République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahleb de Blida

Faculté des Sciences

Département d'informatique

Mémoire de fin d'étude

Pour L'Obtention du diplôme de Master en *Informatique*
Option : *Ingénerie de Logiciel*

Du pixel au sentiment : Analyse de sentiments dans les images avec le deep learning

Réalisé par : ZEROUAL Ahmed Zakaria

Mme. Présidente : Mme MADANI Amina

Mme. Examinateur : Mme CHERFA

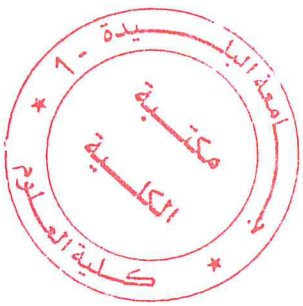
Mme. Promoteur : Mme BOUMAHDI Fatima

Soutenu le : **Juin 2018**

Abstract

The use of machines to perform different tasks is constantly increasing in society. Providing machines with perception can lead them to perform a greatvariety of tasks;even very complex ones such as elderly care. Machine perception requires that machines understand about their environment and interlocutor's intention. Recognizing facial emotions might help in this regard. During the development of thiswork, deep learning techniques have been used over images displaying the following facial emotions : happiness, sadness, anger, surprise, disgust, and fear.

In this research, a pure convolutional neural network approach out performed others tatistical methods' results achieved by other authors that include feature engineering. Utilizing convolutional networks involves feature learning; which sounds very promising for this task where definingn features is not trivial. Moreover, the network was evaluated using two different corpora: one was employed during network's training and it was also helpful for parametertuning and for network's architecture definition. This corpus consisted of facial acted emotions. The network providing best classification accuracy results was tested against the second dataset. Eventhough the network was trained using only one corpus; the network reported auspicious results when tested on a different dataset, which displayed facial non-acted emotions. While the results achieved were not state-of-the-art; the evidence gathered points out deep learning might be suitable to classify facial emotion expressions. Thus, deep learning has the potential to improve human-machine interaction because its ability to learn features will allow machines to develop perception. And by having perception, machines will potentially provide smoother responses, drastically improving the user experience.



Résumé

L'utilisation de machines pour effectuer différentes tâches est en constante augmentation dans la société. Donner de la perception aux machines peut les amener à accomplir une grande variété de tâches, même des tâches très complexes comme les soins aux personnes âgées. La perception des machines exige que les machines comprennent leur environnement et l'intention de l'interlocuteur. Reconnaître les émotions faciales pourrait aider à cet égard. Au cours du développement de ce travail, des techniques d'apprentissage profond ont été utilisées sur des images montrant les émotions faciales suivantes : bonheur, tristesse, colère, surprise, dégoût et peur.

Dans cette recherche, une approche de réseau neuronal convolutionnel pur a surpassé les résultats obtenus par d'autres méthodes statistiques par d'autres auteurs qui incluent l'ingénierie des caractéristiques. L'utilisation de réseaux convolutifs implique l'apprentissage des caractéristiques, ce qui semble très prometteur pour cette tâche où la définition des caractéristiques n'est pas triviale. De plus, le réseau a été évalué à l'aide de deux corpus différents : l'un a été utilisé pendant la formation du réseau et il a également été utile pour le réglage des paramètres et pour la définition de l'architecture du réseau. Ce corpus se composait d'émotions faciales. Le réseau fournissant les meilleurs résultats de précision de classification a été testé par rapport au deuxième ensemble de données. Même si le réseau n'a été formé qu'avec un seul corpus, le réseau a fait état de résultats prometteurs lorsqu'il a été testé sur un autre ensemble de données, qui montrait des émotions faciales non dirigées. Bien que les résultats obtenus n'étaient pas à la fine pointe de la technologie, les preuves recueillies indiquent qu'un apprentissage en profondeur pourrait convenir pour classer les expressions d'émotions faciales. Ainsi, l'apprentissage en profondeur a le potentiel d'améliorer l'interaction homme-machine parce que sa capacité d'apprentissage permettra aux machines de développer la perception. Et en ayant la perception, les machines fourniront potentiellement des réponses plus lisses, ce qui améliorera considérablement l'expérience de l'utilisateur.

ملخص

ن استخدام الآلات لأداء مهام مختلفة يتزايد باستمرار في المجتمعن استخدام الآلات لأداء مهام مختلفة
.يتزايد باستمرار في المجتمع
يمكن أن يؤدي توفير الآلات بالإدراك إلى أداء مجموعة كبيرة ومتنوعة من المهام ؛
حتى تلك معقدة للغاية مثل رعاية المسنين. تصور الجهاز يتطلب ذلك
.الآلات فهم بيئتهم ونوايا المحاور
قد يساعد التعرف على مشاعر الوجه في هذا الصدد. خلال تطوير هذا
العمل ، وقد استخدمت تقنيات التعلم العميق أكثر من الصور التي تظهر ما يلي
.العواطف الوجه: السعادة والحزن والغضب والمفاجأة والاشمئزاز والخوف
في هذا البحث ، تفوق نهج الشبكة العصبية التحويلي النقي على غيرها
.نتائج الأساليب الإحصائية التي حققها المؤلفون الآخرون والتي تشمل الهندسة المميزة
استخدام الشبكات التلافيفية ينطوي على تعلم الميزات ؛ الذي يبدو جدا
واعدا لهذه المهمة حيث تعريف الميزات ليست تافهة. وعلاوة على ذلك ، فإن الشبكةتم تقييمه باستخدام
اثنين من المجلات المختلفة: واحد كان يعمل خلال الشبكةالتدريب وكان من المفيد أيضا لضبط المعلمة
ولهندسة الشبكةفريف. تألفت هذه الجثة من المشاعر تصرف الوجه. شبكة توفير أفضلتم اختبار نتائج دقة
التصنيف في مقابل مجموعة البيانات الثانية. على الرغم من أنتم تدريب الشبكة باستخدام مجموعة واحدة
فقط ؛ سجلت الشبكة نتائج مبشرةعند اختبارها على مجموعة بيانات مختلفة ، والتي أظهرت انفعالات
الوجه غير الفعالة. في حينالنتائج التي تحققت لم تكن على أحدث طراز. الأدلة التي تم جمعها تشير إلى
عمقالتعلم قد يكون مناسبًا لتصنيف تعبيرات انفعال الوجه. وهكذا ، التعلم العميقلديه القدرة على تحسين
التفاعل بين الإنسان والآلة بسبب قدرتها على التعلمسوف تسمح الميزات للآلات لتطوير الإدراك. ومن
خلال التصور ،من المحتمل أن توفر الآلات استجابات أكثر سلاسة ، مما يعمل على تحسين المستخدم
بشكل كبير
..تجربة

Dédicace

Je dédie ce travail à :

- **Mes chères parents qui sont la lumière de ma vie.**
- **Toute ma famille.**
- **Tous mes amis.**
- **Mme BOUMAHDI pour son aide précieuse**

TABLE DES MATIÈRES

TABLE DES FIGURES

CHAPITRE 1

INTRODUCTION

Les gens ont toujours été intéressés par l'opinion de l'autre, non seulement avant l'achat d'un nouveau produit, mais également avant de prendre une décision importante.

Les entreprises s'intéressent à l'analyse des sentiments pour connaître l'opinion du public sur un produit. De plus, des acheteurs potentiels tiennent beaucoup à l'opinion publique avant d'acheter un produit de sorte que des entreprises prêtent énormément d'attention au sentiment des clients. L'imp ortance de l'analyse des sentiments est présente dans plusieurs domaines, à savoir politique, marketing, gestion de la réputation, ...

L'Analyse des Sentiments (Opinion Mining), font partie d'un domaine émergent. Ce dernier s'occupe de traitement de la subjectivité : opinions, avis, sentiments, émotions, évaluations, croyances ou jugements personnel. Ensuite, il attribue une polarité (positive, négative ou neutre) à cette opinion. Ces données d'opinion revêtent aujourd'hui une importance stratégique et économique évidente car leur analyse permet de connaître les points forts et les points faibles des produits, d'estimer la perception du produit par les consommateurs, afin d'améliorer les profits.

Il permet également au consommateur de donner son opinion, de l'aider à la prise de décision, en s'inspirant des sentiments et d'opinions d'autres clients sur un produit donné.
Dans ce projet nous nous intéressons au problème de l'analyse et de la visualisation des sentiments contenus dans les images. Compte tenu du nombre des images sur les réseaux sociaux, il est impossible de manipuler les données manuellement, et des outils de traitement automatique doivent donc être mis en place.

Les objectifs

Les émotions faciales fournissent des informations sur l'état intérieur du sujet. Si une machine est capable d'obtenir une séquence d'images faciales, l'utilisation de techniques d'apprentissage profond aiderait les entreprises à prendre conscience de l'humeur de leur clients. Dans ce contexte, l'apprentissage profond a le potentiel de devenir un facteur clé pour construire une meilleure solution de détection des sentiments.

L'objectif de ce projet est de proposer une nouvelle approche permettant d'analyser efficacement et de comprendre les sentiments des gens sur une image au fil du temps, afin de restituer le maximum d'informations aux utilisateurs. Les objectifs majeurs de ce travail sont les suivants :

1. La proposition et conception d'une architecture à base de deep learning pour extraire les sentiments associés à une image.
2. La proposition d'une nouvelle stratégie pour associer les images aux sentiments correspondants.

La structure du mémoire

Le reste du mémoire est organisé en trois chapitres : nous consacrons un premier chapitre se focalise sur l'état de l'art de Machine Learning, Réseaux neuraux artificiels et l'apprentissage Approfondi, nous présentons notamment les travaux inhérents à l'analyse des sentiments dans les

images. nous consacrons un second chapitre à présenter des généralités sur le domaine d'analyse des sentiments en particulier les expressions faciales.

Notre troisième chapitre illustre la méthode proposée afin détecter les émotions d'une personne à partir de sa photo, en se basant sur la technique Deep learning. Le dernier chapitre présente l'expérimentation et la méthode proposée en considérant les phases d'apprentissage et de test. Nous concluons avec une synthèse de travail et des perspectives.

CHAPITRE 2

CONTEXTE THÉORIQUE

Dans cette section, une description des concepts pertinents pour ce projet est présentée. Cette section a pour but de fournir un contexte sur les sujets qui seront discutés au cours de la suite du rapport. Pour décrire les concepts, une approche descendante sera utilisée. De plus, des recherches connexes à l'approche utilisée dans le cadre de ce projet sont également présentées.

2.1 Machine Learning

Machine Learning (ML) est un sous-domaine de l'intelligence artificielle. Une explication simple de ML est celui présenté par Arthur Samuel en 1959[45] : "..... domaine d'études qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmé".
Cette déclaration fournit un puissant aperçu de l'approche particulière de ce domaine. C'est complètement différent à partir d'autres champs où toute nouvelle caractéristique doit être ajoutée à la main.
Par exemple, dans le développement de logiciels, lorsqu'une nouvelle exigence apparaît, un programmeur doit créer un nouveau logiciel pour traiter cette nouvelle exigence.

Dans le ML, ce n'est pas exactement le cas. Les algorithmes ML créer des modèles, basés sur les données d'entrée. Ces modèles génèrent un résultat qui est généralement un ensemble de prédictions ou de décisions. Puis, lorsqu'une nouvelle exigence apparaît, l'attribut pourrait être en mesure de le gérer ou de fournir une réponse sans qu'il soit nécessaire d'ajouter un modèle nouveau code.

ML est généralement divisé en 3 grandes catégories. Chaque catégorie se concentre sur la façon dont le processus d'apprentissage est exécuté par un système d'apprentissage. Ces catégories sont : **l'apprentissage supervisé**, **l'apprentissage non supervisé** et **l'apprentissage de renforcement**, comme le monter la figure 2.1

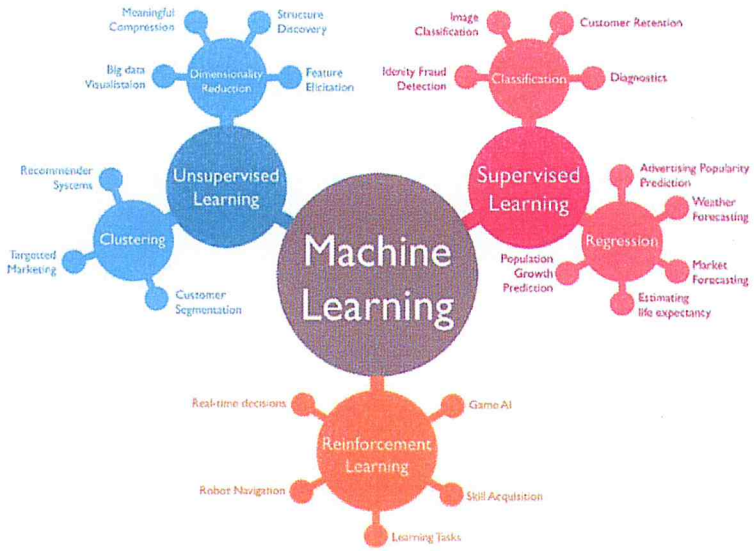


FIGURE 2.1: Les trois cagoteries de ML [65]

L'apprentissage supervisé est quand un modèle reçoit un ensemble d'intrants labellisés, C'est a dire qu'ils contiennent également la classe d'appartenance correspondante. Le modèle essaie de s'adapter d'une manière qui permet de mapper chaque entrée avec la classe de sortie correspondante.

D'autre part, **l'apprentissage non supervisé** reçoit un ensemble d'intrants sans qu'ils ne soient labellisé. En ce sens, le modèle tente d'apprendre à partir des données en explorant des modèles sur eux. Enfin, le renforcement de l'apprentissage, c'est lorsqu'un agent est récompensé ou puni. en conséquence, les décisions qu'il a prises pour atteindre un objectif.

Sur ce projet, notre problème entre dans la catégorie de **l'apprentissage supervisé** puisque les images à traiter sont labellisées. Dans notre cas, le label est l'émotion que l'image représente.

2.2 Réseaux neuraux artificiels

L'apprentissage supervisé dispose d'un ensemble d'outils axés sur la résolution de problèmes dans son domaine.

L'un de ces outils est appelé Réseaux neuronaux artificiels (ANN). Un ANN est un ensemble de fonctions de prédiction de label. Si l'ANN est analysé comme une boîte noire, l'élément se composerait d'exemples étiquetés, et la sortie serait un vecteur contenant un ensemble de prédictions. Habituellement, ces prédictions sont exprimées sous forme de probabilité. pour toutes les labels.
D'autres définitions de l'ANN mettent l'accent sur d'autres aspects. comme ses propriétés de transformation [80] : "Un processeur massivement parallèle et distribué fait d'unités de traitement simples qui ont une propension naturelle au stockage d'expériences. et de le rendre disponible pour utilisation." .

Toutefois, un ANN peut ne pas nécessairement être massive. Les petites implémentations sont faites juste pour essayer de nouveaux idées. Engelbrecht a fourni une définition avec une intention différente, plus focalisée sur la topologie : "C'est un réseau de neurones artificiels. Réseau de neurones artificiels peut se composer d'une couche d'entrée, de couches cachées et d'une couche

de sortie. Un système artificiel est un modèle de neurone biologique." .

Un ANN peut être expliqué à travers les trois étapes suivantes[25] :

— Entrer des données dans le réseau.

— La transformation sur les données d'entrée s'effectue au moyen d'une valeur de la somme pondérée.

— Un état intermédiaire est calculé en appliquant une fonction non linéaire à la transformation précédente.

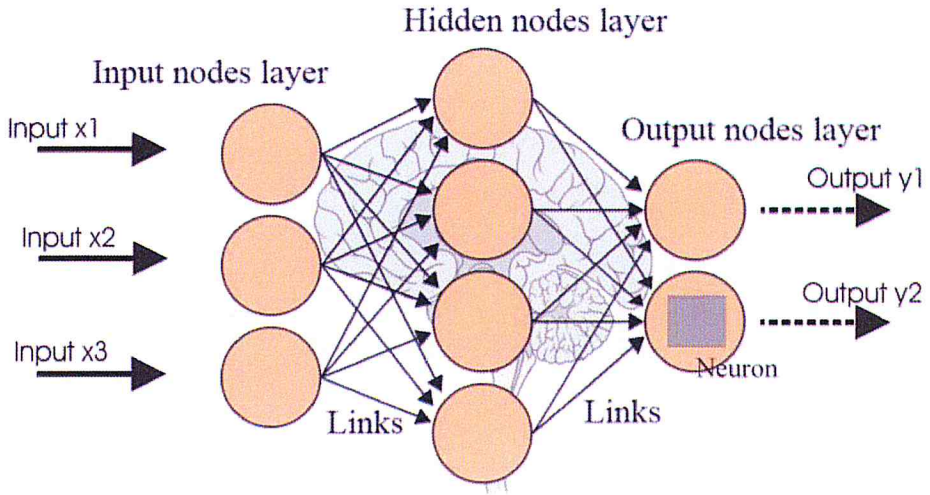


FIGURE 2.2: Topologie des réseaux de neurones artificiels [41] .

Pour en revenir à la définition d'Engelbrecht, une question intéressante se pose : comment un modèle inspiré de la mécanique du cerveau a-t-il fini par devenir un modèle informatique ? Afin d'apporter une réponse, un certain contexte historique est nécessaire.

2.2.1 L'ascension et la chute de l'ANN

ANN date des années 1940[10]. Alors que certains chercheurs ont commencé à étudier le cerveau aucun d'entre eux n'a été en mesure de le formuler comme un dispositif informatique. C'était jusqu'en 1943[53], quand Warren McCulloch et Walter Pitts ont été en mesure de formuler un ANN comme modèle approprié pour effectuer des calculs [61].
Quelques années plus tard (1949)[69], Donald Hebb a fourni une théorie pour décrire comment les neurones s'adaptent sur le cerveau pendant qu'ils s'adaptent, le processus d'apprentissage a lieu Après cela, il a fallu près d'une décennie pour qu'un ANN mise en œuvre : le perceptron. Le perceptron a été introduit par Frank Rosenblatt. [3]. C'est l'architecture ANN la plus simple. D'ailleurs, c'était la première fois qu'en moyen d'apprentissage supervisé, un ANN a été capable d'apprendre.

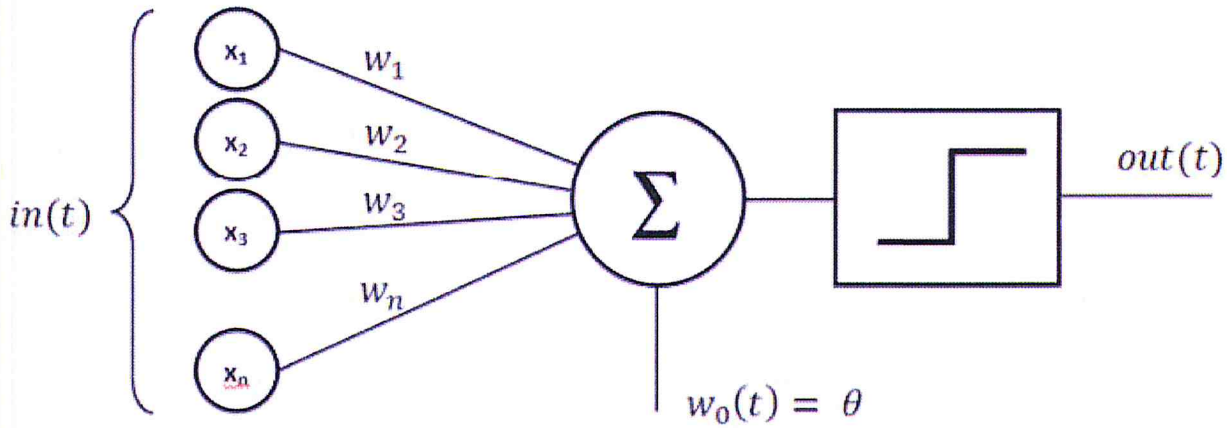


FIGURE 2.3: Topologie du Pércéptron[6]

Dans la figure 2.2 , la topologie d'un perceptron est présentée. Heureusement, la plus grande partie des concepts ANN peuvent être expliqués dans cette architecture simple. Comme on peut le voir, il y a un ensemble d'entrées, **X1** à **Xn**. Cette couche est appelée **couche d'entrée**. Chacun de ces entrées a un poids correspondant, **Wn**.

Sur le neurone (unité), une somme pondérée est effectuée. De plus, un biais est ajouter au neurone afin qu'il puisse mettre en œuvre une fonction linéaire. L'indépendance du parti pris se déplacera la courbe sur l'axe**X**

$$y = \mathrm{f(t)} = \sum_{i1}^{n} Xi * Wi\ \theta$$

Après cela, le résultat de f(t) est l'entrée d'une fonction d'activation. La fonction d'activation définit la sortie du nœud. Comme le perceptron est un classificateur binaire, l'attribut est adaptée à cette topologie. Il n'émettra qu'un couple de classes, 0 ou 1.

$$\text{output} =$$

$$\begin{pmatrix} 0 & si & y & 0 \\ 1 & si & y & 0 \end{pmatrix}$$

Enfin, la prédiction est mesurée par rapport à la valeur réelle. Ce signal d'erreur va être utilisé pour mettre à jour les poids sur la première couche afin d'améliorer les résultats de prédiction. Ceci est réalisé par l'apprentissage de la rétropagation[6].

2.2.2 La renaissance de ANN

Au cours des années 1970 et 1980 [10], le passage à l'IA s'est exclusivement orienté vers le traitement symbolique. Cependant, un ensemble de facteurs a permis à ANN d'être à nouveau sur place au milieu des années 80 Ces facteurs appartiennent à des domaines différents. L'un des facteurs était la lenteur des progrès du traitement symbolique, qui a été rétréci à de petites simulations.
Un autre facteur était l'accessibilité des ordinateurs et du matériel informatique. par rapport à celle des décennies précédentes.

De nos jours, ce facteur est toujours d'actualité puisque les expériences exigent beaucoup de puissance de calcul. La plupart des simulations actuelles ne seraient pas réalisables à cette époque. Enfin, les chercheurs connectionnistes ont commencés à montrer des résultats intéressants. Par exemple, en 1988[72], Terrence. Sejnowski et Charles R. Rosenberg ont publié un article avec

les résultats de NETtalk . NETtalk est un ANN qui a été formé pour apprendre à prononcer des mots anglais. Ainsi, le connectionnisme a commencé à prendre de l'ampleur.

Un an plus tard, Yann LeCun a montré des résultats impressionnants sur la reconnaissance de code postal manuscrit en utilisant un ANN multicouche[22]. Cet article est très intéressant car il a été un pionnier dans la collaboration entre la reconnaissance d'images et l'apprentissage automatique.

De plus, cet article a introduit des concepts liés aux réseaux neuronaux convolutifs tels que les cartes de caractéristiques et le partage des poids[77].

Au cours des années 1990, les machines vectorielles de support (SVM) ont été très utilisées par de nombreuses chercheurs. Sa popularité était due à sa simplicité par rapport à l'ANN, et aussi parce qu'ils obtenaient d'excellents résultats. Aujourd'hui, on ne parle plus d'ANN, mais d'un très populaire terme : L'apprentissage profond.

2.3 Apprentissage Approfondi

La dernière réincarnation de l'ANN est connue sous le nom de Deep Learning (DL). Selon Yann LeCun, ce terme désigne ".... toute méthode d'apprentissage qui peut entraîner un système avec plus de 2 ou 3 couches cachées non linéaires"[22].
DL a obtenu des succès dans des domaines tels que la vision par ordinateur, le traitement du langage naturel et la reconnaissance automatique de la parole. L'une des principales forces de l'utilisation des techniques DL est qu'il n'y a pas besoin d'ingénierie de fonctionnalités. Les algorithmes sont capables d'apprendre les caractéristiques par eux-mêmes sur des représentations de base. Par exemple, sur la reconnaissance d'images, un ANN peut être alimenté avec des représentations d'images en pixels. Ensuite, l'algorithme déterminera si certains représente n'importe quelle caractéristique particulière, qui est répétée par le biais de la combinaison de pixels de l'image. Au fur et à mesure que les données sont traitées à travers les couches, les caractéristiques passeront

d'un niveau des formes abstraites à une représentation significative des objets.

2.3.1 Les applications du DL

DL a commencé à devenir populaire après quelques meilleurs résultats que ceux de l'état de l'art. réalisés sur plusieurs domaines. Par exemple, le premier document contenant des informations sur une application industrielle majeure était une application liée à la reconnaissance automatique de la parole[42]. Dans cet article de 2012, ANN a surpassé les modèles de mélange gaussiens dans plusieurs tests d'étalonnage.
Ce document est le fruit d'une collaboration entre quatre groupes de recherche : Université de Toronto, Microsoft Research, Google Research et IBM Research. Deux ans plus tard, une autre publication en petits groupes portait sur le domaine du traitement du langage naturel[76].

Cette recherche présentait que la mémoire à long terme (une mémoire à court terme particulière ANN appelée réseau neuronal récurrent spécialisé sur les séquences) a permis d'obtenir de meilleurs résultats. que la traduction automatique statistique, qui était l'outil par défaut pour la traduction. à ce moment-là. Ce réseau a été en mesure de traduire des mots et des phrases de l'anglais vers le français.

Enfin, une technique d'apprentissage approfondi pertinente pour ce projet est présentée : Réseaux neuronaux convolutionnels (CNN). Un article publié en 2012 par un groupe de chercheurs financés par des chercheurs de l'Université de Toronto[55] ont montré des résultats jamais atteints auparavant sur le site Web du Concours de classification ImageNet. Cette recherche est devenue un travail fondamental sur DL.

Sur l'édition 2012, sa solution utilisant CNN profond a atteint un taux d'erreur de **15,3** % dans le top 5, tandis que le deuxième meilleur a atteint **26,2** %. Dans cette recherche, deux concepts, largement adoptés par la communauté ML, sont les suivants souligné : l'utilisation

d'une unité linéaire rectifiée en tant que fonction d'activation[33] et l'utilisation de GPU pour l'entraînement[88].

2.3.2 Unité linéaire rectifiée

La fonction d'activation d'une unité (neurone) est une partie essentielle d'une architecture ANN. L'utilisation de différentes fonctions a été utilisée par les chercheurs depuis les débuts de l'ANN. La fonction ReLU a été introduite comme fonction d'activation. Cependant, la nature binaire de la fonction ReLU ne permet pas d'avoir une bonne approximation des erreurs.

Afin de surmonter cette situation, les fonctions sigmoïdes ont été utilisées. Ils ont fourni une très bonne performance pour les petits réseaux. Bien que l'utilisation de la fonction sigmoïde ait prouvé ne pas être évolutif sur les grands réseaux[55]. Le coût computationnel de l'exponentielle L'opération pourrait être très coûteuse puisqu'elle pourrait mener à des nombres très longs. Un autre facteur contre l'utilisation de la fonction sigmoïde est le problème de disparition du gradient. Cela signifie que la valeur du gradient sur les queues de la courbe devient trop petite pour qu'elle empêche l'apprentissage.

Dans ce scénario, la fonction d'unité linéaire rectifiée (ReLU) offrait des avantages par rapport aux fonctions d'activation communes précédentes : son coût de calcul était moins élevé, elle fournissait une bonne approximation des erreurs et elle ne souffrait pas du problème de disparition du gradient. ReLU est affiché sur la Figure 2.4, et il est défini comme suit : f(t) $=max(0,x)$

La recherche de Krizhevsky et al[55] a montré que l'utilisation de ReLU réduisait le nombre d'époques requises pour converger lors de l'utilisation de la descente à gradient stochastique par un facteur 6.

Cependant, un inconvénient majeur lors de l'utilisation de ReLU est sa fragilité lorsque l'intrant est inférieure à zéro. Cela se produit lorsque le neurone atteint un point où il ne sera plus activé par aucun point de données pendant l'entraînement.

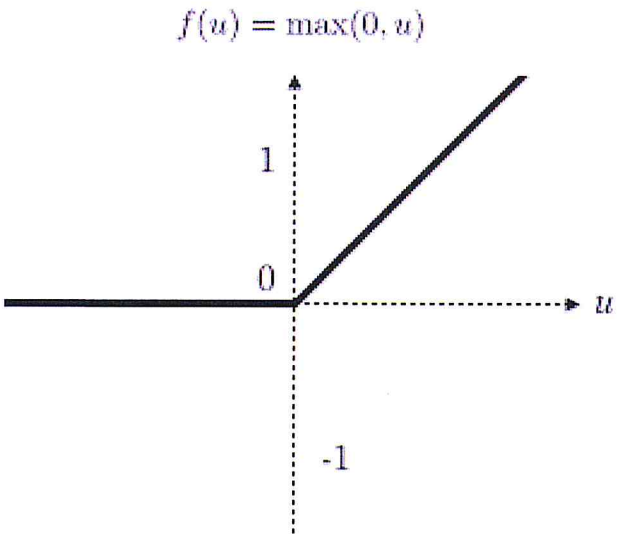


FIGURE 2.4: Rectified Linear Unit (ReLU)[43]

2.3.3 Utilisation du Graphical Prossesing Unite

L'utilisation de GPU pour la formation est devenue fondamentale pour la formation de réseaux profonds pour des raisons pratiques. La raison principale est la réduction du temps de formation par rapport à la formation CPU.
Bien que des accélérations différentes soient rapportées en fonction de la topologie du réseau, il est courant d'avoir une vitesse d'environ 10 fois supérieure lorsque l'on utilise le GPU .

La différence entre le CPU et le GPU réside dans la façon dont ils traitent les tâches. CPU conviennent pour effectuer un traitement séquentiel en série sur quelques cœurs. D'autre part, le GPU englobe une architecture massive parallèle. Cette architecture implique des milliers de de petits noyaux conçus pour traiter simultanément plusieurs tâches.
Ainsi, les opérations DL sont aptes à s'entraîner sur GPU puisqu'elles impliquent des opérations vectorielles et des opérations matricielles qui peuvent être traitées en parallèle. Malgré cela, au cours de ce projet seulement un nombre limité d'expériences ont été menées à l'aide de GPU, il est important de souligner son importance pratique en réduisant le temps d'apprentissage.

2.4 Réseau neuronal convolutif

L'étude du cortex visuel est étroitement liée au développement des réseaux neuronaux convolutionnels. En 1968, Hubel et Wiesel ont présenté une étude centrée sur les champs réceptifs du cortex visuel des singes [89]. Cette étude était pertinente en raison de la description de l'architecture du cortex strié (cortex visuel primaire) et de la façon dont les cellules y sont disposées.

De plus, il présentait également deux types de cellules différentes : **simple** et **complexe**. Les plus simples se concentrent sur les formes en bordure, tandis que les plus complexes couvrent un spectre plus large d'objets et sont localement invariants. Par conséquent, les différents ensembles de dispositions de cellules dans le cortex sont capables de cartographier l'ensemble du champ visuel en exploitant la corrélation des objets et des formes dans les zones visuelles locales.

L'une des premières implémentations inspirées par les idées de Hubel et de Wiesel était l'une des suivantes appelé Neocognitron. Neocognitronest un modèle de réseau de neurones développé par Kunihiko Fukushima en 1980[31]. La première couche du modèle est composée d'unités qui représentent des cellules simples, tandis que les unités de la deuxième couche représentent des cellules complexes.La mise en œuvre de la propriété d'invariance locale du cortex visuel est la plus importante. Réalisation de Neocognitron. De plus, le mappage de sortie est de un à un. Chaque des cartes cellulaires complexes à un seul et unique motif spécifique.

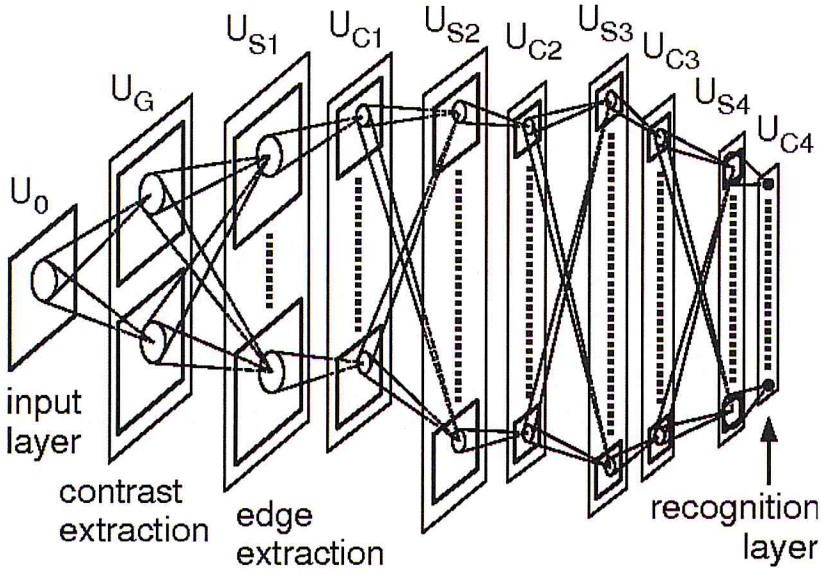


FIGURE 2.5: Néocognitron pour la reconnaissance de chiffres écrits à la main[31]

Cependant, l'un des principaux inconvénients de Neocognitron était son processus d'apprentissage[15] À cette époque, il n'y avait pas de méthode pour régler les valeurs de poids par rapport à une mesure d'erreur pour l'ensemble du réseau, comme la rétropropagation. Alors que le mathématicien Seppo Linnainmaa a dérivé la forme moderne de la rétropagation en 1970, son utilisation dans ANN n'a été appliquée qu'en 1985. Durant cette période, peu d'applications ont été développées en utilisant la rétropagation[3]. En 1985, les travaux de Rumelhart, Hinton et Williams[21] introduisent l'utilisation de la rétropagation dans ANN.

Conceptuellement, la rétropropagation mesure le gradient de l'erreur par rapport aux

poids sur les unités.
Le gradient changera à chaque fois que les valeurs de l'attribut des poids sont changés.
Ensuite, le gradient sera utilisé lors de la descente en pente afin de trouver des poids qui minimiseront l'erreur du réseau.
Lors de l'utilisation de la rétropropagation avec un optimiseur tel que Gradient Descent (GD), le réseau est capable de s'auto-régler ses paramètres.
GD est un algorithme d'optimisation de premier ordre. Il cherche un minimum d'une fonction prenant des mesures proportionnelles au négatif du gradient[12]. La figure 2.6 montre les étapes générales pour effectuer la rétropropagation.

Comme on l'a déjà dit, La première véritable application pratique de la rétropagation a été le classificateur de LeCun sur les chiffres manuscrits (MNIST)[77]. Ce système était l'une des utilisations les plus réussies de CNN à l'époque puisqu'il lisait un grand nombre de chèques manuscrits. La recherche de LeCun a conduit à des topologies CNN qui ont été utilisées comme source d'inspiration pour les futurs chercheurs, l'une des plus populaires étant LeNet-5[77].

LeNet-5 a été mis en œuvre dans le cadre d'une expérience de reconnaissance de documents. ça souligne l'idée que pour résoudre les problèmes de reconnaissance des formes, elle pourrait Il serait préférable d'utiliser des solutions d'apprentissage automatique plutôt que des solutions conçues à la main.

Puisque le traitement de tous les différents cas que les données d'entrée pourraient avoir sur une manière naturelle est une tâche assez complexe, l'apprentissage machine convient mieux à cet effet.
De plus, un système simple de reconnaissance de formes est décrit. Ce système se compose de deux éléments principaux modules : un extracteur de caractéristiques, qui transforme les données d'entrée en données de faible dimension. vecteurs ; et un classificateur, qui la plupart du temps est d'usage général et formable.

De plus, les composants clés de CNN sont décrites : champ réceptif local, partage de poids, opération de convolution, sous-échantillonnage spatial, décrochage et gradient stochastique descente.

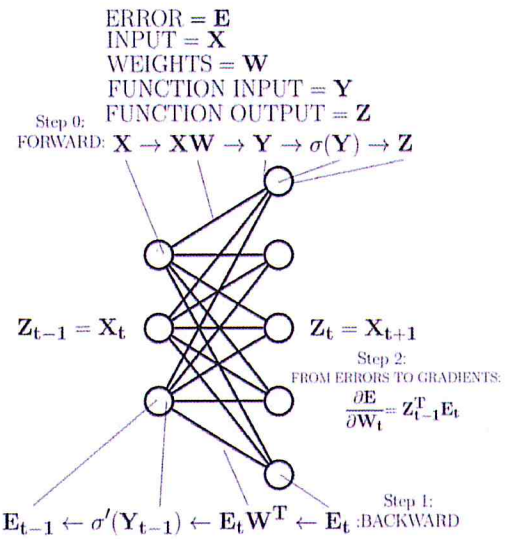


FIGURE 2.6: Algorithme de rétropagation[86] .

2.4.1 Opération de convolution

En mathématiques, une opération de convolution est définie comme une façon de mélanger deux fonctions[56]

Une analogie couramment utilisée est que cette opération fonctionne comme un filtre. Un noyau filtre tout ce qui n'est pas important pour la carte des caractéristiques, en se concentrant uniquement sur certaines informations spécifiques.

Pour exécuter cette opération, deux éléments sont nécessaires.

— Les données d'entrée

— Le filtre de convolution (noyau)

Le résultat de cette opération est une carte des caractéristiques. La figure 2.8 fournit une explication graphique sur la mécanique de l'opération convolutionnelle. Le nombre de cartes de caractéristiques (canaux de sortie) donne au réseau neuronal la capacité d'apprendre les caractéris-

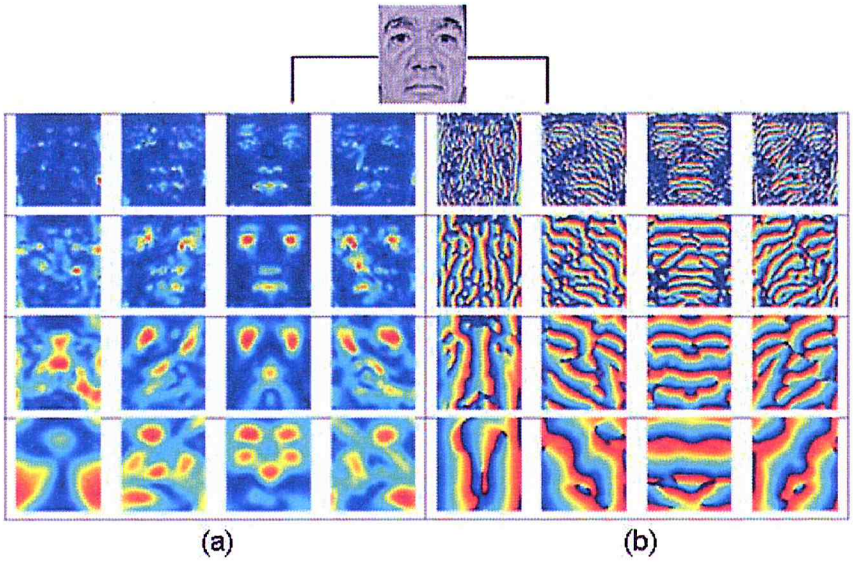


FIGURE 2.7: Opération de convolution[11]

tiques. Chaque canal est indépendant puisqu'ils visent à apprendre chaque nouvelle caractéristique à partir de l'image en cours de convolution.

Enfin, le type de rembourrage définit l'algorithme à utiliser lors de l'exécution de la convolution. Il y a un cas particulier sur les bords de l'entrée. Un type de remplissage rejettera la bordure d'entrée, puisqu'il n'y a plus d'entrée à côté de celle-ci qui peut être balayée. D'autre part, l'autre rembourrage complètera l'entrée avec une valeur de 0, il s'agit de réduire les paramètres tout en convolutant.

2.4.2 Le partage des poids

Comme il a été présenté précédemment, après l'opération de convolution, un plan composé de résultats de l'application du même filtre à travers toute l'entrée est générée. Ce plan s'appelle la carte des entités.

— Chaque carte de caractéristiques est le résultat d'une opération convolutionnelle avec un noyau.

— Les noyaux sont initialisés avec des poids différents afin de pouvoir percevoir différentes caractéristiques.

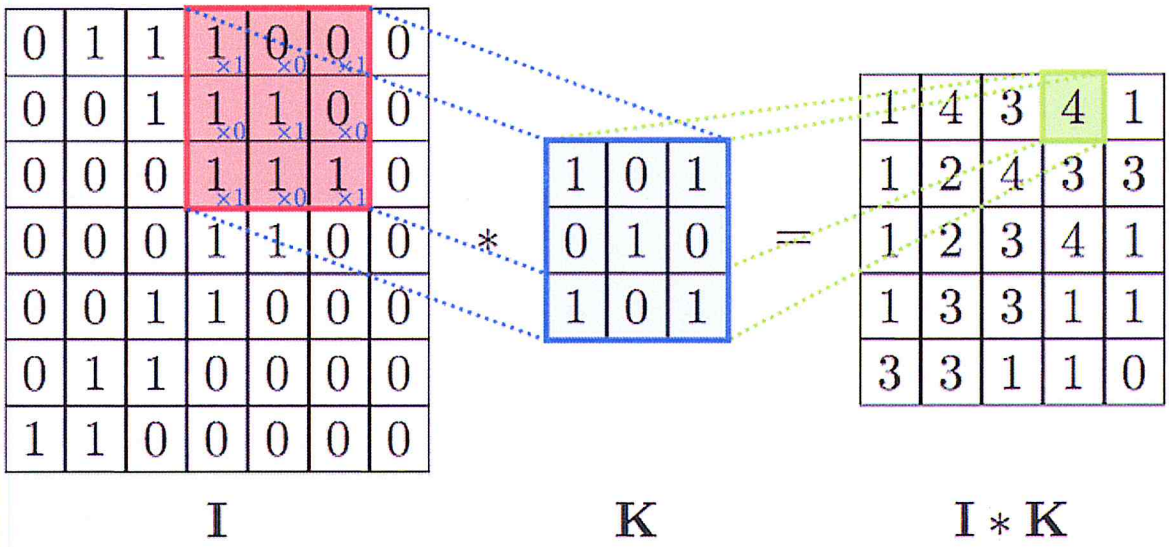


FIGURE 2.8: Processus de convolution pour un noyau de convolution 3 × 3[11].

— L'entité trouvée est conservée sur toute la carte de l'entité. sa position n'est pas pertinente pour le réseau.

Une couche convolutionnelle a généralement un ensemble de cartes de caractéristiques qui extraient différentes caractéristiques à chaque emplacement d'entrée, telles que définies par la taille du filtre. Le processus de balayage d'entrée, puis le stockage de l'état des unités sur la carte des caractéristiques est appelé convolution .

2.4.3 Champ réceptif local

Aussi connu sous le nom de taille de filtre ou de taille de noyau, un champ réceptif local est la zone à laquelle un neurone sera connecté sur l'entrée haute dimensionnelle. Le champ récepteur local est un hyperparamètre du réseau, c'est-à-dire que sa forme est définie à l'avance.

Ce concept a une forte influence des neurones sur le cortex visuel qui sont sensible au niveau local. L'idée n'est pas de connecter tous les neurones à tout l'espace d'entrée, mais de se concentrer sur les zones connectées localement. Ces connexions locales n'ont lieu que sur la largeur et la hauteur.La profondeur de l'entrée n'est pas connectée localement, mais entièrement connecté

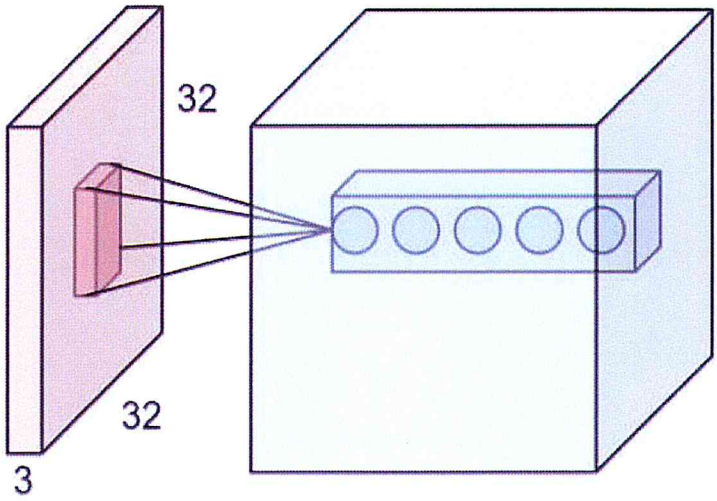


FIGURE 2.9: Champ récepteur local de taille 5x5x3 pour une image CIFAR-10 typique, 32x32x3[92]

à travers tous ses canaux.

Par exemple, une entrée haute dimensionnelle est une image. Une image est représentée par 3 dimensions : largeur, hauteur et profondeur. Quand un champ réceptif local est appliqué sur une image, son noyau n'agit que localement sur l'image. dimensions de largeur et de hauteur ; pas sur la profondeur, où il prend toutes les dimensions en compte. La dimension de profondeur est similaire au nombre de canaux.

Par exemple, un L'image RVB a 3 canaux : rouge, vert et bleu. L'image finale est une composition de toutes ces 3 images dans chaque couleur. En ayant l'entrée allouée de cette façon, les neurones sont capables d'extraire des caractéristiques élémentaires comme les bords, les points d'extrémité ou les coins. En appliquant cette idée aux couches suivantes, le réseau sera en mesure d'extraire des caractéristiques d'ordre supérieur. De plus, la réduction des connexions réduit également le nombre de paramètres, ce qui permet d'atténuer le surdimensionnement.

La figure 2.9 montre comment un seul neurone est connecté à une carte d'entités de taille 5x5x3. l'opération de convolution va se répéter à travers toute l'entrée (image) en utilisant ce filtre. Cela signifie que la largeur et la hauteur de l'entrée diminueront après l'opération ; mais ce n'est pas le cas pour la dimension de profondeur d'entrée.

Après l'opération de convolution, la dimension de profondeur sera le nombre de filtres appliqués à l'entrée. Un ensemble de filtres est initialisé pour capturer les différentes caractéristiques que l'on peut trouver dans l'image. Chaque filtre est initialisé avec des poids différents. Cependant, les poids restent les mêmes pour un filtre pendant qu'il se convolute à travers l'ensemble de l'entrée ; c'est ce qu'on appelle le partage des poids.

Il est important de rappeler que ces opérations étaient fortement axées sur l'apprentissage des caractéristiques, mais pas sur la classification. L'utilisation de couches entièrement connectées (également connues sous le nom de multicouches-perceptrons) et de réseaux convolutionnels offre les deux possibilités. L'avantage principal de ces couches est qu'elles peuvent être optimisées en utilisant la descente de gradient stochastique sur le style de propagation arrière, ainsi que les poids pour les couches convolutives.

2.4.4 Sous-échantillonnage spatial

Le sous-échantillonnage spatial est une opération également connue sous le nom de pooling. L'opération consiste à réduire les valeurs d'une zone donnée à une seule[]. Ainsi, il réduit l'influence de la position de l'entité sur la carte de l'entité en diminuant sa résolution spatiale. Cela se fait en choisissant le pixel le plus réactif après une opération de convolution.

Il existe deux types de pooling : **moyen** et **maximum**. Le moyen calcule la moyenne sur la zone définie, tandis que le maximum ne sélectionne que la valeur la plus élevée sur la zone.
La taille de la zone peut entraîner une réduction de la performance de prédiction, si la valeur est trop grande. Elle se déroule de la même façon qu'une opération de convolution, car un filtre et une foulée sont définis. Étant donné la région du filtre, cette opération renvoie le pixel avec la valeur la plus élevée.

Ainsi, la dimension de la carte des caractéristiques est réduite. Cette réduction empêche

le système d'apprendre les caractéristiques par position. Ensuite, il est utile de généraliser la fonctionnalité pour de nouveaux exemples. Ceci est important car les caractéristiques des nouveaux exemples peuvent se trouver sur des positions différentes.

2.4.5 Dropout

Le Dropout minimise l'impact des unités qui ont une forte activation. Cette méthode arrête les unités pendant l'entraînement, de sorte que les autres unités peuvent apprendre les caractéristiques par elles-mêmes[41]. Le fait d'accorder plus d'indépendance à toutes les unités réduit le fort biais de l'unité, ce qui mène à une forte régularisation et à une meilleure généralisation.

2.4.6 Descente Gradient Stochastique

La descente du gradient stochastique (SGD) n'a qu'une seule différence par rapport au gradient. Descente (GD). La différence est le nombre d'exemples pris en compte pour le calcul des gradients des paramètres. La version originale effectue cette opération à l'aide de tous les les exemples sur le set d'entraînement. La stochastique n'utilise que peu d'exemples définis. par la taille du lot[5] Il est important de noter que lors de l'utilisation de SGD, le taux d'apprentissage et sa diminution d'ordonnancement est plus difficile à définir par rapport à GD puisqu'il y a beaucoup plus de variance. dans la mise à jour du gradient[?].

2.5 Traveaux de recherche liés

Affectiva est le leader mondial de la recherche commerciale sur la reconnaissance des émotions. Son portefeuille de brevets actuel est le plus important par rapport aux entreprises en démarrage dans ce domaine. Leur recherche a adopté des méthodologies d'apprentissage en profondeur puisque son corpus privé comprend 3,2 millions de vidéos faciales. De plus, leur collecte de données a été faite dans 75 pays, ce qui empêche la recherche de tomber sur les comportements culturels ou régionaux.

Afin de mesurer la précision de son détecteur, la surface sous une caractéristique de fonctionnement du récepteur. (ROC) est utilisée. La valeur de la note ROC se situe entre 0 et 1. est plus précis lorsque la valeur est plus proche de 1. certaines émotions telles que la joie, le dégoût, le mépris et la surprise ont un score supérieur à 0,8.
Bien que des expressions telles que la colère, la tristesse et la peur atteignent une précision plus faible puisqu'elles sont plus nuancées et plus subtile. De plus, Affectiva a été en mesure d'identifier avec succès les unités d'action faciale. sur les expressions faciales spontanées sans utiliser de techniques d'apprentissage profond[73].

Dans les paragraphes suivants, des approches impliquant l'utilisation de l'ingénierie des caractéristiques sont présentées. Bien que les approches en matière d'extraction et de classification des caractéristiques soient différentes, elles ont toutes fait appel à l'ensemble de données de KDFE, dans le cadre de ses travaux. Il convient de mentionner que l'ensemble de données de KDFE a été utilisé dans cette recherche, de sorte que les résultats donnent une comparaison utile.
Kotsia et al[77]se sont concentrés sur l'effet de l'occlusion lors de la classification de 6 émotions faciales. Afin d'y parvenir, plusieurs techniques d'ingénierie des caractéristiques et de l'ingénierie ont été combinés. Les caractéristiques de Gabor, qui est un filtre linéaire utilisé pour la détection des contours et la factorisation matricielle non négative discriminante (DNMF) qui se concentre sur la non-négativité des données à traiter, sont les extracteurs de caractéristiques. techniques. Pour classer ces fonctionnalités, il faut utiliser les machines vectorielles de support multi-classes (SVM) et perceptron multicouche (MLP) ont été utilisés. Les résultats sur KDFE sont les suivants qui suivent : Utilisation d'une MLP avec Gabor 91,6% et avec DNMF : 86,7%. Lors de l'utilisation SVM a atteint 91,4 %.
Un autre corpus utilisé était JAFFE : Gabor combiné avec MLP. a atteint 88,1 % et lorsqu'on l'utilise avec la DNMF, il en est résulté une classification de 85,2%. l'exactitude.
Wang et Yin[84]ont examiné comment la distorsion de la région du visage détectée et les différentes intensités des expressions faciales affectent la robustesse de leur modèle. Les descripteurs d'expression de contexte Topographique (CT) ont été sélectionnés afin d'effectuer l'extraction de caractéristiques. Cette technique permet d'effectuer une analyse topographique. Dans cette analyse, l'image est comme une surface 3D. Chaque pixel est étiqueté en tenant compte des ca-

ractéristiques du terrain. L'utilisation de plusieurs classificateurs a été signalée : le classificateur discriminant quadratique (CDQ), classificateur de discrimination linéaire (LDA), classificateur de vecteur de support (SVC) et baies naïves. (NB). Résultats utilisant un sous-ensemble KDFE (70 sujets, 35 images par sujet pour chaque sujet. 4900 images) : avec QDC : 81,96 %, avec LDA : 82,68 %, avec NB : 76,120%, avec SVC : 77,68 %. Résultats de l'ensemble de données d'expression faciale MMI (5 sujets, 6 images). par sujet pour chaque expression. 180 images) ont également été signalées : avec QDC : 92,78%, avec la LDA : 93,33 %, et avec le N.-B. : 85,56

2.6 Conclusion

Ce chapitre s'intéresse au premier volet de recherche de notre projet, c'est le concept : Machine Learning. Nous avons présenté dans ce chapitre les trois visions de recherche liées au concept ML :

— Réseaux neuraux artificiels.

— Apprentissage Approfondi.

— Réseau neuronal convolutif.

Le but du prochain chapitre consiste à présenter le deuxième aspect de recherche du projet, il s'agit ANALYSE DE L'EXPRESSION FACIALE.

CHAPITRE 3

ANALYSE DE L'EXPRESSION FACIALE

Les émotions sont l'essence de ce qui fait de nous des êtres humains. Ils ont un impact sur nos routines quotidiennes, nos interactions sociales, notre attention, notre perception et notre mémoire.

L'un des indicateurs les plus forts des émotions est notre visage. Lorsque nous rions ou pleurons, nous mettons nos émotions en évidence, permettant aux autres d'entrevoir dans nos esprits pendant qu'ils "lisent" notre visage en fonction des changements dans les principales caractéristiques du visage comme les yeux, les sourcils, les paupières, les narines et les lèvres.

La reconnaissance d'expression faciale assistée par ordinateur imite de façon impressionnante nos capacités de codage humain, car elle capture des réponses émotionnelles brutes et non filtrées vers n'importe quel type de contenu émotionnellement engageant. Mais comment cela fonctionne-t-il exactement ?

3.1 Que sont les expressions faciales ?

Notre visage est une partie complexe et hautement différenciée de notre corps - en fait, c'est l'un des systèmes de signalisation les plus complexes dont nous disposons. Il comprend plus de 40 muscles structurellement et fonctionnellement autonomes, dont chacun peut être déclenché indépendamment l'un de l'autre.

Le système musculaire facial est le seul endroit de notre corps où les muscles sont attachés à un os et à un tissu facial (d'autres muscles du corps humain se connectent à deux os), ou à un tissu facial seulement, comme le muscle entourant les yeux ou les lèvres.

Evidemment, l'activité des muscles faciaux est hautement spécialisée dans l'expression, ce qui nous permet de partager l'information sociale avec les autres et de communiquer verbalement et non verbalement.

Le nerf facial contrôle la majorité des muscles faciaux

Tous les muscles de notre corps sont innervés par des nerfs qui vont jusqu'à la moelle épinière et au cerveau. La connexion nerveuse est bidirectionnelle, ce qui signifie que le nerf déclenche des contractions musculaires basées sur des signaux cérébraux (brain-to-muscle), tandis qu'il communique en même temps des informations sur l'état musculaire actuel au cerveau (muscle-to-brain)[37].

Presque tous les muscles du visage sont innervés par un seul nerf, donc simplement appelé nerf facial.

En termes un peu plus médicaux, le nerf facial est également connu sous "VII. nerf crânien"[46].

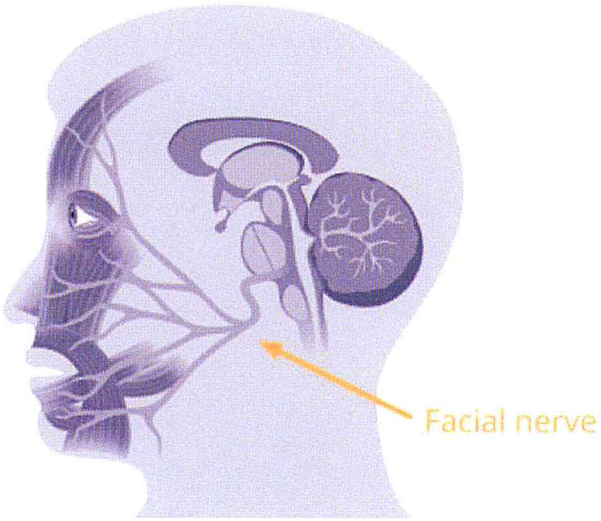


FIGURE 3.1: Nevers facial[46]

3.2 Expressions faciales et les émotions

L'hypothèse de la rétroaction faciale : Comme Fritz Strack et ses collègues l'ont découvert ingénieusement en 1988[75], les expressions faciales et les émotions sont étroitement liées. Dans leur étude, on a demandé aux répondants de tenir un stylo dans leur bouche tout en évaluant les caricatures pour leur contenu humoristique. Tandis qu'un groupe tenait le stylo entre les dents avec les lèvres ouvertes (imitant un sourire), l'autre groupe tenait le stylo avec les lèvres seulement (empêchant un bon sourire).

Voici ce que Fritz Strack a découvert : Le premier groupe a évalué le dessin animé comme étant plus humoristique. Strack et l'équipe ont pris ceci comme preuve pour l'hypothèse de rétroaction faciale postulant que l'activation sélective ou l'inhibition des muscles faciaux a un impact fort sur la réponse émotionnelle aux stimuli.

Émotions, sentiments, humeurs

Qu'est-ce que c'est exactement les émotions ?

dans le langage de tous les jours, les émotions sont des expériences conscientes relativement brèves caractérisées par une activité mentale intense et un degré élevé de plaisir ou de déplaisir.

Les émotions sont étroitement liées à l'excitation physiologique et psychologique avec différents niveaux d'excitation liés à des émotions spécifiques. En termes neurobiologiques, les émotions pourraient être définies comme des programmes d'action complexes déclenchés par la présence de certains stimuli externes ou internes[26].

Ces programmes d'action contiennent les éléments suivants :

— **Symptômes corporels,** tels qu'une augmentation de la fréquence cardiaque ou de la conductivité de la peau. La plupart de ces symptômes sont inconscients et involontaires.

— **Tendances d'action,** ou par exemple des actions de "combat ou de fuite" pour se soustraire immédiatement à une situation dangereuse ou pour préparer une attaque physique de l'adversaire.

— **Expressions faciales,** par exemple, se dénuder les dents et froncer les sourcils.

— **Evaluations cognitives** d'événements, de stimuli ou d'objets.

Les sentiments sont des perceptions subjectives des programmes d'action émotionnelle, les sentiments sont guidés par des pensées et des réflexions conscientes.nous pouvons certainement avoir des émotions sans avoir de sentiments, mais nous ne pouvons tout simplement pas avoir de sentiments sans avoir d'émotions[26].

Les humeurs sont des états internes subjectifs diffus, généralement moins intenses que les émotions et qui durent beaucoup plus longtemps. Les expressions faciales volontaires (sourire par exemple) peuvent produire des effets corporels similaires à ceux déclenchés par une émotion réelle (bonheur par exemple)[26].

L'effet décrit les émotions, les sentiments et les humeurs ensemble.

FIGURE 3.2: joie[27]

FIGURE 3.3: Colére[27]

FIGURE 3.4: Surprise[27]

FIGURE 3.5: Peure[27]

FIGURE 3.6: Neutral[27]

FIGURE 3.7: Tristesse[27]

FIGURE 3.8: Dégout[27]

La découverte que presque tout le monde peut produire et reconnaître les expressions faciales associées à ces émotions a conduit les chercheurs à supposer qu'elles sont universelles.

3.3 Domaines d'application

Avec la reconnaissance des expressions faciales, vous pouvez tester l'impact de tout contenu, produit ou service censé susciter des réactions émotionnelles et faciales - des objets

physiques tels que des sondes ou des emballages alimentaires, des vidéos et des images, des sons, des odeurs, des stimuli tactiles, etc.
Les expressions particulièrement involontaires ainsi qu'un élargissement subtil des paupières sont d'un intérêt clé car elles sont considérées comme reflétant les changements dans l'état émotionnel déclenchés par des stimuli externes réels ou des images mentales.

Quels domaines de la recherche commerciale et universitaire ont récemment adopté des techniques de reconnaissance des émotions ? Voici un aperçu des domaines de recherche les plus importants :

— **Neurosciences grand public et neuromarketing**[37]

Il n'y a aucun doute : l'évaluation des préférences des consommateurs et la communication persuasive sont des éléments critiques du marketing.
Bien que les autodéclarations et les questionnaires puissent être des outils idéaux pour obtenir un aperçu des attitudes et de la prise de conscience des répondants, ils peuvent être limités dans la capture de réponses émotionnelles non biaisées par la conscience de soi et la désirabilité sociale.
C'est là qu'intervient la valeur de l'analyse des émotions : le suivi des expressions faciales peut être utilisé pour enrichir substantiellement les auto-évaluations avec des mesures quantifiées des réponses émotionnelles plus inconscientes vers un produit ou un service. L'analyse de l'expression faciale permet d'optimiser les produits, d'évaluer les segments de marché et d'identifier les publics cibles et les personnes. Il y a beaucoup de choses que l'analyse de l'expression faciale peut faire pour vous afin d'améliorer votre stratégie de marketing.

— **Essais et publicité dans les médias**

Dans la recherche sur les médias, les répondants individuels ou les groupes de dis-

cussion peuvent être exposés à des publicités télévisées, des bandes-annonces et des pilotes de longue durée tout en surveillant leurs expressions faciales. Identifier des scènes où l'on s'attendait à des réactions émotionnelles (en particulier des sourires), mais où le public ne l'a pas " compris " est aussi crucial que de trouver les images clés qui donnent lieu aux expressions faciales les plus extrêmes.

Dans ce contexte, vous pourriez vouloir isoler et améliorer les scènes qui déclenchent des expressions négatives indésirables indiquant des niveaux élevés de dégoût, de frustration ou de confusion (ce genre d'émotions n'aiderait pas exactement une émission comédie à devenir une série à succès, n'est-ce pas ? ou utiliser la réponse de votre auditoire lors d'une projection afin d'augmenter le niveau global d'expressions positives dans le communiqué final.

— **Recherche psychologique**[64]

Les psychologues analysent les expressions faciales pour identifier comment les humains réagissent émotionnellement aux stimuli externes et internes. Dans les études systématiques, les chercheurs peuvent faire varier les propriétés des stimuli (couleur, forme, durée de la présentation) et les attentes sociales afin d'évaluer l'impact des caractéristiques de la personnalité et des histoires d'apprentissage individuelles sur les expressions faciales.

— **Psychologie clinique et psychothérapie**[64]

Les populations cliniques telles que les patients souffrant de troubles du spectre autistique (TSA), de dépression ou de trouble de la personnalité limite (TPL) sont caractérisées par de fortes déficiences dans la modulation, le traitement et l'interprétation de leurs propres expressions faciales et de celles des autres. La surveillance des expressions faciales pendant que les patients sont exposés à des stimuli émotionnellement stimulants ou à des indices sociaux (visages d'autres personnes, par exemple)

peut accroître de façon significative le succès de la thérapie cognitivo-comportementale sous-jacente, tant pendant la phase de diagnostic que pendant la phase d'intervention. Un excellent exemple est le "Labyrinthe du sourire" développé par le Temporal Dynamics of Learning Center (TDLC) à UC San Diego. Ici, les enfants autistes entraînent leurs expressions faciales en jouant à un jeu de type Pacman où le sourire dirige le personnage du jeu.

— **Applications médicales et chirurgie plastique**[70]

Les effets de la paralysie du nerf facial peuvent être dévastateurs. Les causes comprennent la paralysie de Bell [37], les tumeurs, les traumatismes, les maladies et les infections. Les patients sont généralement aux prises avec des changements importants dans leur apparence physique, la capacité de communiquer et d'exprimer des émotions. L'analyse de l'expression faciale peut être utilisée pour quantifier la détérioration et évaluer le succès des interventions chirurgicales, de l'ergothérapie et de la physiothérapie visant à réactiver les groupes musculaires paralysés.

— **Conception de l'interface utilisateur du logiciel et du site Web**

Idéalement, le maniement des logiciels et la navigation sur les sites Web devraient être une expérience agréable - les niveaux de frustration et de confusion devraient certainement être maintenus aussi bas que possible. La surveillance des expressions faciales pendant que les testeurs naviguent sur des sites Web ou des dialogues de logiciels peut fournir un aperçu de la satisfaction émotionnelle du groupe cible désiré. Chaque fois que les utilisateurs rencontrent des barrages routiers ou se perdent dans des sous-menus complexes, vous pouvez certainement constater une augmentation des expressions faciales "négatives" telles que le froncement des sourcils ou le froncement des sourcils.

— **Ingénierie des agents sociaux artificiels (avatars)**

Jusqu'à récemment, les robots et les avatars étaient programmés pour répondre aux commandes de l'utilisateur en fonction du clavier et de la souris. Les dernières avancées en matière de technologie matérielle, de vision par ordinateur et d'apprentissage automatique ont jeté les bases d'agents sociaux artificiels, capables de détecter et de réagir de manière fiable et flexible aux états émotionnels du partenaire de communication humain. Le Siri d'Apple pourrait bien être la première génération de machines vraiment intelligentes sur le plan émotionnel, mais les informaticiens, les médecins et les neuroscientifiques du monde entier travaillent dur sur des capteurs et des algorithmes encore plus intelligents pour comprendre l'état émotionnel actuel de l'utilisateur humain et pour réagir de manière appropriée.

3.4 Conclusion

Dans ce chapitre nous avons montrés ce que les émotion sont aussi que leurs applications dans plusieur domaine et cela pour conclure notre état de l'art et commencer a conceptualiser un systeme pour répondre a la problématique

CHAPITRE 4

LA SOLUTION PROPOSÉE

Ce chapitre est divisé en deux sections principales, chacune correspondant aux deux phases de travail . Dans chaque section, on décrit comment l'image est prétraité, les raisons du choix d'une topologie et de valeurs particulières pour plusieurs paramètres, et comment la précision du réseau est évaluée. Ce chapitre décrit comment les techniques et les concepts décrits jusqu'à présent dans le rapport sont utilisées et integrées pour une nouvelle solution à notre probléme.

4.1 Première phase : Pétraitement et Entrainemet

Au cours de cette phase, le travail autour de l'ensemble de Karolinska Directed Emotional Faces (KDEF) est présenté. Il comprend :

— Définition de la topologie du réseau.

— Le prétraitement des images KDEF.

— L'alimentation du réseau avec les images KDEF traiter

— La formation du réseau.

— Les paramètres de réglage.

— L'évaluation de la précision du réseau

4.1.1 pré-traitement des images

Le pré-traitement de l'image est présenté dans les paragraphes suivants. L'entrée est le data-set d'image, et la sortie est un fichier binaire. Le fichier binaire a été utilisé pour alimenter le réseau pendant la formation. C'est le début de la partie expérimentale du projet. Au départ, certaines complications sont apparues en raison de la structure des dossiers de l'ensemble de données et de l'absence d'étiquettes pour certaines séquences d'images. En utilisant l'algorithme walk Python ,

— tous les fichiers image trouvés dans le répertoire de l'arborescence KDEF ont été déplacés dans le même dossier de destination.Ceci était possible étant donné que le nom de fichier était suffisant pour identifier chaque image à sa séquence et son sujet correspondants

— Une approche similaire a été utilisée pour déplacer les fichiers d'étiquettes.

— Toutes les séquences d'images sans étiquette ont été déplacées dans un dossier séparé.

— Chaque séquence d'images étiquetées commence par un visage neutre et se termine par l'image de l'expression faciale.

— Comme on peut le déduire, les premières images ne sont pas aussi significatives pour le processus de formation puisque aucune émotion faciale n'y est affichée. Afin de minimiser l'impact de ces images, les 3 premières images de chaque séquence ont été rejetées et seules les 10 dernières images ont été prises en compte pour la formation. Ces nombres ont été définis comme une heuristique puisque toutes les séquences n'ont pas le même nombre d'images, ni la même distribution d'affichage des émotions pour chaque image.

La taille de l'ensemble de données a changé en fonction des différentes sélections qui lui ont été appliquées :

— Un total de 4480 images étiquetées ; après avoir exclu les 3 premières images de chaque séquence.

— 3064 images étiquetées ; il suffit de prendre en compte les 10 dernières images par séquence.

— 1538 labeled images ; after considering only the last 5 images.

Toutes les opérations effectuées sur les paragraphes suivants n'ont été appliquées qu'à l'ensemble des 3064 images étiquetées, car elles étaient les seules utilisées pendant la formation.

—

L'étape suivante consiste à **dimensionner les images**. Le cadrage s'est fait sur deux valeurs : 32 pixels (similaire aux images CIFAR-10) et 64 pixels.

L'idée est de les comparer pendant l'entrainement. Avant de convertir cette image en fichier binaire, elles étaient converties en images en niveaux de gris. Python Imaging Library (PIL) a été l'outil choisi pour accomplir cette tâche. Bien que les caractéristiques liées à la couleur puissent être intéressantes à explorer, elles augmentent également le nombre de paramètres (poids et biais) sur le réseau par ordre de grandeur.

Plus de paramètres impliquent un temps d'entraînement plus long et une chance de sur-ajustement plus élevé. Ce sont là les principales raisons d'utiliser des images en niveaux de gris.

Cette transformation uniformise les données d'entrée en taille (largeur et hauteur) et en nombre de canaux (profondeur). Ceci est important car la base de données contient des images appartenant au corpus original, et celui qui a été prolongé. Les images sur le corpus original ont été enregistrées dans des conditions différentes.

La dernière étape consiste à créer un fichier binaire à partir de cet ensemble d'images. La nécessité de créer un fichier binaire contenant l'étiquette et l'image a été générée par des tentatives infructueuses de chargement séparé des images et des étiquettes dans TensorFlow.

Pour générer le fichier bin, un dictionnaire d'étiquettes est généré. Ce dictionnaire a été utilisé pour faire correspondre chaque séquence avec l'étiquette correspondante. Le dictionnaire est représenté par une liste de listes. De plus, une taille d'enregistrement est définie pour chaque exemple.

Cette taille est définie comme la somme des octets de l'étiquette et du produit de hauteur, largeur et canal de l'image. Pour les images de 32 pixels, la valeur est de 1025 octets par image, Alors qu'il est de 4097 sur des images de 64 pixels.

la 4eme etape, le fichier bin est généré en transformant chaque image et étiquette en un tableau

Numpy. Tous ces tableaux Numpy sont stockés dans un vecteur final, qui a un nombre fixe de positions similaires au nombre d'images en cours de traitement. En fin de compte, le vecteur résultant est matérialisé dans un fichier.

4.1.2 Augmentation des données

L'un des principaux inconvénients de l'apprentissage supervisé est le besoin de données étiquetées. Le travail manuel lié à l'étiquetage des données exige que de nombreuses personnes suivent une procédure stricte d'un ensemble de règles[54]. Plus l'ensemble de données est grand, plus il est complexe à l'étiqueter.

L'apprentissage en profondeur nécessite de grandes quantités de données pour la formation. Puisqu'il s'agit d'une tâche très coûteuse, l'augmentation des données s'est avérée être un moyen efficace d'élargir l'ensemble de données [85] [88]. Un petit ensemble de données peut conduire le réseau à une adaptation excessive ou insuffisante[77].

De plus, ils aident à mieux couvrir l'espace d'exemple, au lieu de se concentrer uniquement sur la région délimitée par l'ensemble de données origine. Ainsi, la formation des réseaux à l'aide de l'augmentation des données généralisera mieux lorsqu'ils sont exposés à de nouveaux exemples.

L'augmentation des données consiste à appliquer la transformation sur le corpus. Dans ce cas, les transformations ont été appliquées sur des images KDEF. Modifier les propriétés de l'image aide à exploiter les caractéristiques invariantes que le réseau peut apprendre.

TensorFlow fournit un ensemble de fonctions adaptées à la transformation de l'image :

- Retourner l'image de gauche à droite et régler la luminosité et le contraste. Tous les paramètres ont été définis à la suite de la configuration du tutoriel TF sur les réseaux neuronaux convolutionnels .
- L'opération de blanchiment est effectuée sur l'image. L'opération de blanchiment calcule la valeur moyenne des pixels et ensuite, il soustrait cette valeur de l'opération de blanchiment de l'image. Par conséquent, la moyenne du pixel est centrée autour de la valeur de zéro.

4.2 Description du réseau

La topologie à utiliser s'inspire de celle développée sur Visual Geometry Group (VGG) à l'Université d'Oxford[91]. Le réseau de VGG a été attribué en tant que finaliste sur la compétition d'ImageNet 2014, et il a souligné l'importance de la profondeur du réseau pour améliorer l'exactitude de la classification.
De plus, VGG s'est avéré être une bonne généralisation à d'autres pays. et des ensembles de données, en comparaison avec d'autres types d'architectures[91]. Ce réseau a été choisi parce qu'il met fortement l'accent sur l'utilisation d'un réseau convolutionnel pur. par rapport à d'autres topologies de pointe comme GoogleNet.
Après avoir terminé le traitement d'image, l'étape suivante consiste à **créer des lots d'images**. pour alimenter le réseau. La taille du lot est un autre paramètre à prendre en compte.

Puisque l'optimisation du réseau est réalisée par la descente stochastique de gradient (SGD) avec l'élan, le choix d'une taille de lot correcte est critique.
Quelques aspects à prendre en compte pour définir la taille du lot sont la taille de l'ensemble de données et la disponibilité du matériel. Il s'agirait être optimal pour prendre en compte l'ensemble des données à chaque étape afin d'optimiser l'ensemble des données en vue d'atteindre les objectifs suivants le gradient ; toutefois, une telle approche serait coûteuse en temps et en argent, Il est courant de trouver dans la littérature que la taille du lot est une puissance. de deux. Par exemple, sur Krizhevsky et al[55], la taille du lot est de 128. Dans ce projet, la taille du lot a été fixée à 64. Cette valeur a été définie après avoir essayé plusieurs valeurs. Il a fourni un bon compromis entre le temps de formation et la réduction des pertes.

4.2.1 Topologie

La topologie du réseau est présentée dans dans la figure 4.1 . Le réseau est composé de 4 types d'opérations différentes :

— 3 couche convolutionnelle.

— 3 opération de MaxPooling.

— 2 couches entièrement connectées

— Une couche softmax

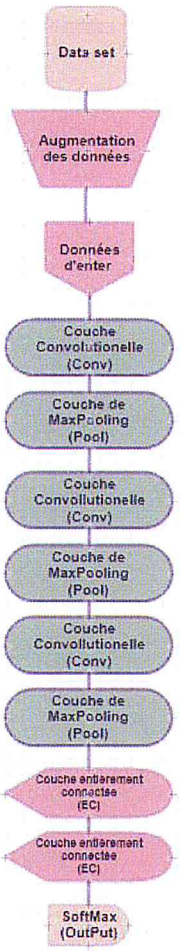


FIGURE 4.1: Topologie du réseau pour l'ensemble de données KDEF sur la phase 1.

4.2.2 Spécification des couches

Comme nous l'avons décrit ci-dessus, un ConvNet simple est une séquence de couches, et chaque couche d'un ConvNet transforme un volume d'activations en un autre à travers une fonction différentiable. Nous utilisons trois principaux types de couches pour construire des architectures ConvNet :

- Couche convolutive.
- Couche de Pooling.
- Couche entièrement connectée.
- Couche SoftMax.

Nous allons empiler ces couches pour former notre architecture proposée complète.Nous entrerons dans les détails ci-dessous, mais un simple ConvNet pour la classification de notre Convnet elle pourrait avoir l'architecture[INPUT - CONV - POOL - EC]. Plus de détail et présenté comme suit :

- INPUT[64x64x3] contiendra les valeurs brutes des pixels de l'image, dans ce cas une image de largeur 64, hauteur 64, et avec trois canaux de couleur R,G,B.
- La couche CONV calculera la sortie des neurones qui sont connectés aux régions locales dans l'entrée, chacun calculant un produit de point entre leur poids et une petite région à laquelle ils sont connectés dans le volume d'entrée. Cela peut se traduire par un volume tel que[64x64x32] si nous avons décidé d'utiliser 32 filtres.
- La couche POOL effectuera une opération de sous-échantillonnage le long des dimensions spatiales (largeur, hauteur), résultant en un volume tel que[32x32x32].
- EC (c'est-à-dire entièrement connecté) calcule les scores de classe, ce qui donne un volume de taille[1x1x7], où chacun des 7 nombres correspond à un score d'une classe, par exemple parmi les 7 catégories de notre ConvNet. Comme avec les réseaux de neurones ordinaires et comme le nom l'indique, chaque neurone de cette couche sera connecté à tous les nombres du volume précédent.

De cette façon, ConvNet transforme l'image d'origine couche par couche des valeurs de pixels d'origine aux scores finaux de classe. Notez que certaines couches contiennent des paramètres et d'autres pas.

En particulier, les couches CONV/FC effectuent des transformations qui sont des fonctions non seulement des activations du volume d'entrée, mais aussi des paramètres (poids et biais des neurones).
D'autre part, la couche POOL mets en œuvre une fonction fixe. Les paramètres des couches CONV/FC seront entraînés avec descente de gradient stochastique afin que les scores de classe que le modèle proposé calcule soient cohérents avec les étiquettes dans le jeu de formation pour chaque image.

- Une architecture ConvNet est dans le cas le plus simple une liste de calques qui transforment le volume de l'image en volume de sortie (par exemple en tenant les scores de classe)
- Il existe quelques types de couches distinctes (par exemple, CONV/FC/RELU/POOL sont de loin les plus populaires).
- Chaque couche accepte un volume d'entrée 3D et le transforme en volume de sortie 3D grâce à une fonction différentiable.
- Chaque couche peut avoir ou non des paramètres (par exemple, CONV/FC ont a, POOL n'a pas).
- Chaque couche peut ou non avoir des hyperparamètres supplémentaires (par exemple CONV/FC/POOL on n'ont).

4.2.3 La description de l'architecture globale de notre system

Couche convolutive

La couche Conv est la pierre angulaire d'un réseau convolutionnel qui effectue la plupart des calculs de levage lourd.

Vue d'ensemble et intuition sans trucs de cerveau. Discutons d'abord de ce que la couche CONV calcule sans analogies cerveau/neurones. Les paramètres de la couche CONV sont constitués d'un ensemble de filtres apprentissables.
Chaque filtre est petit dans l'espace (la longer, la largeur et la hauteur), mais s'étend sur toute la

profondeur du volume d'entrée.

Par exemple, un filtre typique sur une première couche d'un ConvNet peut avoir une taille de 3x3x3 (c'est-à-dire une largeur et une hauteur de 3 pixels, et 3 parce que les images ont une profondeur de 3, les canaux de couleur).

Pendant le passage vers l'avant, nous glissons (plus précisément, convolution) chaque filtre sur la largeur et la hauteur du volume d'entrée et calculons les produits de points entre les entrées du filtre et l'entrée à n'importe quelle position.

En glissant le filtre sur la largeur et la hauteur du volume d'entrée, nous produirons une carte d'activation bidimensionnelle qui donne les réponses de ce filtre à chaque position spatiale.

Intuitivement, le réseau apprendra les filtres qui s'activent lorsqu'ils voient un certain type de caractéristique visuelle comme un bord d'une orientation ou une tache d'une certaine couleur sur la première couche, ou éventuellement des motifs entiers en nid d'abeilles ou en forme de roue sur les couches supérieures du réseau. Maintenant, nous aurons un ensemble complet de filtres dans chaque couche de CONV (par exemple 32 filtres), et chacun d'eux produira une carte d'activation bidimensionnelle séparée. Nous empilerons ces cartes d'activation le long de la dimension de profondeur et produirons le volume de sortie.

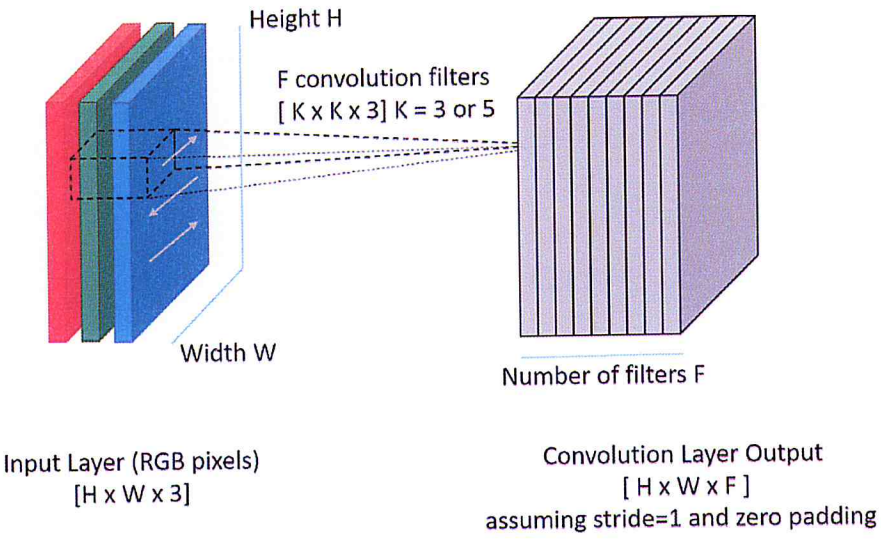


FIGURE 4.2: Architecture d'une image(RGB) et un filtre[77]

La vue du cerveau/neuron, chaque entrée dans le volume de sortie 3D peut aussi être

interprétée comme une sortie d'un neurone qui ne regarde qu'une petite région dans l'entrée et partage les paramètres avec tous les neurones à gauche et à droite dans l'espace (puisque ces nombres résultent tous de l'application du même filtre). Nous discutons maintenant des détails des connectivités des neurones, de leur disposition dans l'espace et de leur schéma de partage des paramètres.

Connectivité locale.

Lorsqu'il s'agit d'entrées à haute dimension comme les images, comme nous l'avons vu plus haut, il n'est pas pratique de connecter les neurones à tous les neurones du volume précédent. Au lieu de cela, nous connecterons chaque neurone à seulement une région locale du volume d'entrée. L'étendue spatiale de cette connectivité est un hyperparamètre appelé champ réceptif du neurone (ce qui correspond à la taille du filtre).
L'étendue de la connectivité le long de l'axe de profondeur est toujours égale à la profondeur du volume d'entrée. Il est important de souligner à nouveau cette asymétrie dans la façon dont nous traitons les dimensions spatiales (largeur et hauteur) et la dimension de profondeur : Les connexions sont locales dans l'espace (largeur et hauteur), mais toujours pleines sur toute la profondeur du volume d'entrée.
Dans notre cas le volume d'entrée a une taille[64x64x3]. Si le champ réceptif (ou la taille du filtre) est de 3x3, alors chaque neurone de la couche Conv aura des poids pour une région[3x3x3] dans le volume d'entrée, pour un total de 3*3*3 = 27 poids (et +1 paramètre de biais). Notez que l'étendue de la connectivité le long de l'axe de profondeur doit être de 3, puisqu'il s'agit de la profondeur du volume d'entrée.

à gauche de la : Un exemple de volume d'entrée en rouge (par exemple une image 64x64x3 EmoDeep), et un exemple de volume de neurones dans la première couche convolutionnelle. Chaque neurone de la couche convolutionnelle n'est relié qu'à une région locale dans le volume d'entrée spatialement, mais à la profondeur totale (c'est-à-dire tous les canaux de couleur). Notez qu'il y a plusieurs neurones (5 dans cet exemple) le long de la profondeur, tous regardant la même région dans l'entrée : Les neurones du Réseau de neurones restent inchangés : ils calculent toujours un

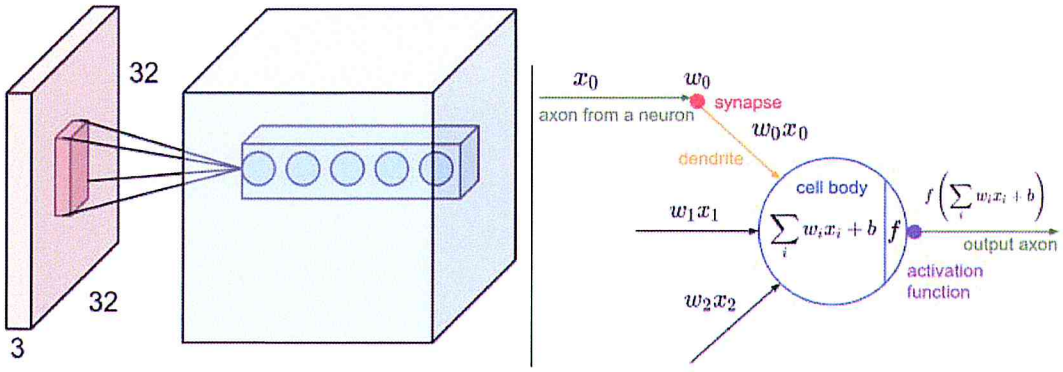


FIGURE 4.3: Un exemple de volume d'entrée en rouge[77]

produit de points par leur poids avec l'entrée suivie d'une non-linéarité, mais leur connectivité est maintenant restreinte pour être locale dans l'espace.à gauche : Un exemple de volume d'entrée en rouge (par exemple une image 64x64x3 EmoDeep), et un exemple de volume de neurones dans la première couche convolutionnelle. Chaque neurone de la couche convolutionnelle n'est relié qu'à une région locale dans le volume d'entrée spatialement, mais à la profondeur totale (c'est-à-dire tous les canaux de couleur). Notez qu'il y a plusieurs neurones (5 dans cet exemple) le long de la profondeur, tous regardant la même région dans l'entrée . A droite de la figure : Les neurones du Réseau de neurones restent inchangés : ils calculent toujours un produit de points par leur poids avec l'entrée suivie d'une non-linéarité, mais leur connectivité est maintenant restreinte pour être locale dans l'espace.

Arrangements spatiaux

Nous avons expliqué la connectivité de chaque neurone de la couche Conv au volume d'entrée, mais nous n'avons pas encore discuté du nombre de neurones présents dans le volume de sortie ou de leur disposition. Trois hyperparamètres contrôlent la taille du volume de sortie : la profondeur, la foulée et le zero-padding. Le controle de la taille du volume de sortie est resumée en :

— Tout d'abord, **la profondeur** du volume de sortie est un hyperparamètre : elle cor-

respond au nombre de filtres que nous aimerions utiliser, chacun apprenant à chercher quelque chose de différent dans l'entrée. Par exemple, si la première couche convolutionnelle prend comme entrée l'image brute, alors différents neurones le long de la dimension de profondeur peuvent s'activer en présence de divers bords orientés ou de taches de couleur. Nous nous référerons à un ensemble de neurones qui regardent tous la même région de l'entrée comme une **colonne de profondeur** (certaines personnes préfèrent aussi le terme fibre).

— Deuxièmement, il faut préciser **la foulée** avec laquelle on glisse le filtre. Lorsque la foulée est de 1, nous déplaçons les filtres un pixel à la fois. Lorsque la foulée est de 2 (ou rarement 3 ou plus, bien que cela soit rare dans la pratique), les filtres sautent 2 pixels à la fois au fur et à mesure que nous les faisons glisser. Cela produira des volumes de production plus petits dans l'espace.

— Comme nous le verrons bientôt, il sera parfois pratique de remplir le volume d'entrée avec des zéros autour de la frontière. La taille de ce **zero-padding** est un hyperparamètre. L'avantage du zero padding est qu'il nous permettra de contrôler la taille spatiale des volumes de sortie (le plus souvent, comme nous le verrons bientôt, nous l'utiliserons pour préserver exactement la taille spatiale du volume d'entrée afin que la largeur et la hauteur de l'entrée et de la sortie soient les mêmes).

Par exemple, pour une entrée 64x64 et un filtre 3x3 avec foulée 1 et pad 0, on obtiendrait une sortie 31x31. Avec la foulée 2, nous obtiendrions une sortie 21x21. Voyons aussi un autre exemple graphique :

Dans cet exemple présenter dans la figure 4.2, il n'y a qu'une seule dimension spatiale (axe des x), un neurone avec une taille de champ réceptif de F = 3, la taille d'entrée est W = 5, et il n'y a aucun remplissage de P = 1. Gauche : Le neurone a traversé l'entrée en foulée de S = 1, donnant une sortie de taille (5 - 3 + 2)/1+1+1 = 5. C'est vrai : Le neurone utilise la foulée de S = 2, ce qui donne une sortie de taille (5 - 3 + 2)/2/2+1 = 3. Notez que la foulée S = 3 n'a pas pu être utilisée car elle ne s'adapterait pas parfaitement au volume. En termes d'équation, cela peut être déterminé puisque (5 - 3 + 2) = 4 n'est pas divisible par 3. Les poids des neurones sont dans cet exemple[1,0,-1]. (montré à droite), et son biais est nul. Ces poids sont partagés entre tous les neurones jaunes (voir le partage des paramètres ci-dessous). Utilisation du zero-padding.

Dans l'exemple ci-dessus à gauche, notez que la dimension d'entrée était de 5 et la dimension de sortie était égale : 5. cela a fonctionné ainsi parce que nos champs réceptifs étaient de 3 et nous avons utilisé un remplissage de zéro de 1. S'il n'y avait pas eu de remplissage de zéro, alors le volume de sortie aurait eu une dimension spatiale de seulement 3, parce que c'est le nombre de neurones qui aurait "s'adapterait" à l'entrée originale. En général, le réglage du rembourrage de zéro à P $(F-1)/2$ lorsque la foulée est S 1 garantit que le volume d'entrée et le volume de sortie auront la même taille dans l'espace. Il est très courant d'utiliser le zero-padding de cette façon et nous discuterons de toutes les raisons lorsque nous parlerons plus en détail des architectures ConvNet.

Contraintes sur les foulées

Notons encore une fois que les hyperparamètres d'agencement spatial ont des contraintes mutuelles. Par exemple, lorsque l'entrée a une taille W 10, aucun remplissage zéro n'est utilisé P 0, et que la taille du filtre est F 3, alors il serait impossible d'utiliser la foulée S 2, puisque $(W-F\ 2P)/S\ 1\ (10-3\ 0)/2\ 1\ 4.5$, c'est-à-dire pas un nombre entier, indiquant que les neurones ne s'ajustent pas de façon nette et symétrique à travers l'entrée. Par conséquent, ce paramètre des hyperparamètres est considéré comme invalide, et une bibliothèque ConvNet pourrait lancer une exception ou un pad zéro le reste pour l'ajuster, ou recadrer l'entrée pour l'ajuster, ou quelque chose comme ça. Comme nous le verrons dans la section Architectures ConvNet, dimensionner les ConvNets de manière appropriée pour que toutes les dimensions "travailler" puisse être un véritable casse-tête, ce que l'utilisation du zero-padding et de quelques lignes directrices de conception permettra d'alléger de manière significative .

Pooling Layer

Il est courant d'insérer périodiquement une couche Pooling entre les couches successives de Conv dans une architecture ConvNet. Sa fonction est de réduire progressivement la taille spa-

tiale de la représentation pour réduire le nombre de paramètres et de calculs dans le réseau, et donc de contrôler également le suréquipement. Le Pooling Layer fonctionne indépendamment sur chaque tranche de profondeur de l'entrée et la redimensionne dans l'espace, en utilisant l'opération MAX. La forme la plus courante est une couche de pooling avec des filtres de taille 2x2 appliqués avec une foulée de 2 downsamples chaque tranche de profondeur dans l'entrée par 2 le long de la largeur et de la hauteur, rejetant 75% des activations. Chaque opération MAX prendrait dans ce cas un maximum de 4 chiffres (petite région de 2x2 dans une tranche de profondeur). La dimension de profondeur reste inchangée. Plus généralement, la couche de pooling :

- Nécessite deux hyperparamètres :
  - leur étendue spatiale (F),
  - le pas (S),
- Produit un volume de taille $W_2 \times H_2 \times D_2 \times D_2$ où :
  - W_2 $(W_1 - F)/S$ 1
  - H_2 $(H_1 - F)/S$ 1
  - D_2 D_1
- Introduit des paramètres zéro puisqu'il calcule une fonction fixe de l'entrée.
- Notez qu'il n'est pas courant d'utiliser le zero-padding pour les couches Pooling.

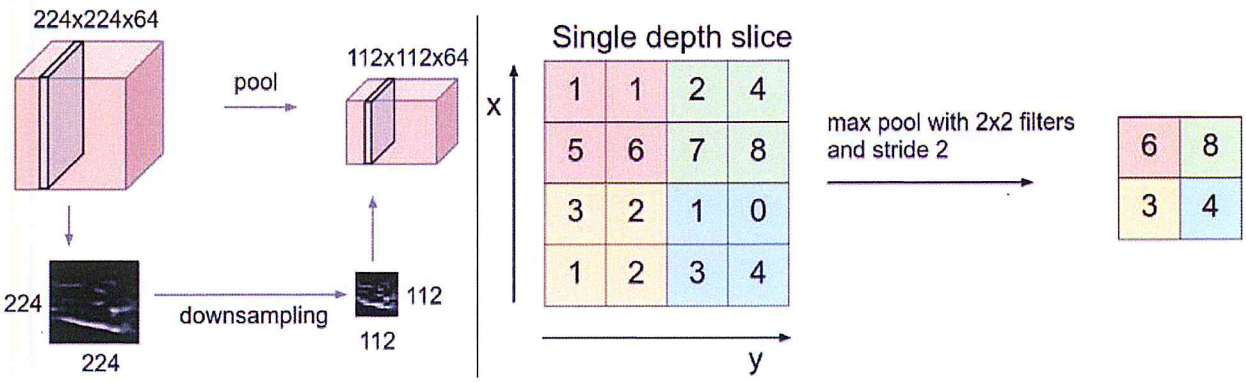


FIGURE 4.4: Couche de Pooling.[77]

La couche de mise en commun sous-échantillonne le volume dans l'espace, indépendamment dans chaque tranche de profondeur du volume d'entrée. Gauche : Dans cet exemple, le

volume d'entrée de la taille[224x224x64] est regroupé avec la taille de filtre 2, le pas 2 dans le volume de sortie de la taille[112x112x64]. Notez que la profondeur du volume est préservée. C'est vrai : L'opération la plus courante est max, donnant lieu à un max pooling, ici montré avec une foulée de 2, c'est-à-dire que chaque max est pris sur 4 numéros (petit carré de 2x2).

Couche entièrement connectée

Les neurones d'une couche entièrement connectée ont des connexions complètes à toutes les activations de la couche précédente, comme on le voit dans les réseaux neuronaux réguliers. Leurs activations peuvent donc être calculées avec une multiplication matricielle suivie d'un décalage de biais.

Récapitulation

Le couche d'entrée (qui contient l'image) devrait être divisible par 2 plusieurs fois. Les numéros communs comprennent 32 (p. ex. CIFAR-10), 64, 96 (EmoDeep) ou 224 (ImageNet ConvNets), 384 et 512.

Les couches conv devraient utiliser de petits filtres (Dans notre cas 3x3), en utilisant une foulée de S 1, et surtout, en remplissant le volume d'entrée avec des zéros de telle sorte que la couche conv n'altère pas les dimensions spatiales de l'entrée. C'est à dire, quand F 3, alors utiliser P 1 conservera la taille originale de l'entrée. Quand F 5, P 2. Pour un général F, on peut voir que P $(F-1)/2$ préserve la taille d'entrée. Si vous devez utiliser des filtres de plus grande taille (par exemple 7x7), il n'est courant de le voir que sur le tout premier calque conv qui regarde l'image d'entrée.

Les couches de pooling sont chargées de sous-échantillonner les dimensions spatiales de l'entrée. Le réglage le plus courant est d'utiliser max-pooling avec 2x2 champs réceptifs (i.e. F 2), et avec une foulée de 2 (i.e. S 2). Notez que cela élimine exactement 75 % des activations dans un volume d'entrée (en raison d'un sous-échantillonnage de 2 à la fois en largeur et en hauteur).

Un autre réglage un peu moins courant est d'utiliser des champs réceptifs 3x3 avec une foulée de 2, mais cela fait. Il est très rare de voir des tailles de champ réceptif pour la mise en commun maximale qui sont supérieures à 3 parce que la mise en commun est alors trop perdue et agressive. Cela conduit généralement à des performances moins bonnes.

4.2.4 Entraînement

Dans cette section, la configuration de l'entraînement est affichée : la fonction de coût et le taux d'apprentissage.

Cross entropy

L'objectif de l'entrainement est d'ajuster les poids et les biais des valeurs sur les couches du réseau. Cela se fait en minimisant une fonction de coût. La fonction de coût est cross-entropy entre les étiquettes d'origine et la prédiction du modèle. Cette fonction est choisie parce que la dernière couche renvoie une distribution sur l'ensemble des 7 étiquettes. Cross-entropy est une mesure d'erreur spécifique pour les distributions de probabilités. Le fait que chaque image possède une et une seule étiquette (étiquettes mutuellement exclusives) a été exploitée pour utiliser l'entropie croisée. .

Un algorithme pour optimiser la fonction d'entropie croisée doit être défini. Celui utilisé est le momentum avec une valeur de 0,9, similaire à Krizhevsky et al[55]. L'étape suivante consiste à mettre à jour les poids et les biais en conséquence. Alors que la perte de cross-entropy est minimisée, il déplace les poids vers les gradients.

Taux d'apprentissage

Le taux d'apprentissage est un hyperparamètre à prendre en compte. Le taux d'apprentissage indique la vitesse de la recherche à travers l'espace de poids. Si le taux d'apprentissage est trop faible, la recherche ne sera pas trop exhaustive. D'autre part, un taux d'apprentissage plus

élevé permettra une meilleure recherche, mais cela peut conduire à un poids trop important .
Dans cette expérience, un taux d'apprentissage en décomposition a été mis en œuvre. Cela signifie que le taux d'apprentissage initial commencera à diminuer après quelques itérations. Pour effectuer ce calcul, trois hyperparamètres sont réglés :

1 Taux d'apprentissage initial

2 Facteur de décroissance

3 Épochs par désintégration

Le taux d'apprentissage initial a été fixé à 0,1. Cette valeur est une heuristique qui habituellement est performant sur les réseaux convolutifs de formation. Le facteur de désintégration indique que le proportion dans laquelle le taux d'apprentissage initial sera diminué. Enfin, les épochs par facteur de désintégration est le déclencheur de la diminution. Une époch, c'est le moment où tous les images définies ont été traitée à partir de la file d'attente. Cette valeur est fixée à 50 épochs. Le taux d'apprentissage et les valeurs du facteur de désintégration ont été calculés par essais et erreurs. L'époch par désintégration a été calculée en fonction de la taille de l'ensemble de données et de la taille du lot.

4.2.5 Expérimentation

L'évaluation a été faite en attribuant un point à chaque prédiction correcte sur un lot. Après avoir exécuté un nombre arbitraire de lots (10), la somme de toutes les prédictions correctes est divisée par le nombre total d'exemples. Deux types d'évaluation différents sont effectués : sur la classe top-1 et sur la classe top-3. Prédiction de la classe Top-1 signifie que l'étiquette n'est comparée qu'à la probabilité la plus élevée retournée par le la dernière couche. De la même manière, la prédiction de la classe Top-3 retourne vrai si l'étiquette est prédites sur les 3 probabilités les plus élevées.
Ensuite, un vecteur avec la taille du lot est renvoyée. Chaque élément est vrai ou faux, selon la prédiction. Tous les vrais éléments sont comptés. Et, finalement, la précision totale du modèle est calculée.
Chaque lot est généré au hasard à partir de la file d'attente des images. Cela signifie que la

précision peut changer en raison de la sélection de l'image. Afin de minimiser cet effet, les résultats de précision ont été présentés après avoir effectué l'évaluation 10 fois. De plus, la moyenne est également rapportée. Pendant l'évaluation, l'augmentation des données n'est pas appliquée sur les images. La seule opération d'image est celle du blanchiment. C'est la seule différence entre l'alimentation du réseau et l'étape de formation.

4.3 La phase d'évaluation

La deuxième phase évalue l'ensemble de données Japanese Female Facial Expression (JAFFE) sur le réseau formé au cours de la phase 1. Y a pas eu de pré-traitement sur cette data-set parce-que les images sont déjà pré-retraitées.

Teste

Le réseau formé sur sept émotions a été évalué par un sous-ensemble d'images JAFFE. Ce sous-ensemble d'images ne comprend les mémés émotions que KDFE représente En ce qui concerne la méthode d'évaluation, elle est similaire à celle utilisée pour la phase.

4.4 Conclusion

La solution proposée de notre problématique est présentée dans ce chapitre, en effet notre proposition est divisée en deux phases :

1 La phase 1 : Prétraitement et Entrainement : elle comprend :
- — Définition de la topologie du réseau.
- — Le prétraitement des images.
- — L'alimentation du réseau avec les images KDEF modifiées.
- — L'entrainement du réseau.
- — Les paramètres de réglage.
- — L'évaluation de la précision du réseau.

2 La phase 2 : d'évaluation

CHAPITRE 5

TESTES ET RÉSULTATS

Dans ce chapitre, les résultats expérimentaux seront présentés. Un ensemble de données de base est présenté afin de mieux comprendre la performance des expériences par rapport à d'autres travaux de recherches.

5.1 Base de données de référence

Dans cette section, un ensemble de données de référence est présenté. Le premier d'entre eux montre l'exactitude d'une proposition au hasard. Les lignes de base suivantes ont été extraites de l'Emotion Recognition In The Wild Challenge and Workshop (EmotiW)[34]. De plus, une recherche présentée dans la section Travaux connexes est présentée comme une base de référence à la fine pointe de la technologie.

Tous les chiffres indiqués dans cette section renvoient à la précision de prédiction du modèle correspondant.

Comme nous l'avons déjà dit, la recherche suivante appartient à la catégorie de l'apprentissage supervisé. Le besoin d'un ensemble de données contenant des images d'émotions faciales et leur étiquette correspondante est crucial. Pour ce faire, deux ensembles de données ont été choisis pour réaliser l'expérience :

— Karolinska Directed Emotional Faces (KDEF)

— Japanese Female Facial Expression (JAFFE)

Une autre paire d'ensembles de données semblait également prometteuse au début. Cette paire est composée de l'ensemble de données EURECOM Kinect [19] et de l'ensemble de données Florence Superface[8]. Cependant, ils ont été rejetés parce que leur manque d'étiquettes ou de toute information qui pourrait mener à une génération automatique d'étiquettes.

5.2 Karolinska Directed Emotional Faces (KDEF)

particularités

Le Karolinska Directed Emotional Faces (KDEF) est un ensemble de 4900 photos d'expressions faciales humaines. L'ensemble d'images contient 70 individus affichant 7 expressions émotionnelles différentes. Chaque expression est vue sous 5 angles différents.

Sujets

Population : 70 acteurs amateurs, 35 femmes et 35 hommes. Critères de sélection : Âge compris entre 20 et 30 ans. Pas de barbe, moustaches, boucles d'oreilles ou lunettes, et de préfé-

rence pas de maquillage visible pendant la séance photo. **Méthode** Tous les sujets ont reçu des instructions écrites à l'avance. Ces instructions comprenaient une description des sept expressions différentes qu'ils devaient poser pendant la séance photo. On a demandé au sujet de répéter les différentes expressions pendant une heure avant de venir à la séance photo. Il a été souligné que le sujet devrait essayer d'évoquer l'émotion qui devait être exprimée et - tout en conservant une manière d'exprimer l'émotion qui leur semblait naturelle - essayer de rendre l'expression forte et claire.

Tous les sujets portaient des T-shirts gris spéciaux. Ils étaient assis à une distance d'environ trois mètres de la caméra. La distance absolue a été adaptée pour chaque sujet en ajustant la position de l'appareil photo jusqu'à ce que les yeux et la bouche du sujet se trouvent à des positions verticales et horizontales prédéfinies sur l'écran grille de l'appareil photo.

Les lumières étaient réglées de manière à projeter une lumière indirecte douce et uniforme des deux côtés du visage.

Après une séance de répétition, les sujets ont été tournés en une seule expression à l'époque jusqu'à ce que les sept expressions aient été tournées (première série). Les sujets ont été le plan une fois de plus dans toutes les expressions et les angles (série deux).

Équipement

— Caméra : Pentax LX.
— Lentille : Pentax original 135 mm.
— Extra : Grille de grille.
— Film : Kodak 320 T.
— Lumières : 3 * 500 W lampes.

Numérisation

— Matériel : Macintosh 8500/120, Polaroid Sprintscan 35.

— Logiciels : Adobe Photoshop 4.

— Réglages : Les positifs (36 * 24 mm) ont été scannés en couleur RGB, avec une résolution de 625 dpi.

Extra : Chaque image a été ajustée à une grille numérique. Une fois de plus, la position verticale et horizontale des yeux et de la bouche a été ajustée à des positions spécifiques sur la grille, puis recadrée à une taille de 562 pixels de largeur et 762 pixels de hauteur.

Détails

— Participants : 70 (35 hommes et 35 femmes).

— Âge : m 25 ans, allant de 20 à 30 ans.

— Expressions : 7 (neutre, heureux, en colère, effrayé, dégoûté, triste, surpris).

— Angles : 5 (profil plein gauche, demi profil gauche, droit, demi profil droit, demi profil droit, profil plein droit, profil plein droit).

Sessions : 2.

— Nombre de photos : 4900.

— Taille : 562 * 762 pixels.

— Résolution : 72*72 dpi.

— Couleurs : 16,7 millions (32 bits).

— Taille gonflée : 1.6 Mo.

— Taille compressée : environ 122 kb (de 85 à 158 kb).

— Format de fichier : JPEG.

— Qualité de compression : 94

Codes : Example : AF01ANFL.JPG

— Letter 1 : Session

  — A = series one

  — B = series two

— Letter 2 : Gender

  — F = female

  — M = male

— Letter 3 & 4 : Identity number 01 - 35

- Letter 5 & 6 : Expression
  - AF = afraid
  - AN = angry
  - DI = disgusted
  - HA = happy
  - NE = neutral
  - SA = sad
  - SU = surprised
- Letter 7 & 8 : Angle
  - FL = full left profile
  - HL = half left profile
  - S = straight
  - HR = half right profile
  - FR = full right profile
- Extension : Picture format
- JPG = jpeg (Joint Photographic Experts Group)

5.3 Japanese Female Facial Expression (JAFFE)

La base de données contient 213 images de 7 expressions faciales (6 expressions faciales de base + 1 neutre) [?] posées par 10 modèles féminins japonais. Chaque image a été évaluée sur 6 adjectifs émotionnels par 60 sujets japonais. La base de données a été planifiée et assemblée par Michael Lyons, Miyuki Kamachi et Jiro Gyoba. Les photos ont été prises au département de psychologie de l'Université de Kyushu.

5.3.1 Deviner au hasard

Une estimation aléatoire calculée sur les 7 étiquettes fournies par KDEF donne une prédiction avec une performance de **14,27 %**. Ce résultat est équivalent au calcul d'une probabilité uniforme pour chaque étiquette.

FIGURE 5.1: exemples d'images de la base de données KDFE

Comme il a été mentionné précédemment, la base de référence pour **7** étiquettes est de **16,67 %**. L'étiquette la plus courante en KDEF est la surprise avec 83 séquences, si on attribuait toujours cette étiquette, la performance de prédiction serait de **25,38%**. La même approche pour les 3 étiquettes les plus courantes (surprise, joie et dégoût) aurait permis d'obtenir **64,5 %** de précision de prédiction.

L'un des derniers travaux sur la détection des émotions faciales correspond à Ramachandran et al [67]. Ses recherches ont atteint **97%** de la précision de prédiction par rapport à KDEF. Il est bon de souligner que ce travail a employé un mélange d'ANN avec l'ingénierie des caractéristiques.

FIGURE 5.2: exemples d'images de la base de données JAAFE

5.4 Choix techniques

Hardware

Avan de mener notre expérimentation et l'évaluation, nous avons utilisé un PC marque DELL Inspiron, équipé d'un processeur multi-core I3, cadencé par une horloge d'une fréquence de 2.40GHZ, avec 4 GO Octets de RAM, un disque dur d'une capacité de 400 Giga Octets.

5.5 Software

De nos jours, de nombreux frameworks ont été développés pour l'apprentissage approfondi. Certains des plus populaires sont les bibliothèques telles que : Caffe, Keras et TensorFlow. L'utilisation de Python en tant qu'élément sur tous ces frameworks montre qu'il s'agit du langage préféré de l'API front-end pour l'apprentissage machine. Habituellement, Python est combiné avec un langage de programmation qui fournit un soutien pour les opérations de bas niveau telles que : JAVA ou C++, pour agir sur back end.
Dans cette section nous présenterons le choix technique effectués pour réaliser notre système

5.5.1 Qu'est-ce que Python ?

Python est un langage de programmation informatique " batteries incluses ". Plus concrètement, Python est un langage de programmation qui, contrairement à d'autres langages de pro-

grammation tels que C, Fortran ou Java, permet aux utilisateurs de se concentrer et de résoudre plus facilement les problèmes de domaine au lieu de faire face à la complexité du fonctionnement d'un ordinateur. Python atteint ce but en ayant les attributs suivants :

— Python est un langage de **haut niveau**, ce qui signifie qu'il résume les détails techniques liés à l'informatique. Par exemple, Python ne fait pas trop réfléchir ses utilisateurs à la gestion de la mémoire de l'ordinateur ou à la déclaration correcte des variables et utilise des hypothèses sûres sur ce que le programmeur essaie de transmettre. De plus, un langage de haut niveau peut être exprimé d'une manière plus proche de la prose anglaise ou des équations mathématiques. Python est parfait pour la programmation alphabétisée en raison de sa légèreté et de sa nature de "cérémonie basse".

— Python est un langage **à usage général**, ce qui signifie qu'il peut être utilisé pour tous les problèmes dont un ordinateur est capable, plutôt que de se spécialiser dans un domaine spécifique comme l'analyse statistique. Par exemple, Python peut être utilisé à la fois pour l'intelligence artificielle et l'analyse statistique. Python peut être utilisé pour une variété de tâches hétérogènes dans un flux de travail donné,

— Python est un langage **interprété**, ce qui signifie que l'évaluation du code pour obtenir des résultats peut se faire immédiatement plutôt que d'avoir à passer par un cycle de compilation et d'exécution qui prend beaucoup de temps, ce qui accélère les processus de réflexion et d'expérimentation. IPython est une forme interactive du langage Python également inventé par Fernando Pérez. Ces environnements excellent pour le prototypage rapide de code ou l'expérimentation rapide et simple avec de nouvelles idées.

— Python dispose d'une bibliothèque standard et de nombreuses bibliothèques tierces donnant accès à une vaste gamme de bases de code existantes et d'exemples de résolution de problèmes.

— Python a beaucoup d'utilisateurs, ce qui signifie que les programmeurs peuvent rapidement trouver des solutions et des exemples de code aux problèmes avec l'aide de Google et Stackoverflow.

Ces fonctionnalités, peut-être, viennent avec un coût mineur de réduction des performances linguistiques, mais c'est un compromis que la grande majorité des utilisateurs sont prêts à faire afin d'obtenir tous les avantages que Python a à offrir.

5.5.2 Framework utilisés

Caffe[47] a commencé comme un projet de doctorat par Yangqing Jia alors qu'il travaillait au Berkeley Vision and Learning Center (BVLC). Aujourd'hui, Caffe est un projet maintenu par BVLC, et a acquis de nombreux contributeurs de sa communauté grandissante. Il a été écrit en Python et en C+++ ; et il supporte les principales plates-formes : Linux, OSX et Windows. Caffe a été principalement conçu pour effectuer des calculs de vision par ordinateur.
Keras est une bibliothèque réseau de neurones à source ouverte écrite en Python. Il est capable de fonctionner sur TensorFlow, Microsoft Cognitive Toolkit, Theano ou MXNet. Conçu pour permettre une expérimentation rapide avec des réseaux neuronaux profonds, il se concentre sur la convivialité, la modularité et l'extensibilité. Il a été développé dans le cadre de l'effort de recherche du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), et son auteur et mainteneur principal est François Chollet, un ingénieur de Google.

En 2017, l'équipe TensorFlow de Google a décidé de prendre en charge Keras dans la bibliothèque centrale de TensorFlow. Chollet a expliqué que Keras a été conçu pour être une interface plutôt qu'un cadre autonome d'apprentissage machine. Il offre un ensemble d'abstractions de plus haut niveau, plus intuitif, qui facilite le développement de modèles d'apprentissage en profondeur, quel que soit le backend informatique utilisé[?]. Microsoft a également ajouté un backend

CNTK à Keras, disponible à partir de CNTK v2.0.
Keras contient de nombreuses implémentations de blocs de construction de réseaux de neurones couramment utilisés tels que des couches, des objectifs, des fonctions d'activation, des optimiseurs et une foule d'outils pour faciliter le travail avec les données d'image et de texte.

Keras permet aux utilisateurs de produire des modèles profonds sur les smartphones (iOS et Android), sur le web, ou sur la machine virtuelle Java.

Il permet également l'utilisation de la formation distribuée de modèles d'apprentissage profond sur des clusters d'unités de traitement graphique (GPU).
Tensorflow TensorFlow (TF)[2] est une bibliothèque logicielle open source pour l'apprentissage machine écrite en Python et en C++ . Sa sortie il y a 2 ans (15 novembre 2015) a fait l'objet d'une forte couverture médiatique. La raison principale est que TF a été développé par Google Brain Team. Google a déjà utilisé TF pour améliorer certaines tâches sur plusieurs produits. Ces tâches comprennent la reconnaissance vocale dans Google Now, les fonctions de recherche dans Google Photos et la fonction de réponse intelligente dans Inbox by Gmail.
Certaines décisions en matière de conception dans TF ont conduit à l'adoption rapide de ce cadre par le une grande communauté. L'une d'entre elles est la facilité de passer du prototype à la production. Il n'est pas nécessaire de compiler ou de modifier le code pour l'utiliser sur un produit. Ensuite, le cadre n'est pas seulement considéré comme un outil de recherche, mais aussi comme un outil de production. Un autre aspect principal de la conception est qu'il n'est pas nécessaire d'utiliser des API différentes lorsque l'on travaille sur CPU ou GPU. De plus, les calculs peuvent être déployés sur des postes de travail, des serveurs, des serveurs et des appareils mobiles.

Un élément clé de la bibliothèque est le graphique de flux de données. Le sens d'exprimer des calculs mathématiques avec des noeuds et des bords est une marque de commerce de TF. Les nœuds sont généralement les opérations mathématiques, tandis que les bords définissent l'association entrée/sortie entre les nœuds. L'information voyage autour du graphique comme un tenseur, un tableau multidimensionnel. Enfin, les nœuds sont alloués sur des dispositifs où ils sont exécutés

en asynchrone ou en parallèle lorsque toutes les ressources sont prêtes.

5.5.3 Pourquoi Tensorflow ?

Alors que Caffe et Keras semblaient des frameworks appropriés pour réaliser ce projet, en fin de compte, celui qui a été choisi était TF r0.7. TF a été choisi pour deux raisons principales :

— **La Première est que** TF a le soutien de Google. Le fait que des millions de personnes ont utilisé des produits utilisant TF en arrière-plan signifie que le framework a été correctement testé. De plus, Google dispose d'un grand nombre de ressources pour continuer à travailler sur le framework, et pour fournir de la documentation et des ressources d'apprentissage.

— **Une autre raison** est que TF a bénéficié de l'expérience de développement autour d'autres frameworks, en particulier Theano. Ainsi, la portée du framework n'est pas seulement limitée à la recherche et au développement, mais aussi au déploiement.
Le support de Google a positionné TF comme l'une des principales bibliothèques pour l'apprentissage machine. dans un laps de temps relativement court depuis sa sortie en novembre 2015. Google s'engage à un développement à long terme du cadre en l'utilisant sur ses propres produits. Par exemple, Google DeepMind, qui sont les créateurs d'AlphaGo (AlphaGo est un ordinateur qui a été capable de battre pour la première fois un joueur de Go professionnel humain), a décidé de faire passer tous leurs projets d'un cadre nommé Torch7[22] à TF. De plus, une version distribuée de TF est sortie en avril 2016. Tous ces signaux sont des signaux que Google pousse TF à devenir son principal outil de recherche sur l'apprentissage machine. Quand il s'agit à la documentation, la page web de TF offre une explication détaillée de l'ensemble de Python. API, et pour toutes les versions majeures à ce jour. En outre, un cours en ligne ouvert à tous sur Deep Learning enseigné par Google a été publié juste après quelques mois après TF de lâcher. L'instructeur est Vincent Vanhoucke, qui est chercheur scientifique principal. sur Google. Son travail est lié à l'équipe Google Brain, et TF est l'outil utilisé pour terminer les devoirs du cours.

— Une autre raison de choisir TF est que le cadre englobe une maturité élevée. malgré le peu de temps écoulé depuis sa sortie. Le fait que beaucoup de développeurs de TF ont déjà été impliqués dans d'autres projets tels que Theano et Torch7 est réellement pertinentes. TF a bénéficié de l'expérience en matière TF est en mesure de corriger de nombreux problèmes trouvés aux premières étapes d'autres frameworks depuis sa conception initiale. En conséquence, TF a atteint l'état de l'art de la performance sans compromettant la lisibilité du code. En outre, la flexibilité de définir différentes opérations ; en particulier les topologies des réseaux de neurones conduit à un prototypage rapide. La flexibilité n'est pas uniquement lié à la composition du réseau ou à la définition des opérations, mais le déploiement de calcul plates-formes. L'API de TF est la même, même lorsque les calculs sont exécutés sur CPU ou GPU dans un ordinateur de bureau, un serveur ou un appareil mobile.

5.5.4 Résultats de la phase de prétraitement et d'entrainement

Dans le tableau de la figure 5.3, un résumé des paramètres est présenté. Les expériences de cette phase sont menées selon cette configuration, sauf indication contraire explicite.

Perte de réseau et taux d'apprentissage

Le tableau des pertes présenté à la figure 5.4 montre comment la fonction de coût est minimisée au cours des étapes de formation.
Pendant les **450** premières étapes, la perte est réduite en douceur. Après ce point, la courbe commence à converger. A l'étape **800**, la valeur de perte est de **0,79** ; à l'étape 900, la valeur de perte est de **0,77**. C'est juste une réduction de **0,02** sur une centaine d'étapes. Compte tenie qu'une formatu de ce comportement, on peut dire que le modèle a convergé.

Cela signifie qu'un entrainement plus longue ne réduira pas substantiellement la perte du réseau. Par exemple, l'utilisation d'un taux d'apprentissage plus faible (0,01) fournit une courbe différente, comme on peut le voir à la figure 5.5. Le modèle n'a pas encore convergé, un temps

Parametres	Valeur
Batch size	64
Output channels	64
Kernel size	3 x 3
Kernel stride	1
Padding	same
Pooling	Max Pooling
Pool size	3 x 3
Pool stride	2
FC1 neurons	385
FC2 neurons	192
Dropout	0,99
Activation function	ReLU
Number of classes	7
Cost function	Cross entropy
Optimizer	Momentum
Momentum	0,9
Leaning rate	0,1

FIGURE 5.3: Sommaire de configuration pour la première phase d'expérimentation

de formation plus long est donc nécessaire. Bien qu'une formation plus longue le temps pourrait améliorer l'exactitude du réseau ; exposer un réseau à une formation prolongée. finira en surdimensionnement. Le temps moyen de formation était de l'ordre de 10 minutes et de 37 secondes.

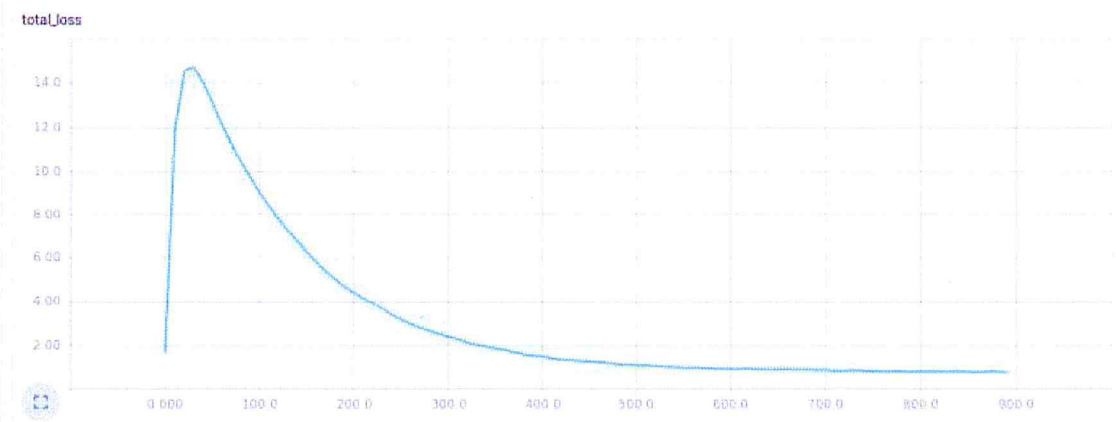


FIGURE 5.4: Perte totale sur 900 étapes de formation avec un taux d'apprentissage de 0.1

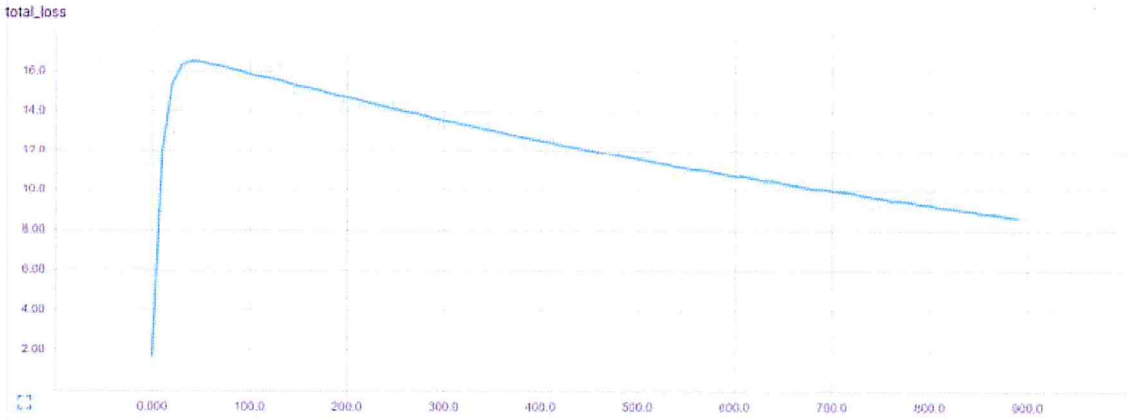


FIGURE 5.5: Perte totale sur 900 étapes de formation avec un taux d'apprentissage de 0.01

5.5.5 Précision de la classification

La précision du réseau sur le poste d'entraînement est indiquée dans la figure 5.6. Les résultats rapportés correspondent au classement du Top-1 et du Top-3. Sur les prédictions Top-1, la plus grande précision était de **72.5%**, et la moyenne est de **70.54%**. Au contraire, pour les prédictions du Top-3, la plus grande précision était de **80%** ; avec une moyenne de **77,9%**. Table dans la figure 5.7 montre que les résultats utilisant la configuration sur table dans la figure 5.3 ont donné de meilleurs résultats que les lignes de base de devinettes aléatoires.

	Run									
Top	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	70.0	71.1	72.5	70.8	68.1	71.9	71.1	71.3	68.4	70.2
3	75.8	78.4	77.8	77.7	80.0	77.8	78.7	76.4	79.1	77.3

FIGURE 5.6: Précision de la classification du réseau sur 7 étiquettes d'émotions sur KDEF dataset pour un taux d'apprentissage fixé à 0,1.

5.5.6 L'effet de dropout

Le figure 5.8 montre la précision du réseau lorsque le décrochage est réglé à **0,5**. Comme on peut le constater, la précision de la prédiction Top-1 est réduite par rapport aux résultats

présentés dans la figure 5.6. La moyenne de précision est de **57,24 %**. Il est d'environ **13 %** moins précis par rapport à la moyenne lorsque le taux de décrochage était fixé à **0,99**, **70,54 %**.
Alors que sur la prédiction Top-3, il n'y a pas d'effet de ce genre. Les résultats sont assez similaires à ceux affichés à la figure 5.6. La moyenne est de **77.5%**, alors que la moyenne initiale était de **77.9%**.

Model	Accuracy
6-label random guess	16.67
Most frequent label	25.38
EmotiW 2014's winner	39.35
EmotiW 2015's winner	44.70
3 most frequent labels	64.50
Top-1 prediction	70.54
Top-3 prediction	77.90
Ramachandran	97.00

FIGURE 5.7: Comparaison des résultats par rapport aux niveaux de référence proposés

	Run									
Top	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	56.2	57.0	58.8	58.1	58.9	55.6	57.3	57.7	55.9	56.9
3	75.9	76.9	78.9	77.2	78.6	77.5	78.7	78.0	77.7	76.1

FIGURE 5.8: Précision du réseau lorsque l'abandon est réglé sur 0,5.

5.5.7 Différents optimiseurs

En plus de momentum, quelques nouveaux optimiseurs ont été essayés : Adam Optimizer et FTRL Optimizer. En utilisant Adam Optimizer, le modèle converge déjà autour de la **100ème étape**, comme on peut le voir dans la figure 5.7 .
Cependant, les résultats présentés dans le tableau 6.5 montrent que la précision moyenne de prédiction est de **65,01 %** pour la prédiction Top-1 et de **75,95 %** pour la prédiction Top-1. sur la

	Adam Optimizer / Run									
Top	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	64.4	62.2	67.0	64.2	66.6	64.8	66.4	65.9	65.2	63.4
3	75.3	75.8	75.5	74.7	75.6	76.1	76.1	78.1	76.2	76.1

FIGURE 5.9: Précision de classification lors de l'utilisation de l'optimiseur d'Adam

prédiction du Top-3. Les deux valeurs sont légèrement inférieures à celles obtenues avec le momentum.

La figure 5.8 révèle que l'utilisation de l'optimiseur FTRL a provoqué une convergence précoce autour de la 30 ème étape. L'oscillation de cette étape jusqu'à la fin n'est pas lisse. En ce qui concerne l'exactitude de la classification, une valeur moyenne de **19,06 %** sur la prédiction Top-1 a été atteinte. Cette valeur se situe dans la plage des lignes de base des suppositions aléatoires. Cependant, la prédiction du Top-3 a atteint **79,02%** sur la prédiction Top-3, ce qui est légèrement meilleur que la prédiction Top-3 rapportée en utilisant le momentum.

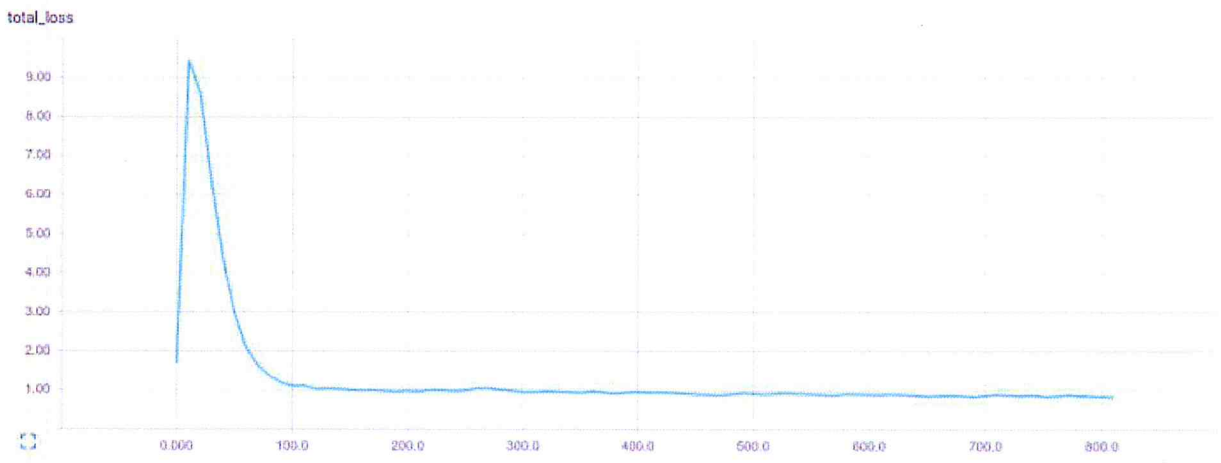


FIGURE 5.10: Perte totale sur 900 étapes d'entraînement à l'aide d'Adam Optimizer

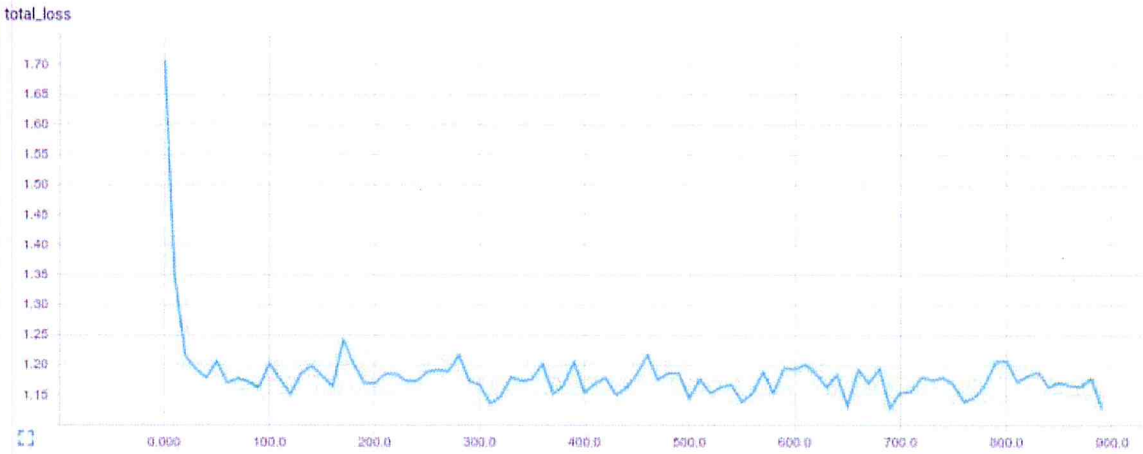


FIGURE 5.11: Perte totale sur 900 étapes d'entraînement à l'aide d'FTRL Optimizer

	FTRL Optimizer / Run									
Top	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	18.6	18.9	18.6	20.0	20.3	18.0	18.9	19.2	19.8	18.3
3	78.9	79.2	79.5	79.7	78.9	79.2	79.1	78.7	78.3	78.7

FIGURE 5.12: Précision de classification lors de l'utilisation de l'optimiseur FTRL

5.6 Résultats de la phase d'évaluation.

La deuxième phase consistait à évaluer l'exactitude du réseau à l'aide d'un ensemble de données JAFFE.

Le réseau a été formé en utilisant les paramètres affichés dans la figure 5.3 .

5.6.1 Précision de la classification

La précision du réseau sur l'ensemble JAFFE est indiquée dans le tableau de la figure 5.13

Les résultats rapportés correspondent au classement du Top-1 et du Top-3. Sur les prévisions Top-1, la précision la plus élevée était de **55,5 %**, et la moyenne s'établit comme suit **53.78%**.

Au contraire, pour les prévisions du Top-3, la précision la plus élevée était de **89,8 %**; avec une moyenne de **89,32 %**.

Le tableau dans la figure 5.14 montre que les résultats obtenus à l'aide de la configuration sur le tableau dans la figure 5.3 a obtenu de bien meilleurs résultats que les lignes de base des suppositions aléatoires. De plus, la prédiction du Top-3 était meilleure que celle rapportée sur KDEF. test de **Wilcoxon** a été exécuté sur les deux ensembles de résultats. La valeur p était inférieure à **0,001**, ce qui signifie que que la prédiction du Top-3 est encore nettement meilleure, même si la différence n'est pas aussi importante comme prévu.

Top	1	2	3	4	5	6	7	8	9	10
1	53.6	54.5	53.7	52.7	53.1	55.5	55.5	52.8	53.7	52.7
3	89.7	89.1	89.5	88.7	89.4	88.6	89.8	89.5	89.2	89.7

FIGURE 5.13: Précision du réseau grâce à l'ensemble de données JAFFE.

Model	Pressision
7-label random guess	16,67
Most frequent label	25,38
EmotiW 2014's winner	39,35
EmotiW 2015's winner	44,7
Top-1 prediction (JAFFE)	53,78
Top-1 prediction (KDEF)	70,53
Top-3 prediction (KDEF)	77,9
Top-3 prediction (JAFFE)	89,32
Ramachandran	97

FIGURE 5.14: Comparaison des résultats, y compris les résultats de la phase 2 par rapport aux de référence proposés.

5.7 Discussion

Dans cette section, les résultats sont discutés pour obtenir un aperçu de la façon dont différents paramètres affectent la précision de prédiction du réseau. Ceci illustre la complexité des différents composants qui articulent une expérience d'apprentissage machine.

5.7.1 Résultats de première phase : Pétraitement et Entrainemet

Cette section présente l'interprétation des résultats de la phase **Pétraitement et Entrainemet**

Précision de la classification

Lorsque l'on compare les résultats fournis dans la section **4.5** par rapport à nos travaux de base, on peut déterminer qu'il est bien meilleur qu'une supposition aléatoire pour sept étiquettes d'émotions et qu'il suffit d'attribuer l'étiquette la plus fréquente.
La comparaison avec **EmotiW** pourrait être délicate car différents corpus sont utilisés. Le gagnant de 2014 a fourni quelques idées pour améliorer la précision de sa classification. L'une d'entre elles consistait à essayer quelques catégories difficiles, tandis que l'autre était axée sur les catégories suivantes explorer une stratégie de fusion plus efficace.
Comme il est expliqué plus loin, l'utilisation de dropout a amélioré l'exactitude de nos recherches. L'utilisation de dropout n'est pas signalée. sur leurs expériences. Les différences entre les prévisions du Top-1 et du Top-3 ne sont pas les suivantes aussi grand qu'on aurait pu s'y attendre. La différence moyenne est de **7,36 %**.
L'utilisation de l'ingénierie des caractéristiques a une influence importante sur la précision de la classification. La recherche de **Kotsia et Pitas**[50] a atteint **99,7 %** sur la reconnaissance de l'expression faciale.
De plus, **Ramachandran** et al [67] ont obtenu résultats de classification autour de **97%**. Ces chiffres sont largement supérieurs à la méthode. présenté dans ce document. Cette preuve souligne le fait que DL a besoin d'un grand corpus. lorsqu'une expérience d'apprentissage supervisé est mise en place. Cependant, certaines stratégies ont été mises à l'essai pour augmenter la précision du réseau sur cet ensemble de données. En plus de dropout, un réseau plus petit et le redimensionnement des images de 640 x 480 pixels à 32 x 32 pixels a également contribué à l'amélioration de la précision de prédiction

Perte de réseau et taux d'apprentissage

Les figures 5.3 et 5.5 montrent l'importance du taux d'apprentissage lorsque l'on minimise la fonction de coût. De plus, il s'agit d'un bon moyen de déterminer la durée de la formation. Une durée de formation précise est importante pour éviter le surdimensionnement du réseau. Le surdimensionnement réduit la capacité du réseau à généraliser. Dans ce contexte, généraliser signifie prédire avec la même précision sur des exemples qui ne sont pas utilisés pendant la formation. D'un point de vue pratique, la réduction de la durée de la formation présente certains avantages. L'une d'entre elles est que les nouvelles idées peuvent être facilement essayées sur un réseau. Un autre avantage est l'économie de ressources. Cet impact peut être encore plus visible dans un grand centre de recherche où un temps de formation plus long implique moins de productivité sur plusieurs personnes. L'utilisation de GPU sur DL est une véritable preuve de la préoccupation de la communauté concernant la réduction du temps de formation.
Le taux d'apprentissage peut également entraîner une explosion des pertes. Cela se produit lorsque le taux d'apprentissage est réglé sur une valeur trop élevée pour le réseau. Au début de ce projet, l'explosion des pertes était un problème courant. Plusieurs valeurs de taux d'apprentissage ont été utilisées, mais elles n'ont fait que prolonger le processus. La solution à ce problème était de retourner la sortie de la couche softmax sous la forme d'une distribution de probabilité.
Enfin, la perte de réseau influence la précision de la classification. Cependant, il serait difficile de définir mathématiquement une équation qui relie les deux valeurs puisque la valeur de perte initiale est liée à la topologie du réseau.

L'effet de dropout

La variation de la valeur du dropout a eu une grande influence sur la précision de la prédiction top-1. Tel qu'indiqué à la section 4.5.7 , la réduction de la valeur de dropout à **0,5** a réduit l'exactitude de la classification de 13 %. Cependant, il n'a pas eu le même effet sur la prédiction des trois premiers.
Dropout est directement lié à la façon dont les neurones des couches entièrement connectées classifient les caractéristiques. Il semble que chaque neurone a été forcé d'apprendre des caractéristiques

pour chaque classe. Les activations de grosses unités ont été rejetées en raison de la probabilité de dropout élevée **(0,99)**. Lorsque cette valeur a été réduite à **0,5**, l'effet des caractéristiques qui étaient plus communes à toutes les catégories était plus fort. Ainsi, le réseau n'a pas été en mesure de classer correctement puisqu'il était biaisé par ces caractéristiques.

Différents optimiseurs

L'expérience de l'utilisation de momentum, d'Adam Optimizer et de FTRL a également permis de mieux comprendre le processus d'apprentissage. Momentum a un temps d'entraînement équilibré, un compromis de précision de classification De plus, il n'ajoute qu'un nouvel hyperparamètre : la valeur de momentum. Les meilleurs résultats de cette recherche pour la prédiction du Top-1 ont été obtenus en utilisant un momentum fixé à **0,9**.

Un autre optimiseur utilisé dans le cadre de cette recherche est celui d'Adam. Adam optimiseur convergé environ **400** pas plus tôt par rapport à momentum présenté dans les figures **??** et **??**. Cependant, il y a eu quelques pertes au niveau de l'exactitude de la classification. Une moyenne différence de **5,53 %** pour la prédiction Top-1 et de **1,95 %** pour la prédiction Top-3. Basé sur ces résultats, il semble que l'élan permet aux unités d'apprendre des caractéristiques plus spécifiques. qui leur permettent d'atteindre une meilleure précision. Bien que, un scénario où l'optimiste d'Adam Optimizer est plus bénéfique lorsqu'il n'y a pas assez de ressources pour une utilisation prolongée. le temps de formation et une certaine précision peuvent être perdus. L'un des avantages de cet optimiseur est qu'il permet de n'ajoute aucun hyperparamètre supplémentaire. L'utilisation de FTRL a donné deux résultats différents lorsqu'on l'applique au Top-1 et au Top-3. prédiction. Sa précision sur le Top-1 n'est meilleure que lorsqu'on la compare au hasard. La différence moyenne entre les deux était de **2,39 %**. Cependant, sa précision Top-3 a atteint **79,02 %**. Cette valeur de différence est légèrement meilleure que celle obtenue. en utilisant l'élan (**77,90 %**), **1,12 %**. Une explication possible est que FTRL est resté bloqué. sur un minimum local. En restant la plupart du temps dans cette région, comme on peut le voir en figure **??**, certaines caractéristiques peuvent être devenues vraiment influentes. Ensuite, une classe contenant ceci pourrait

toujours être le vainqueur de la prédiction Top-1.

5.7.2 Résultats de la phase d'évaluation

Précision de la classification

L'évaluation du réseau formé sur la phase 1 à l'aide de JAFFE a fourni des informations sur la façon dont la généralisation du réseau. La principale différence entre les deux ensembles de données est que JAFFE a été capturé dans la nature. Par conséquent, les visages ne sont pas toujours au centre de l'image, et les participants regardent devant la caméra. De plus, aucune révision manuelle des images n'a été effectuée. Certaines images peuvent ne pas être considérées comme le bonheur d'un être humain.
Comme on l'a déjà dit, une seule étiquette a été utilisée pour l'évaluation : le bonheur. l'émotion. La précision du Top-1 du réseau pour cette émotion particulière était de **53,78 %**. Le on s'attendait à une réduction de l'exactitude en raison des différences expliquées sur le précédent paragraphe. Cependant, la précision du Top-3 a atteint **89,32 %**. Cela signifie que le réseau a été en mesure d'apprendre des fonctionnalités pour les personnes souriantes ; même sur un ensemble de données différent. Wilcoxon a été exécuté sur les deux ensembles de résultats. La valeur p est inférieure à **0,001**, c'est-à-dire que la prédiction du Top-3 est encore nettement meilleure, même si la différence n'est pas aussi importante. comme prévu.

5.7.3 Travaux futurs

Il est possible d'améliorer encore l'exactitude et la généralisation du réseau grâce aux pratiques suivantes.

— La première est d'utiliser l'ensemble des données pendant l'exécution de la fonction l'optimisation. L'utilisation de l'optimisation par lots convient mieux aux ensembles de données de plus grande taille. Un autre est d'évaluer les émotions une par une.

Cela peut mener à la détection de quelles émotions sont plus difficiles à classer. Enfin, l'utilisation d'un ensemble de données plus important pour l'apprentissage bénéfique.

— Deuxièmement, en raison de contraintes de temps, il n'a pas été possible d'évaluer chaque émotion. De cette façon, il aurait été possible de détecter quelles émotions sont plus faciles à classer, ainsi que ceux qui sont plus difficiles. De plus, la pré-entraînement sur chaque émotion pourrait conduire à un meilleur apprentissage des fonctionnalités. Après ça, le réseau aurait pu recevoir cet apprentissage (apprentissage par transfert). Cela aurait pu améliorer la réduction de la formation, du temps, ainsi que la minimisation à un degré plus élevé de la fonction de coût.

— De plus, l'utilisation d'un ensemble de données plus important peut mener à une formation à plus grande échelle. La formation dans un plus grand et pour plus de temps améliore la précision du réseau. Une plus grande échelle de formation permet au réseau d'apprendre des fonctionnalités plus pertinentes. Si ce n'est pas le cas, la fonction l'ingénierie est toujours nécessaire pour cette tâche.

— Enfin, l'utilisation d'un ensemble complet de données pour la formation, la préformation sur chaque émotion et l'utilisation d'un ensemble de données plus important semble avoir la possibilité d'améliorer les performances du réseau. Elles devraient donc être abordées dans les recherches futures sur ce sujet.

5.8 Conclusion

Au cours de ce dernier chapitre, nous avons réussi à tester notre prposition sur des ensembles de données standards, en effet deux ensembles de données ont été choisis pour réaliser les tests :

— Karolinska Directed Emotional Faces (KDEF)

— Japanese Female Facial Expression (JAFFE)

Nous avons aussi réussi à tester les différentes couches suivant l'architecture décrite dans chapitre précédent.

CONCLUSIONS

Le domaine des travaux de notre mémoire se positionne à l'intersection de l'analyse des sentiments et de l'apprentissage approfondi. En effet, Dans ce projet, une recherche visant à classer les émotions faciales par rapport aux images faciales statiques à l'aide de techniques d'apprentissage approfondi qui ont été mises au point. Il s'agit d'un problème complexe qui a déjà a été approché plusieurs fois avec différentes techniques. Bien que de bons résultats ont été obtenus en utilisant le deep learning.

Aussi, dans la première partie de ce travail, nous avons focalisé notre attention sur des sources bibliographiques théoriques issues des sciences : l'analyse des sentiments et l'apprentissage approfondi qui nous ont permis, dans un premier temps, de se rendre compte de la difficulté quant à la modélisation et la conception d'une solution à base de deep learning et, dans un deuxième temps, de cerner ensemble des travaux de recherches dans notre contexte.

Dans la deuxième partie de ce mémoire, nous avons proposé une solution à base de deep learning qui répond à notre problématique de départ et qui prend en compte les différents constats issus de nos études. Nous avons ainsi présenté et introduit une nouvelle architecture qui englobe la dimension 'Détection des émotions'. En conséquence, de nouvelles couches ont été introduites notamment : Prétraitement des images, Augmentation des données, et la couche réseau qu'est

composée de :

- Couche convolutionnelle.
- Opération de MaxPooling.
- couche entièrement connectée.
- Une couche softmax.

.

Bien que les résultats obtenus n'étaient pas à la fine pointe de la technologie, ils étaient légèrement meilleurs. que d'autres techniques, y compris machine learning. Cela signifie qu'éventuellement le prétraitement de l'image donne un élan au traitement de l'image. l'exactitude de la classification. Par conséquent, il réduit le bruit sur les données d'entrée.

De nos jours, les logiciels de détection des émotions faciales incluent l'utilisation de l'ingénierie des caractéristiques. Une solution totalement basée sur l'apprentissage des fonctionnalités ne semble pas encore très proche parce que d'une limitation majeure : l'absence d'un vaste ensemble de données sur les émotions. Par exemple, Le concours ImageNet utilise un ensemble de données contenant 14 197 122 images. En ayant un plus grand les réseaux ayant une plus grande capacité d'apprentissage pourraient être mis en œuvre. Ainsi, la classification des émotions est réalisée au moyen de techniques d'apprentissage profond.

Problématique

La difficulté de notre projet consiste à dévoiler les secrets de l'analyse des sentiments en adoptant une approche d'apprentissage automatique dans les images.

La problématique principale de ce projet consiste à trouver une technique pour former un réseau de neurones profonds avec des images étiquetées d'émotions faciales statiques. Plus tard, ce réseau pourrait être utilisé dans le cadre d'une partie d'un logiciel pour détecter les émotions des personnes.

La technique proposée dans ce projet est multidisciplinaire impliquant l'analyse des entiments et l'apprentissage profond. Apprendre comment ces différents domaines sont liés et comprendre comment ils peuvent apporter des solutions à des problèmes complexes est l'objectif de ce projet.

5.9 Travaux futurs

Il est possible d'améliorer encore l'exactitude et la généralisation du réseau grâce aux pratiques suivantes.

— La première est d'utiliser l'ensemble des données pendant l'exécution de la fonction l'optimisation. L'utilisation de l'optimisation par lots convient mieux aux ensembles de données de plus grande taille. Un autre est d'évaluer les émotions une par une. Cela peut mener à la détection de quelles émotions sont plus difficiles à classer. Enfin, l'utilisation d'un ensemble de données plus important pour l'apprentissage bénéfique.

— Deuxièmement, en raison de contraintes de temps, il n'a pas été possible d'évaluer chaque émotion. De cette façon, il aurait été possible de détecter quelles émotions sont plus faciles à classer, ainsi que ceux qui sont plus difficiles. De plus, la pré-entraînement sur chaque émotion pourrait conduire à un meilleur apprentissage des fonctionnalités. Après ça, le réseau aurait pu recevoir cet apprentissage (apprentissage par transfert). Cela aurait pu améliorer la réduction de la formation, du temps, ainsi que la minimisation à un degré plus élevé de la fonction de coût.

— De plus, l'utilisation d'un ensemble de données plus important peut mener à une formation à plus grande échelle. La formation dans un plus grand et pour plus de temps améliore la précision du réseau. Une plus grande échelle de formation permet au réseau d'apprendre des fonctionnalités plus pertinentes. Si ce n'est pas le cas, la fonction l'ingénierie est toujours nécessaire pour cette tâche.

— Enfin, l'utilisation d'un ensemble complet de données pour la formation, la préformation sur chaque émotion et l'utilisation d'un ensemble de données plus important semble avoir la possibilité d'améliorer les performances du réseau. Elles devraient

donc être abordées dans les recherches futures sur ce sujet.

BIBLIOGRAPHIE

[1] Keith Adams. Multi-GPU Training of ConvNets. pages 10–13, 2012.

[2] Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Man, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Vi, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow : Large-Scale Machine Learning on Heterogeneous Distributed Systems. 2015.

[3] A C C Cyfronet Agh. Towards co-evolution of fitness predictors and Deep Neural Networks.

[4] Rami Al-rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Justin Bayer, Anatoly Belikov, Alexander Belopolsky, Yoshua Bengio, Arnaud Bergeron, James Bergstra, Valentin Bisson, Josh Bleecher Snyder, Nicolas Bouchard, Nicolas Boulanger-lewandowski, Xavier Bouthillier, Alexandre De Br, Olivier Breuleux, Pierre-luc Carrier, Kyunghyun Cho, Jan Chorowski, Paul Christiano, Tim Cooijmans, Marc-alexandre C^, Yann N Dauphin, Olivier Delalleau, Julien Demouth, Guillaume Desjardins, Sander Dieleman, Laurent Dinh, Mathieu Germain, Xavier Glorot, Ian Goodfellow, Matt Graham, Caglar Gulcehre, Sean Lee, Simon Lefrancois, Simon Lemieux, L Nicholas, Zhouhan

Lin, Jesse A Livezey, Cory Lorenz, Jeremiah Lowin, Qianli Ma, Pierre-antoine Manzagol, Olivier Mastropietro, Robert T Mcgibbon, Roland Memisevic, Bart Van Merri, Joseph Turian, Sebastian Urban, Pascal Vincent, Francesco Visin, Harm De Vries, and David Warde-farley. Theano : A Python framework for fast computation of mathematical expressions. pages 1–19.

[5] N E C Labs America and Princeton Nj. Large-Scale Machine Learning with Stochastic Gradient Descent.

[6] V Ariables. A D ISTRIBUTION A DAPTIVE F RAMEWORK FOR P RE -. pages 1–8, 2016.

[7] Jochanan Benbassat and Reuben Baumal. What Is Empathy, and How Can It Be Promoted during Clinical Clerkships ? pages 832–839, 2004.

[8] Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. Superfaces : A Super-Resolution Model for 3D Faces. pages 73–82, 2012.

[9] P E Blöchl. Projector augmented-wave method. *Phys. Rev. B*, 50(24) :17953–17979, 1994.

[10] J L Castro and I Requena. Are Artificial Neural Networks Black Boxes ? 8(5) :1156–1164, 1997.

[11] Ken Chatfield, Karen Simonyan, and Andrew Zisserman. Return of the Devil in the Details : Delving Deep into Convolutional Nets CNN-based Methods. page 2013, 2013.

[12] Yutian Chen, Matthew W Hoffman, Misha Denil, and Timothy P Lillicrap. Learning to Learn without Gradient Descent by Gradient Descent. 2017.

[13] Thomas Cluzeau. Mathématiques pour l ' Ingénieur.

[14] Adam Coates, Brody Huval, Tao Wang, David J Wu, and Andrew Y Ng. Deep learning with COTS HPC systems. 2012.

[15] Neo Cognitron. UNIT - V.

[16] Jeffrey F Cohn. Observer-Based Measurement of Facial Expression With the Facial Action Coding System. pages 203–221, 2005.

[17] Arthur Crenn, Alexandre Meyer, Rizwan Ahmed Khan, Hubert Konik, Arthur Crenn, Alexandre Meyer, Rizwan Ahmed Khan, Hubert Konik, Saïda Bouakaz, Arthur Crenn, Université De Lyon, Alexandre Meyer, and Hubert Konik. Toward an Efficient Body Expression

Recognition Based on the Synthesis of a Neutral Movement To cite this version : HAL Id : hal-01675222 Toward an Efficient Body Expression Recognition Based on the Synthesis of a Neutral Movement. 2018.

[18] Antonio R Damasio. *n r F motion FD t , e e.*

[19] Abhinav Dhall, Roland Goecke, Jyoti Joshi, and Tom Gedeon. Emotion Recognition In The Wild Challenge 2014 : Baseline , Data and Protocol Categories and Subject Descriptors. 2014.

[20] Abhinav Dhall, Student Member, and Roland Goecke. Collecting Large , Richly Annotated Facial-Expression Databases from Movies. 6(1) :1–14, 2007.

[21] S A N Diego. 862 18 120,. (V), 1985.

[22] Weiguang Ding, Ruoyan Wang, and Graham Taylor. T - l -s v r - m gpu. pages 3–7, 2015.

[23] Fadi Dornaika and Franck Davoine. Simultaneous Facial Action Tracking and Expression Recognition in the Presence of Head Motion. pages 257–281, 2008.

[24] S L Dudarev, G A Botton, S Y Savrasov, C J Humphreys, and A P Sutton. Electron-energy-loss spectra and the structural stability of nickel oxide : An LSDA+U study. *Phys. Rev. B*, 57(3) :1505–1509, jan 1998.

[25] Third Edition. *Neural Networks and.*

[26] Paul Ekman. *Emotions Revealed.*

[27] Paul Ekman and Erika L Rosenberg. *What the Face Reveals :.*

[28] Andries P Engelbercht. 29.pdf.

[29] Irfan Aziz Essa. Analysis , Interpretation and Synthesis of Facial Expressions. (1988), 1995.

[30] Lixin Fan. Revisit Fuzzy Neural Network : Demystifying Batch Normalization and ReLU with Generalized Hamming Network. (Nips) :1–10, 2017.

[31] Kunihiko Fukushima. Biological Cybernetics. 202, 1980.

[32] Deepak Ghimire and Joonwhoan Lee. Geometric Feature-Based Facial Expression Recognition in Image Sequences Using Multi-Class AdaBoost and Support Vector Machines. pages 7714–7734, 2013.

[33] Xavier Glorot and Antoine Bordes. Deep Sparse Rectifier Neural Networks. 15 :315–323, 2011.

[34] Roland Goecke, Jyoti Joshi, Jesse Hoey, and Tom Gedeon. From Individual to Group-Level Emotion Recognition :. pages 524–528, 2016.

[35] Anna Gorbenko. On Face Detection from. 6(96) :4763–4766, 2012.

[36] Kunal Gupta. Wearable Tools for Affective Remote Collaboration for Affective Remote Collaboration.

[37] Gulam Hasan, Ashfaqul Hasan, Kulbeer Kaur, Muzaffar Ahmad, and Mohd Shafi. Student ' s Page The Facial Nerve : The Anatomical and Surgical important. 12(1) :53–57, 2005.

[38] D. 0. HEBB. The Organization of Behavior. 1949.

[39] G.a Henkelman, A.b Arnaldsson, and H.b c Jonsson. A fast and robust algorithm for Bader decomposition of charge density. *Computational Materials Science*, 36(3) :354–360, 2006.

[40] G E Hinton, N Srivastava, A Krizhevsky, I Sutskever, and R R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors arXiv : 1207 . 0580v1 [cs . NE] 3 Jul 2012. pages 1–18.

[41] Geoffrey Hinton. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. 15 :1929–1958, 2014.

[42] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep Neural Networks for Acoustic Modeling in Speech Recognition. pages 1–27.

[43] Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. (3).

[44] Minh Hoai. MULTI-SCALE REGION CANDIDATE COMBINATION FOR ACTION RECOGNITION Zhichen Zhao , Huimin Ma , Xiaozhi Chen. pages 2–6.

[45] No Harm Intended and Marvin L Minsky. No Harm Intended. (1969) :1–9, 1988.

[46] Dr.Ban I.S. The Face : Lec [1] Skin of the face : Muscles of the face :.

[47] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, Trevor Darrell, and U C Berkeley Eecs. Caffe : Convolutional Architecture for Fast Feature Embedding .

[48] philip N Johnson-Laird. The interaction between reasoning and decision making : an introduction. 49 :1–9, 1993.

[49] Takeo Kanade and Jeffrey F Cohn. Comprehensive Database for Facial Expression Analysis The Robotics Institute. (March), 2000.

[50] Irene Kotsia, Ioan Buciu, and Ioannis Pitas. An analysis of facial expression recognition under partial facial image occlusion. 26 :1052–1067, 2008.

[51] G Kresse and J Furthmüller. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Computational Materials Science*, 6(1) :15–50, 1996.

[52] G Kresse and J Hafner. \textit{Ab initio} molecular dynamics for liquid metals. *Phys. Rev. B*, 47(1) :558–561, jan 1993.

[53] David Kriesel. Neural Networks.

[54] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. 2009.

[55] Alex Krizhevsky and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. pages 1–9.

[56] Deep Learning, F O R Detecting, Space-time Action Tubes, Suman Saha, Gurkirt Singh, Oxford Oxford, U K Michael Sapienza, Philip H S Torr, and Fabio Cuzzolin. Deep Learning for Detecting Multiple Space-Time Action Tubes in Videos. pages 1–13, 2015.

[57] Yuqing Li. Deep Learning of Human Emotion Recognition in Videos.

[58] 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. pages 494–501, 2014.

[59] Andrew L Maas and Andrew Y Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models. 28, 2013.

[60] Yafeng Niu, Dongsheng Zou, Yadong Niu, Zhongshi He, and Hua Tan. A breakthrough in Speech emotion recognition using Deep Retinal Convolution Neural Networks. pages 1–7.

[61] Bulletin Of and Mathematical Biophysics. IDEAS IMMANENT IN NERVOUS ACTIVITY. 5 :115–133, 1943.

[62] Seyed A L I Osia, A L I Shahin Shamsabadi, A L I Taheri, Kleomenis Katevas, Sina Sajadmanesh, Hamid R Rabiee, Nicholas D Lane, and Hamed Haddadi. Mobile Analytics. 1(1), 2018.

[63] Omkar Moreshwar Parkhi. Features And Methods for Improving Large Scale Face Recognition. 2015.

[64] Louis A Penner, John F Dovidio, Jane A Piliavin, and David A Schroeder. P ROSOCIAL B EHAVIOR : Multilevel Perspectives. 2005.

[65] R W Picard. Affective Computing. (321).

[66] R W Picard. that recognize. 39 :705–719, 2000.

[67] Vedantham Ramachandran and E Srinivasa Reddy. Facial expression recognition with enhanced feature extraction using PSO & EBPNN. 11(10) :6911–6915, 2016.

[68] Martin Riedmiller. A Direct Adaptive Method for Faster Backpropagation Learning : The RPROP Algorithm.

[69] F Rosenblatt and Contract Nonr. THE PERCEPTRON : A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION. 65(6) :386–408.

[70] Nicu Sebe, Michael S Lew, Ira Cohen, Yafei Sun, Theo Gevers, and Thomas S Huang. Authentic Facial Expression Analysis ¨. (Section 4).

[71] Elaine Sedenberg and John Chuang. Smile for the Camera : Privacy and Policy Implications of Emotion AI. pages 1–12.

[72] Terrence J Sejnowski and Charles R Rosenberg. NETtalk : A Parallel Network That Learns to Read Aloud. 1986.

[73] Thibaud Senechal, Daniel Mcduff, and Rana Kaliouby. Facial Action Unit Detection using Active Learning and an Efficient Non-Linear Kernel Approximation.

[74] Daniel Llatas Spiers. Facial emotion detection using deep learning. 2016.

[75] Fritz Strack. No Title. 2010.

[76] Ilya Sutskever. Sequence to Sequence Learning with Neural Networks. pages 1–9.

[77] Christian Szegedy, Scott Reed, Pierre Sermanet, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. pages 1–12.

[78] Related Work There, Learning Rate, and Annealing There. (3) 1 where H.

[79] Claude Touzet, Claude Touzet, L E S Reseaux, D E Neurones Artificiels, and Introduction A U Connex. HAL Id : hal-01338010 INTRODUCTION AU Claude TOUZET Juillet 1992. 2016.

[80] Rudolph Triebel. Neural Networks and Deep Learning. 1 :2–4, 2016.

[81] Matthew Turk. Perceptive Media : Machine Perception and Human Computer Interaction.

[82] Terry T Um, Franz M J Pfister, Ludwig-maximilians-univ München, Daniel Pichler, Satoshi Endo, Muriel Lang, Urban Fietzek, and Dana Kulić. Data Augmentation of Wearable Sensor Data for Parkinson ' s Disease Monitoring using Convolutional Neural Networks . 2017.

[83] Valentin Vielzeuf, Stéphane Pateux, Frédéric Jurie, Valentin Vielzeuf, Stéphane Pateux, Frédéric Jurie, Temporal Multimodal, and Valentin Vielzeuf. Temporal Multimodal Fusion for Video Emotion Classification in the Wild To cite this version : HAL Id : hal-01590608 Temporal Multimodal Fusion for Video Emotion Classification in the Wild. 2017.

[84] Jun Wang and Lijun Yin. Author ' s personal copy Static topographic modeling for facial expression recognition and analysis.

[85] Lei Wang, Ce Zhu, Jieping Ye, and Juergen Gall. Signal Processing : Image Communication Guest Editors ' Introduction : Special issue on deep learning with applications to visual representation and analysis. *Signal Processing : Image Communication*, 47 :463–464, 2016.

[86] Paul J Werbos. Backpropagation Through Time : What It Does and How to Do It. 78(October) :1550–1560, 1990.

[87] Olga Wichrowska, Niru Maheswaranathan, and Matthew W Hoffman. Learned Optimizers that Scale and Generalize. (1987), 2017.

[88] Ren Wu. Deep Image : Scaling up Image Recognition. 2014.

[89] Robert H Wurtz. Recounting the impact of Hubel and Wiesel. 12 :2817–2823, 2009.

[90] Yaser Yacoob and Larry Davis. Recognizing Facial Expressions by Spatio-Temporal Analysis. pages 2–4.

[91] Jaime Zaratiegui, Ana Montoro, and Federico Castanedo. Actual Telecommunication Challenges.

[92] Matthew D Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. pages 818–833, 2014.