEEE, 2021-12-15.

[147] Nishant Sankaran, Deen Dayal Mohan, Nagashri N. Lakshminarayana, Srirangaraj Setlur, and Venu Govindaraju. Domain adaptive representation learning for facial action unit recognition. *Pattern Recognition*, 102 :107127, 2020-06.

[148] Cheng-Hao Tu, Chih-Yuan Yang, and Jane Yung-jen Hsu. IdenNet : Identity-Aware Facial Action Unit Detection. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8. IEEE, 2019-05.

[149] Bing-Fei Wu, Yin-Tse Wei, Bing-Jhang Wu, and Chun-Hsien Lin. Contrastive Feature Learning and Class-Weighted Loss for Facial Action Unit Detection. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 2478–2483, 2019-10.

[150] Enrique Sanchez-Lozano, Georgios Tzimiropoulos, and Michel Valstar. Joint Action Unit localisation and intensity estimation through heatmap regression, 2018-07-20.

[151] Kaili Zhao, Wen-Sheng Chu, Fernando De la Torre, Jeffrey F. Cohn, and Honggang Zhang. Joint patch and multi-label learning for facial action unit detection. In *2015*

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2207–2216, 2015-06.

[152] Yurou Wang, Yanqing Chen, Xuelong Li, Jia Liu, and Hongjun Zhang. Attention-based multi-modal deep learning framework for facial expression recognition. *Multimedia Tools and Applications*, 2020.

[153] Jun Fu, Jing Liu, Yuhang Wang, Yong Liang, and Shuigeng Zhou. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[154] Huajun Chen, Yalan Zhao, Jian Li, and Bing Zeng. Attention-based facial action unit detection in low-resolution images and noisy images. *IEEE Access*, 8 :132568–132577, 2020.

[155] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv :1409.0473*, 2014.

[156] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM : Convolutional Block Attention Module, 2018-07-18.

[157] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25 :1097–1105, 2012.

[158] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[159] Song Han, Huizi Mao, and William J Dally. Learning both weights and connections for efficient neural networks. In *Advances in Neural Information Processing Systems*, pages 1135–1143, 2015.

[160] Song Han, Jeff Pool, John Tran, and William J Dally. Deep compression : Compres-

sing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations*, 2015.

[161] Bing Li, Xuefang Xu, Qiang Chen, and Zhengjia He. Lightweight convolutional neural network and its application in rolling bearing fault diagnosis under variable working conditions. *Mechanical Systems and Signal Processing*, 138 :106587, 2020.

[162] Abdulrahman H Abdulnabi, Mohammed EH Al-Mualla, and Ghassan Al-Regib. Real-time facial expression recognition on mobile devices. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2018.

[163] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv :1710.09282*, 2017.

[164] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International conference on machine learning*, pages 1737–1746. PMLR, 2015.

[165] James Ren Lee, Linda Wang, and Alexander Wong. EmotionNet Nano : An Efficient Deep Convolutional Neural Network Design for Real-Time Facial Expression Recognition. *Frontiers in Artificial Intelligence*, 3, 2021.

[166] Lutfiah Zahara, Purnawarman Musa, Eri Prasetyo, Irwan Karim, and Saiful Musa. *The Facial Emotion Recognition (FER-2013) Dataset for Prediction System of Micro-Expressions Face Using the Convolutional Neural Network (CNN) Algorithm Based Raspberry Pi.* IEEE, 2020-11-03.

[167] Pham The Vinh and Truong Quang Vinh. Facial Expression Recognition System on SoC FPGA. In *2019 International Symposium on Electrical and Electronics Engineering (ISEE)*, pages 1–4, 2019-10.

[168] Saeed Turabzadeh, Hongying Meng, Rafiq M. Swash, Matus Pleva, and Jozef Juhar.

Facial Expression Emotion Detection for Real-Time Embedded Systems. *Technologies*, 6(1) :17, 2018-03.

[169] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[170] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018.

[171] Mohammad Mahdi Deramgozin, Slavisa Jovanovic, Miguel Arevalillo-HerrÁez, and Hassan Rabah. An Explainable and Reliable Facial Expression Recognition System for Remote Health Monitoring. In *2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 1–4, 2022-10.

[172] Xiaoliang Zhu, Shihao Ye, Liang Zhao, and Zhicheng Dai. Hybrid Attention Cascade Network for Facial Expression Recognition. *Sensors*, 21(6) :2003, 2021-01.

[173] Ce Zheng, Matias Mendieta, and Chen Chen. POSTER : A Pyramid Cross-Fusion Transformer Network for Facial Expression Recognition, 2022.

[174] Lifang Zhou, Siqin Li, Yi Wang, and Junlin Liu. SDNET : Lightweight Facial Expression Recognition For Sample Disequilibrium. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2415–2419, 2022-05.

[175] Rajesh Singh, Sumeet Saurav, Tarun Kumar, Ravi Saini, Anil Vohra, and Sanjay Singh. Facial expression recognition in videos using hybrid CNN & ConvLSTM. *Int. j. inf. tecnol.*, 15(4) :1819–1830, 2023.

[176] Jun Liao, Yuanchang Lin, Tengyun Ma, Songxiying He, Xiaofang Liu, and Guotian He. Facial Expression Recognition Methods in the Wild Based on Fusion Feature of Attention Mechanism and LBP. *Sensors*, 23(9) :4204, 2023.

[177] Yahui Nan, Jianguo Ju, Qingyi Hua, Haoming Zhang, and Bo Wang. A-MobileNet : An approach of facial expression recognition. *Alexandria Engineering Journal*, 61(6) :4435–4444, 2022.

[178] Darshan Gera, S. Balasubramanian, and Anwesh Jami. CERN : Compact facial expression recognition net. *Pattern Recognition Letters*, 155 :9–18, 2022.

[179] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998.

[180] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*, 2014.

[181] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561, 2016.

[182] Augyeris Lioga Seandrio, Awang Hendrianto Pratomo, and Mangaras Yanu Florestiyanto. Implementation of convolutional neural network (CNN) in facial expression recognition. *Telematika : Jurnal Informatika dan Teknologi Informasi*, 18(2) :211–221, 2021.

[183] Weixuan Zhou, Jun Wu, and Y.Y. Tang. A review of recent advances in deep learning for vision systems. *Neural Computing and Applications*, 32(9) :4567–4584, 2020.

[184] Sergey Ioffe and Christian Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456. PMLR, 2015.

[185] Yuxin Wu and Kaiming He. Group normalization. *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[186] Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. Activation Functions in Deep Learning : A Comprehensive Survey and Benchmark, 2021.

[187] Abien Fred Agarap. Deep Learning using Rectified Linear Units (ReLU), 2019.

[188] S. Marra, M.A. Iachino, and F.C. Morabito. Tanh-like Activation Function Implementation for High-performance Digital Neural Systems. In *2006 Ph.D. Research in Microelectronics and Electronics*, pages 237–240, 2006-06.

[189] Binghui Chen, Weihong Deng, and Junping Du. Noisy Softmax : Improving the Generalization Ability of DCNN via Postponing the Early Softmax Saturation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4021–4030, 2017-07.

[190] Xin Zhao, Xiaohui Liang, Liangyue Liu, Teng Li, Fei Han, Nuno Vasconcelos, and Shuicheng Yan. Facial expression recognition : A survey. *arXiv preprint arXiv :1612.02903*, 2016.

[191] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8) :861–874, 2006.

[192] Stephen V Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, 62(1) :77–89, 1997.

[193] Ping Liu, Jiajia Zhou, Ivor W. Tsang, Zhao Meng, and Yew-Soon Han. Facial expression recognition via learning deep sparse autoencoders. *Neurocomputing*, 2013.

[194] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[195] Léon Bottou, Yann LeCun, and Others. Optimization for deep learning highlights in 2017. In *Theoretical Foundations and Applications of Deep Learning*, pages 3–30. NOW, 2018.

[196] Ilya Sutskever, James Martens, George E Dahl, and Geoffrey E Hinton. On the importance of initialization and momentum in deep learning. *ICML (3)*, 28 :1139–1147, 2013.

[197] Diederik P Kingma and Jimmy Ba. Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*, 2014.

[198] Chris Tiesman. Rmsprop : Divide the gradient by a running average of its recent magnitude. *Coursera : Neural Networks for Machine Learning*, 2012.

[199] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[200] Lutz Prechelt. Automatic early stopping using cross validation : quantifying the criteria. *Neural Networks*, 11(4) :761–767, 1998.

[201] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. *Neural Networks : Tricks of the Trade*, 7700 :437–478, 2012.

[202] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1) :1929–1958, 2014.

[203] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv :1611.03530*, 2016.

[204] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017.

[205] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[206] Zhengyao Wen, Wenzhong Lin, Tao Wang, and Ge Xu. Distract Your Attention : Multi-head Cross Attention Network for Facial Expression Recognition, 2022-04-04.

[207] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452 :48–62, 2021-09.

[208] Yifei Guo, Jian Huang, Mingfu Xiong, Zhongyuan Wang, Xinrong Hu, Jihong Wang, and Mohammad Hijji. Facial expressions recognition with multi-region divided attention networks for smart education cloud applications. *Neurocomputing*, 493 :119–128, 2022.

[209] Fuyan Ma, Bin Sun, and Shutao Li. Transformer-Augmented Network with Online Label Correction for Facial Expression Recognition. *IEEE Transactions on Affective Computing*, pages 1–13, 2023.

[210] Yinghui Kong, Shuaitong Zhang, Ke Zhang, Qiang Ni, and Jungong Han. Real-time facial expression recognition based on iterative transfer learning and efficient attention network. *IET Image Processing*, 16(6) :1694–1708, 2022.

[211] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452 :48–62, 2021.

[212] Yiding Shen, Rong Jin, and Kai Chen. An entropy-based attention mechanism for scene text recognition. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3926–3931. IEEE, 2018.

[213] Qiaoyu Tan, Jianwei Zhang, Ninghao Liu, Xiao Huang, Hongxia Yang, Jingren Zhou, and Xia Hu. Dynamic memory based attention network for sequential recommendation, 2021.

[214] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Memory networks. *arXiv preprint arXiv :1410.3916*, 2014.

[215] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka

Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626) :471–476, 2016.

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

[217] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell : Neural image caption generation with visual attention. *arXiv preprint arXiv :1502.03044*, 2015.

[218] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab : Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4) :834–848, 2018.

[219] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words : Transformers for image recognition at scale. *arXiv preprint arXiv :2010.11929*, 2020.

[220] Mengyue Li, Sheng Duan, Jiwen Lu, and Jie Zhou. Facial action unit detection using transformer. *arXiv preprint arXiv :2103.15508*, 2021.

[221] Yuke Chen, Jianshu Li, Hao Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 216–226, 2017.

[222] Saumya Jetley, Nicholas A Lord, Namhoon Lee, and Philip HS Torr. Learn to pay attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6116–6125, 2018.

[223] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv :1608.08710*, 2016.

[224] Sharan Narang, Eric Elsen, Greg Diamos, and Shubho Sengupta. Exploring sparsity in recurrent neural networks. In *Proceedings of the International Conference on Learning Representations*, 2017.

[225] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis : Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

[226] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2074–2082, 2016.

[227] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[228] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient transfer learning. In *International Conference on Learning Representations*, 2016.

[229] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning ? *arXiv preprint arXiv :2003.03033*, 2020.

[230] Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. In *arXiv preprint arXiv :1902.09574*, 2019.

[231] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect : Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*, pages 3123–3131, 2015.

[232] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient

inference : A whitepaper. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 0–0, 2018.

Résumé

Le domaine de la Reconnaissance des Émotions Faciales (FER) est d'une importance capitale pour faire progresser les interactions homme-machine et trouve sa place dans de nombreuses applications, notamment dans le domaine de la santé connectée. En utilisant des Réseaux Neuronaux Convolutifs (CNN), cette thèse présente des modèles visant à optimiser la détection et l'interprétation des émotions pour une implémentation dans les systèmes embarqués. Le modèle initial présenté est de faible complexité et économe en ressources lui permettant de rivaliser favorablement avec les solutions de l'état de l'art sur un nombre limité de jeux de données, ce qui en fait une bonne base pour les systèmes à ressources limitées.

Pour identifier et capturer toute la complexité et l'ambiguïté des émotions humaines, ce modèle initial est amélioré en intégrant les unités d'action faciales (AU). Cette approche affine non seulement la détection des émotions mais fournit également une interprétabilité des décisions fournies par le modèle en identifiant des AU spécifiques liées à chaque émotion.

Une amélioration significative est atteinte en introduisant des mécanismes d'attention neuronale —à la fois spatiaux et par canal— au modèle initial. Ainsi, le modèle basé sur ces mécanismes d'attention se focalise uniquement sur les caractéristiques faciales les plus saillantes. Cela permet au modèle CNN de s'adapter bien aux scénarios du monde réel, tels que des expressions faciales partiellement obscurcies ou subtiles.

La thèse aboutit à un modèle CNN optimisé et efficace en termes de calcul et de taille mémoire, le rendant parfaitement adapté pour les environnements à ressources limitées comme les systèmes embarqués. Tout en fournissant une solution robuste pour la FER, des perspectives et voies pour des travaux futurs, tels que des applications en temps réel et des techniques avancées pour l'interprétabilité du modèle, sont également identifiées.

Mots-clés: Reconnaissance de l'expression faciale, détection d'unité d'action faciale, mécanisme d'attention, réseaux de neurones convolutifs, systèmes embarqués.

Abstract

The field of Facial Emotion Recognition (FER) is pivotal in advancing human-machine interactions and finds application in various domains, particularly in the area of connected health. Leveraging Convolutional Neural Networks (CNNs), this thesis presents a progression of models aimed at optimizing emotion detection and interpretation for implementation in embedded systems. The initial model is resource-frugal but competes favorably with state-of-the-art solutions, making it a strong candidate for systems with limited computational and memory resources.

To capture the complexity and ambiguity of human emotions, the research work presented in this thesis enhances this CNN-based foundational model by incorporating facial Action Units (AUs). This approach not only refines emotion detection but also provides interpretability by identifying specific AUs tied to each emotion.

Further enhancement is achieved by introducing neural attention mechanisms—both spatial and channel-based—improving the model's focus on salient facial features. This makes the CNN-based model adapted well to real-world scenarios, such as partially obscured or subtle facial expressions.

Based on the previous results, in this thesis, we propose finally an optimized, yet computationally efficient, CNN model that is ideal for resource-limited environments like embedded systems. While it provides a robust solution for FER, this research also identifies perspectives for future work, such as real-time applications and advanced techniques for model interpretability.

Keywords: Facial Expression Recognition, Facial Action Unit Detection, Attention mechanism, Convolutional Neural Networks, embedded systems.