

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de Fin d'études Master

Filière : Informatique

Option : Système Informatique

Thème :

La reconnaissance des expressions faciales

Encadré par :

Dr. KOUAHLA Mohamed Nadjib

Présentée par :

DIALLO Nene Adama Dian

Juillet 2019

Résumé

Une expression faciale est un ou plusieurs mouvements ou positions des muscles sous la peau du visage. Ces mouvements traduisent l'état émotionnel d'un individu aux observateurs. Plusieurs travaux ont été développés dans le domaine de l'apprentissage automatique sur la reconnaissance des expressions faciales en utilisant plusieurs algorithmes pour l'extraction des caractéristiques (statistiques ou structurelles) et des classifieurs. Ces travaux ont prouvé leurs puissances en termes du taux de reconnaissance sur les petites bases de données, tout de même, ces résultats restent limités dans le cadre de traitement de très grande masse de données.

Avec l'apparition du concept de deep Learning (apprentissage en profondeur) et les bases de données volumineuses, un nouvel axe de recherche est développé. Notre projet consiste à proposer une approche de reconnaissance des expressions faciales basée sur le deep Learning et plus particulièrement les réseaux de neurones convolutifs. Un test expérimental a été fait sur la base de données FER 2013, le taux de reconnaissance obtenu 61,24% est les résultats sont prometteuses.

Abstract

A facial expression is one or more movements or positions of the muscles under the skin of the face. These movements translate the emotional state of an individual to the observers. Several works have been developed in the field of machine learning on the recognition of facial expressions using several algorithms for the extraction of characteristics (statistical or structural) and classifiers. This work has proven its power in terms of the recognition rate on small databases, all the same, these results are limited in the context of processing very large amounts of data.

With the emergence of the concept of deep learning and large databases, a new line of research is being developed. Our project consists of proposing a facial recognition approach based on deep learning and more particularly on convolutional neural networks. An experimental test was made on the basis of data FER 2013, the recognition rate obtained 61, 24% is the results are promising.

REMERCIEMENT

Au terme de ce mémoire, je tiens à exprimer mes remerciements et ma profonde gratitude avant tout au
bon

DIEU de m'avoir donné la force et le courage pour mener à bien ce modeste travail, je saisis de cette occasion pour exprimer mon profond remerciement à mon enseignant encadreur Monsieur Kouahla Mohamed Nadjib pour son soutien et son louable effort, durant toute la durée de ce travail, mes remerciements y vont aussi à l'endroit de tous les professeurs du département informatique de l'université 08 Mai 1945 de Guelma. Plus particulièrement à monsieur Hallaci Samir qui m'a été d'une très grande aide. Pour terminer, je remercie tous ceux qui nous ont aidés à élaborer ce travail de près ou de loin.

Dédicace

Je dédie ce mémoire aux plus merveilleux et adorables couples, à cette source de tendresse, de patience et de générosité qu'ils m'ont apportée et cette confiance qu'ils ont placée en moi. À mes chers parents « Abdourahmane Diallo et Mariama Souare » Auxquelles je dois ce que je suis aujourd'hui grâce à votre amour, à votre patience et vos innombrables sacrifices. Je prie le bon Dieu de vous donner une longue vie et une santé de fer.

À mes chers tantes et oncles qui ont toujours été à mes côtés « Boubacar Souare, Alpha Oumar Souare, Mamadou Mountagha Souare, Kadiatou Souare, Adama Djoulde Souare, Halimatou Souare ».

À mes chers frères et sœurs : « Alassane Diallo, Fatoumata Binta Diallo, Houssainatou Diallo, Kadiatou Diallo, Boubacar Diallo » pour tout l'amour dont vous m'avez apporté.

À mes chères amies : « Keita Kadia, Sidibé Aïssata, Keïta Aïcha, Boh Fanta Conde, Sidy Diallo » en souvenir des plus beaux instants qu'on a passés ensemble. Aux étudiants la promotion 2014/2015 filière : Informatique pour toute la joie et pour tout le bonheur dont vous m'avez comblé.

À tous ceux qui par un mot m'ont donné la force de continuer je vous remercie de tout cœur.

Table des matières

Résumé.....	1
Abstract	2
Liste des Figures	3
Liste des tableaux.....	4
Introduction générale.....	6
CHAPITRE 01 : Expression faciale et Deep Learning	8
1. Introduction.....	9
2. Expression faciale	9
2.1 Définition.....	9
2.2 Différents types des expressions faciales :.....	9
2.3 Description des six expressions faciales.....	10
2.4 Les différentes méthodes développées pour la reconnaissance faciale.....	12
2.4.1. Méthodes globales	12
2.4.2. Méthodes locales	13
2.5 Synthèse des travaux développés pour la reconnaissance d'expressions faciales	15
3. Réseaux de neurones	16
3.1 Définition.....	16
3.2 Historique	16
3.3 Topologie.....	17
4. L'apprentissage en profondeur (deep Learning).....	17
4.1 Définition.....	18
4.2 Historique	18
4.3 Pour quoi le choix deep Learning.....	19
4.4 Les différentes Architectures du Deep Learning.....	20
4.4.1 Les réseaux de neurones convolutifs	20
4.4.2 Réseau de neurones récurrents	20
4.4.3 Modèle génératif.....	21
4.4.4 Exemples d'application de Deep Learning	22
Conclusion	22
Chapitre 02 : CNN pour la reconnaissance des expressions faciales	23
1. Introduction.....	24
2. Architecture du système	24
.....	25

3.	Les différentes couches de CNN.....	25
a)	L'étape de convolution :.....	25
b)	L'étape de pooling.....	27
c)	La couche totalement connectée.....	28
4.	Optimisation pour l'apprentissage en Deep Learning.....	29
a.	La descente de gradient.....	29
5.	Algorithmes d'optimisation de la descente de gradient.....	29
a.	Adagrad.....	29
b.	RMSprop.....	30
6.	Conclusion.....	30
CHAPITRE 03 : Implémentation et résultats expérimentaux.....		31
Introduction.....		32
1.	Environnement de développement.....	32
1.1.	Google Colab.....	32
1.2.	Pourquoi les GPU ?.....	33
1.3.	Python.....	33
2.	Bibliothèques utilisées.....	34
2.1.	OpenCV (Open Source Computer Vision Library).....	34
2.2.	Numpy :.....	34
2.3.	Matplotlib.....	34
2.4.	Keras.....	34
2.5.	Pandas.....	34
3.	BDD.....	35
4.	Implémentation et résultats.....	35
4.1.	Formation du model avec keras.....	36
Conclusion.....		44
Conclusion générale.....		45
Bibliographie :.....		i
Webographie.....		vi

Liste des Figures

Figure 1. Classification des algorithmes principaux utilisés en reconnaissance faciale.	15
Figure 2. Topologie des Réseaux de neurones artificiels.	17
Figure 3. La relation entre l'intelligence artificielle, le ML et le deep Learning.....	17
Figure 4. Schéma illustratif de DL avec plusieurs couches.	18
Figure 5. Comparaison entre la machine Learning et le Deep Learning.....	19
Figure 6. Différents modèles du Deep Learning.	21
Figure 7. Architecture générale d'un CNN.	24
Figure 8. Architecture proposée.	25
Figure 9.Phase de convolution.	25
Figure 10. Exemple de description de la couche pooling.	27
Figure 11.Calcul du pooling sur une image 4×4.	28
Figure 12.Mise à plat des images.	28
Figure 13.Environnement de Google Colab.....	32
Figure 14.Le logo de python.	33
Figure 15.Échantillon d'images BDD de Fer2013.....	35
Figure 16.formation du modèle avec keras	36
Figure 17.résultat du modèle obtenu.	38
Figure 18.entrainement du modèle avec les images.....	39
Figure 19.Résultat de l'entrainement du modèle avec les images.	39
Figure 20.définissons les émotions de la BDD de test et d'entrainement.....	40
Figure 21. Les émotions obtenues de la BDD de test et d'entrainement.	40
Figure 22. Affichage du Taux d'erreurs.	40
Figure 23.Taux d'erreurs.....	41
Figure 24.Affichage du Taux de précision.....	41
Figure 25.Le taux de précision.....	42
Figure 26. Insertions des images externes.....	43
Figure 27.Attribution des images au modèle.	44
Figure 28. Résultats de quelques images externes.	44

Liste des tableaux

Tableau 1. Différents types d'expressions faciales [2].	10
Tableau 2. Descriptions des six expressions faciales.....	12
Tableau 3. Synthèse des travaux développés.	15
Tableau 4. Tableau comparatif de certains travaux déjà existant [26].....	43

Introduction générale

Introduction générale

Une expression faciale est une manifestation visible de l'état émotionnel ou psychologique d'une personne. La détection d'expression faciale s'avère être l'une des applications les plus pertinentes dans de nombreux domaines tels que : la formation assistée par ordinateur, le domaine médical, le domaine du traitement d'image et de reconnaissance des formes. Les expressions faciales varient d'un individu à un autre, c'est pourquoi cette tâche n'est pas facile à réaliser.

La reconnaissance automatique des expressions faciales suit le schéma général d'un processus de la reconnaissance des formes qu'est défini par les étapes : extraction des données, apprentissage-test, classification et décision.

Plusieurs travaux ont été développés dans le domaine de machine Learning sur la reconnaissance des expressions faciales en utilisant plusieurs algorithmes pour l'extraction des caractéristiques (statistiques ou structurelles) et des classifieurs. Ces travaux ont prouvé leurs puissances en termes du taux de reconnaissance sur les petites bases de données.

Tout de même, ces résultats restent limités dans le cadre de traitement de très grande masse de données telles que Facebook et Google qui ont été formé en utilisant 200 millions d'images et huit millions d'identités uniques. La taille de cet ensemble de données est presque trois fois plus grande que celle de tout ensemble de données de visage disponible publiquement.

Le deep Learning et plus particulièrement les réseaux de neurones convolutionnels (CNN) ont apparu spécialement pour résoudre les problèmes rencontrés du machine Learning. L'un des ingrédients les plus importants pour le succès de ces méthodes est la disponibilité de grandes quantités de données d'entraînement. Le Convolutional Neural Networks (CNN) est l'une des structures réseau les plus représentatives de la technologie d'apprentissage en profondeur et a connu un grand succès dans le domaine du traitement et de la reconnaissance d'images.

L'objectif de ce mémoire consiste à proposer une approche de reconnaissance des expressions faciales en se basant sur la méthode des réseaux de neurones convolutifs.

Structure du mémoire

Ce mémoire se présente sous forme de trois chapitres :

Le chapitre 1 nous présente un aperçu général sur la reconnaissance des expressions faciales et du Deep Learning.

Le chapitre 2 est consacré à l'étude du modèle proposé à savoir le réseau de neurones à convolution et les différents algorithmes d'apprentissages utilisés.

Le chapitre 3 illustre l'implémentation et l'expérimentation de notre système, les outils et logiciels que nous avons eu à utiliser pour le développement, et la réalisation de notre système.

Et enfin, nous terminerons ce mémoire par une conclusion générale et quelques perspectives.

CHAPITRE 01 : Expression faciale et Deep Learning

1. Introduction

De nos jours le suivi et la reconnaissance des activités du visage à partir d'images à attirer une grande attention dans le champ de vision par ordinateur. Parmi eux, la reconnaissance de l'expression du visage a été un sujet de recherche active depuis la dernière décennie. Les expressions faciales sont parmi les formes les plus universelles du langage corporel. Elles traduisent l'état émotionnel d'un individu aux observateurs. En général, les chercheurs ont divisé les expressions faciales en six grandes catégories : la colère, le dégoût, la peur, la joie, la tristesse et la surprise ; qui sont aussi appelés émotions primaires.

Dans ce premier chapitre, nous présentons quelques notions concernant les expressions faciales telles que leurs définitions, leurs différentes théories, et leurs composantes. Ensuite, nous décrivons un état de l'art sur les méthodes développées dans le domaine de la reconnaissance expressions faciales et plus particulièrement nous abordons la notion du deep Learning et ses différentes architectures existantes.

2. Expression faciale

2.1 Définition

Une expression faciale est une manifestation visible d'un visage de l'état d'esprit (émotion, réflexion), l'activité cognitive, physiologique (fatigue, douleur), personnalité et psychopathologie d'une personne. Elle repose sur trois principales caractéristiques influençant sur la nature de l'expression faciale à savoir : la bouche, les yeux et les sourcils. Ainsi caractérisant une déformation du visage par un changement perceptible visuel. [1]

2.2 Différents types des expressions faciales :

Le tableau représenté ci-dessous (tableau 1) présente une classification des expressions faciales selon certains chercheurs à base d'émotion et d'inclusion.

Théoriciens	Émotions de base	Base d'inclusion
Plutchik	Acceptation, colère, anticipation, dégoût, joie, peur, tristesse, surprise	Relation aux processus biologiques adaptatifs
Arnold	Colère, aversion, courage, abattement, désir, désespoir, peur, haine, espoir, amour, tristesse	Relation aux tendances d'action
<u>Frijda</u>	Désir, bonheur, intérêt, surprise, émerveillement, chagrin	Désir, bonheur, intérêt, surprise, émerveillement, chagrin
McDougall	Colère, dégoût, exaltation, peur, soumission, émotion tendre, émerveillement	Relation à l'instinct
<u>Ekman</u> , Friesen et Ellsworth	Colère, dégoût, peur, joie, tristesse, surprise	Expressions faciales universelles

Tableau 1. Différentes classifications d'expressions faciales [2].

2.3 Description des six expressions faciales

Lors de la production d'une expression faciale, il apparaît sur le visage un ensemble de déformation au niveau des traits permanents du visage.

Les émotions les plus fréquemment étudiées et utilisées sont les six émotions d'Ekman : la peur, la colère, la joie, la tristesse, le dégoût, la surprise [3].

Joie : Elle se caractérise par l'état d'une personne dans une condition de satisfaction intense [4]. Elle est due par rapport au désir, à la réussite, au bien-être, et l'accomplissement [5], mais aussi l'approche.

Tristesse : c'est l'état d'une personne qui souffre moralement suite à une insatisfaction et des soucis [4]. Elle est souvent due soit à une perte, ou un deuil, ou un obstacle [5], la personne se replie souvent sur soi.

Chapitre 01 : Expression faciale et Deep Learning

Colère : c'est l'état d'une personne dans une réaction violente et agressive lors d'une contrariété [4]. Elle est souvent due soit à une injustice, ou un dommage, atteinte au système de valeurs [5]. La plupart des personnes en ce moment attaquent.

Dégout : c'est l'état d'une personne qui a une répugnance pour certains aliments ou à un manque d'appétit [4]. Elle est souvent due soit à un rejet, ou contre quelqu'un, ou à une aversion [5]. La plupart des personnes en ce moment préfèrent se retirer.

Peur : c'est l'état d'une personne menacée par un danger réel ou imaginaire [5]. Elle est souvent due soit à une menace, ou à un danger, ou à des inconnus [5]. La plupart des personnes en ce moment préfèrent prendre la fuite.

Surprise : c'est l'état d'une personne étonnée par quelque chose d'inattendu [4]. Elle est souvent due soit à un danger immédiat, ou à un imprévu, ou à des inconnus [4]. La plupart des personnes en ce moment préfèrent prendre la fuite ou elles sursautent.

NB : Bien que nous avons tenté de définir les expressions et leurs caractéristiques, mais une chose reste à retenir que ces expressions varient souvent d'une personne à l'autre.

Expressions	Distance entre paupières	Distance entre œil et sourcil	Distance entre les coins de la bouche	Distance entre lèvre supérieure et lèvre inférieure	Distance entre les coins de l'œil et de la bouche
Joie 	accroît ou décroît	accroît ne change pas ou décroît	accroît	ne change pas ou accroît	décroît
Surprise 	Accroît	Accroît	ne change pas ou décroît	Accroît	ne change pas ou accroît
Dégout 	Décroît	décroît	accroît ne change pas ou décroît	Accroît	accroît ne change pas ou décroît

<p>Colère</p> 	<p>Décroît où accroît</p>	<p>décroît</p>	<p>ne change pas ou décroît</p>	<p>accroît ne change pas ou décroît</p>	<p>ne change pas ou accroît</p>
<p>Tristesse</p> 	<p>Décroît</p>	<p>Accroît</p>	<p>ne change pas ou accroît</p>	<p>ne change pas ou accroît</p>	<p>ne change pas ou décroît</p>
<p>Peur</p> 	<p>ne change pas ou accroît</p>	<p>ne change pas ou accroît</p>	<p>accroît ne change pas ou décroît</p>	<p>ne change pas ou accroît</p>	<p>accroît ne change pas ou décroît</p>

Tableau 2. Descriptions des six expressions faciales [4].

2.4 Les différentes méthodes développées pour la reconnaissance faciale

Ils existent différentes approches qui ont été développées dans le domaine de reconnaissance des expressions faciales. Elles peuvent être séparées en deux grandes familles à savoir : les méthodes globales (ou holistiques) et les méthodes locales. [6]

2.4.1. Méthodes globales

Elles sont basées sur des techniques d'analyse statistique bien connues. Il n'est pas nécessaire de repérer certains points caractéristiques du visage (comme les centres des yeux, les narines, le centre de la bouche, etc.) à part pour normaliser les images. Dans ces méthodes, les images de visage (qui peuvent être vues comme des matrices de valeurs de pixels) sont traitées de manière globale et sont généralement transformées en vecteurs, plus faciles à manipuler. L'avantage principal des méthodes globales est qu'elles sont relativement rapides à mettre en œuvre et que les calculs de base sont d'une complexité moyenne. En revanche, elles sont très sensibles aux variations d'éclairage, de pose et d'expression faciale.

Ceci se comprend aisément puisque la moindre variation des conditions de l'environnement ambiant entraîne des changements inéluctables dans les valeurs des pixels qui sont traités directement. Ces méthodes utilisent principalement une analyse de sous-espaces de visages. L'utilisation de techniques de modélisation de sous-espace a fait avancer la technologie de reconnaissance faciale de manière significative.

Chapitre 01 : Expression faciale et Deep Learning

Nous pouvons distinguer deux types de techniques parmi les méthodes globales : les techniques linéaires et les techniques non linéaires.

Les techniques linéaires projettent linéairement les données d'un espace de grande dimension (par exemple, l'espace de l'image originale) sur un sous-espace de dimension inférieure. Malheureusement, ces techniques sont incapables de préserver les variations non convexes des variétés (géométriques donc au sens mathématique du terme) de visages afin de différencier des individus. Dans un sous-espace linéaire, les distances euclidiennes et plus généralement les distances de Mahalanobis, qui sont normalement utilisées pour faire comparer des vecteurs de données, ne permettent pas une bonne classification entre les classes de formes et entre les individus eux-mêmes. La technique linéaire la plus connue et sans aucun doute l'analyse en composantes principales (PCA), également appelée transformée de Karhunen-Loeve. Le PCA fut d'abord utilisé afin de représenter efficacement des images de visages humains. [7] [8] [9] Bien que ces méthodes globales linéaires basées sur l'apparence évitent l'instabilité des toutes premières méthodes géométriques qui ont été mises au point, elles ne sont pas assez précises pour décrire les subtilités des variétés (géométriques) présentes dans l'espace de l'image originale. Ceci est dû à leurs limitations à gérer la non-linéarité en reconnaissance faciale : les déformations de variétés non linéaires peuvent être lissées et les concavités peuvent être remplies, causant des conséquences défavorables.

Afin de pouvoir traiter ce problème de non-linéarité sur la reconnaissance des expressions faciales, de telles méthodes linéaires ont été étendues à des techniques non linéaires basées sur la notion mathématique de noyau ("kernel") comme le Kernel PCA et le Kernel LDA [10].

2.4.2. Méthodes locales

Elles sont basées sur des modèles, utilisant des connaissances a priori que l'on possède sur la morphologie du visage et s'appuie en général sur des points caractéristiques de celui-ci. Kanade a présenté un des premiers algorithmes de ce type [11] en détectant certains points ou traits caractéristiques d'un visage puis en les comparant avec des paramètres extraits d'autres visages. Ces méthodes constituent une autre approche pour prendre en compte la non-linéarité en construisant un espace de caractéristiques local et en utilisant des filtres d'images appropriés, de manière à ce que les distributions des visages soient moins affectées par divers changements.

Les approches bayésiennes, les machines à vecteurs de support (SVM) [12], la méthode des modèles actifs d'apparence (AAM) [13] ou encore la méthode "local binary pattern"(LBP) ont été utilisées dans ce but.

Chapitre 01 : Expression faciale et Deep Learning

Toutes ces méthodes ont l'avantage de pouvoir modéliser plus facilement les variations de pose, d'éclairage et d'expression par rapport aux méthodes globales. Toutefois, elles sont plus lourdes à utiliser puisqu'il faut souvent placer manuellement un assez grand nombre de points sur le visage alors que les méthodes globales ne nécessitent de connaître que la position des yeux afin de normaliser les images, ce qui peut être fait automatiquement et de manière assez fiable par un algorithme de détection [14]

2.4.3. Méthodes hybrides

Ce sont une combinaison des méthodes globales et locales en combinant la détection de caractéristiques géométriques (ou structurales) avec l'extraction de caractéristiques d'apparence locales. Elles permettent d'augmenter la stabilité de la performance de reconnaissance lors de changements de pose, d'éclairage et d'expressions faciales. [15]

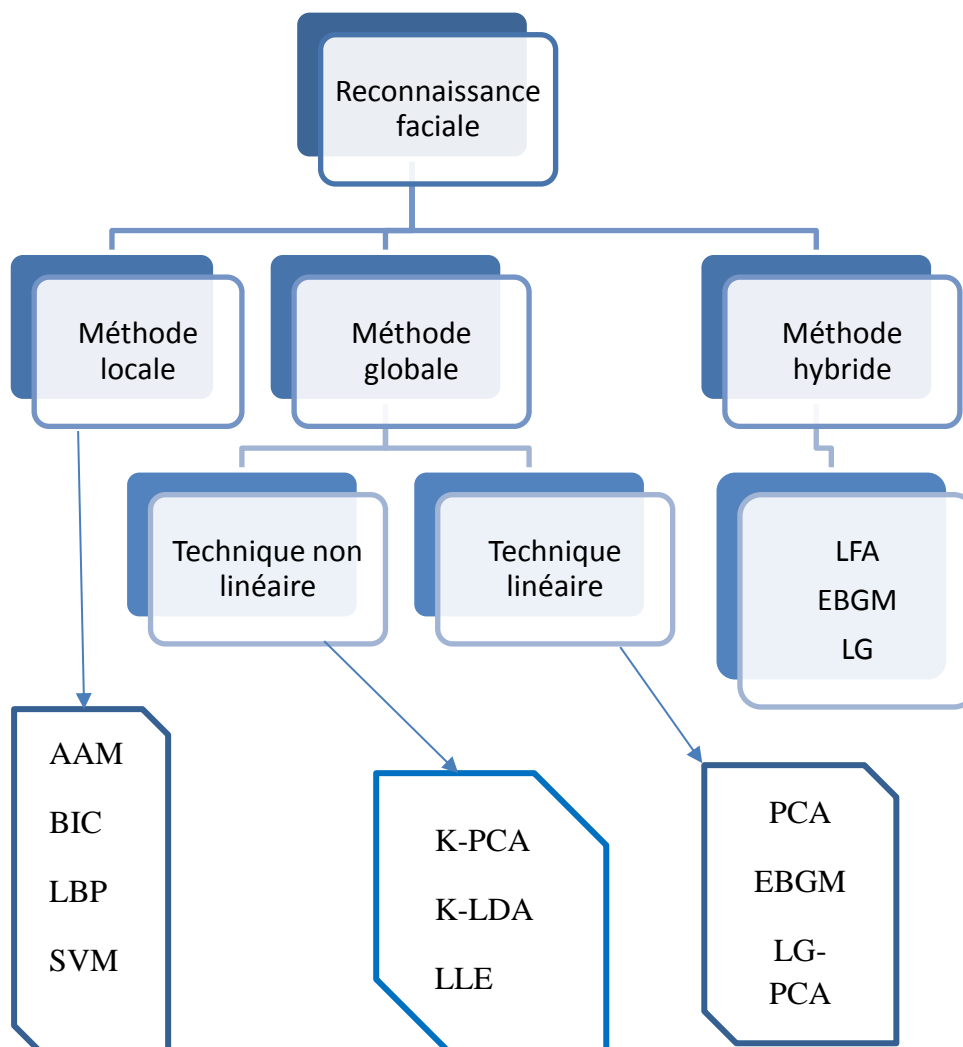


Figure 1. Classification des algorithmes principaux utilisés en reconnaissance faciale[10][13][15].

2.5 Synthèse des travaux développés pour la reconnaissance d'expressions faciales

Le tableau suivant (tableau 1.3) présente quelques techniques récentes développées pour la reconnaissance des expressions faciales :

Références	Techniques	Expressions faciales	Bases de données	Types	Sujets	Performance
[16]	LPB, SVM	Six	CK+	images	50	97%
[17]	EBGM,ELM	Six	CK+	Images	40	96%
[18], [19]	PCA, LDA	Six	CK+	193 images	9	92% 75%
[20]	SVM	Sourire spontané et sourire surnois	MMI	vidéos	52	94%
[21]	HOG, SVM	Six	CK+	images		90%
[22]	HMMS	Sept	-Propre base Cohn-kanade	Videos	5 sujets 53 sujets	84,46% 58,63%
[23]	MAA	Sept	CMU	166 images	30 sujets	84, 34%
[24]	Codification Des règles	Six		images	8 sujets	92%
[25]	Multiboosts, SVM linéaire	Six	BU-3DDFE	Scans 3D	60 sujets	97,75% 98,81%
[26]	CNN	Sept	Fer2013	images	35000 sujets	66 ,67%
[27]	CNN	Six	CK+	images	123 sujets	91%

Tableau 3. Synthèse des travaux développés.

3. Réseaux de neurones

3.1 Définition

Un réseau de neurones artificiels est un système dont la conception est à l'origine inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques [W1]. Les réseaux de neurones artificiels sont des réseaux fortement connectés par des processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire (neurone artificiel) calcule une sortie unique sur la base des informations qu'ils reçoivent.

3.2 Historique

Les réseaux neuronaux ont vu le jour qu'en 1943 par W.MCCulloch et W. Pitts du neurone formel qui est une abstraction du neurone physiologique. Par cette présentation, ils ont pu démontrer que le cerveau est équivalent à une machine de Turing, la pensée devient alors purement des mécanismes matériels et logiques. Ils déclarèrent en 1955 "Plus nous apprenons de choses au sujet des organismes, plus nous sommes amenés à conclure qu'ils ne sont pas simplement analogues aux machines, mais qu'en est-il de cette machine. La démonstration de McCulloch et Pitts a été l'un des acteurs importants de la création de la cybernétique.

En 1949, D. Hebb présenta dans son ouvrage "The Organization of Behavior" une règle d'apprentissage. De nombreux modèles de réseaux aujourd'hui s'inspirent encore de la règle de Hebb.

En 1958, F. Rosenblatt développe le modèle du Perceptron. C'est un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones : une couche de perception et une couche liée à la prise de décision. C'est le premier système artificiel capable d'apprendre par expérience. Dans la même période, le modèle de L'Adaline (ADaptive LINar Element) a été présenté par B. Widrow, chercheur américain à Stanford. Ce modèle sera par la suite le modèle de base des réseaux multicouches.

En 1969, M. Minsky et S. Papert publient une critique des propriétés du Perceptron. Cela va avoir une grande incidence sur la recherche dans ce domaine. Elle va fortement diminuer jusqu'en 1972, où T. Kohonen présente ses travaux sur les mémoires associatives et propose des applications à la reconnaissance de formes.

C'est en 1982 que J. Hopfield présente son étude d'un réseau complètement rebouclé, dont il analyse la dynamique.

Chapitre 01 : Expression faciale et Deep Learning

Aujourd'hui, les réseaux neuronaux sont utilisés dans de nombreux domaines (entre autres, vie artificielle et intelligence artificielle) à cause de leur propriété en particulier, leur capacité d'apprentissage, et qu'ils soient des systèmes dynamiques [W2].

3.3 Topologie

Chaque réseau de neurones est connecté entre eux de diverses manières. De la figure suivante. Nous pouvons distinguer deux familles de réseaux de neurones : non bouclés ou statiques (a) et (b) et bouclés (dynamiques) (c) et (d).

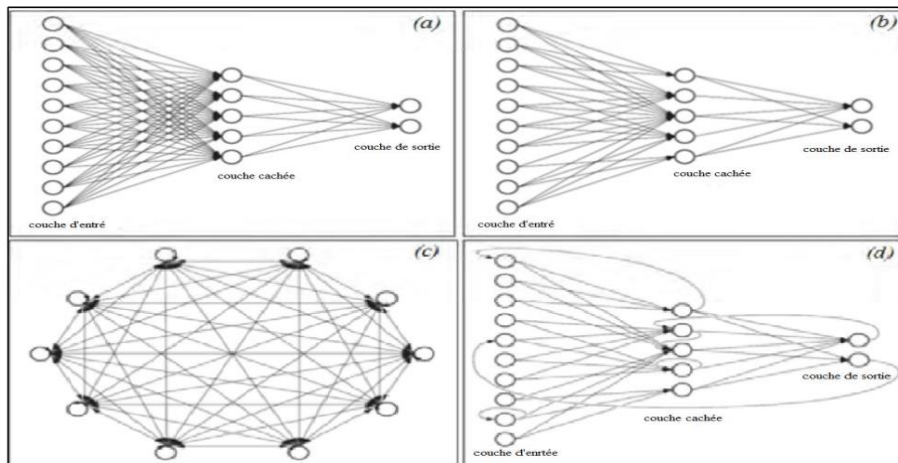


Figure 2. Topologie des Réseaux de neurones artificiels.

4. L'apprentissage en profondeur (deep Learning)

Le Deep Learning est un nouveau domaine de recherche de la machine Learning (ML), qui a été introduit dans le but de rapprocher le ML de son objectif principal à savoir : l'intelligence artificielle. Il concerne les algorithmes inspirés par la structure et du fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données.

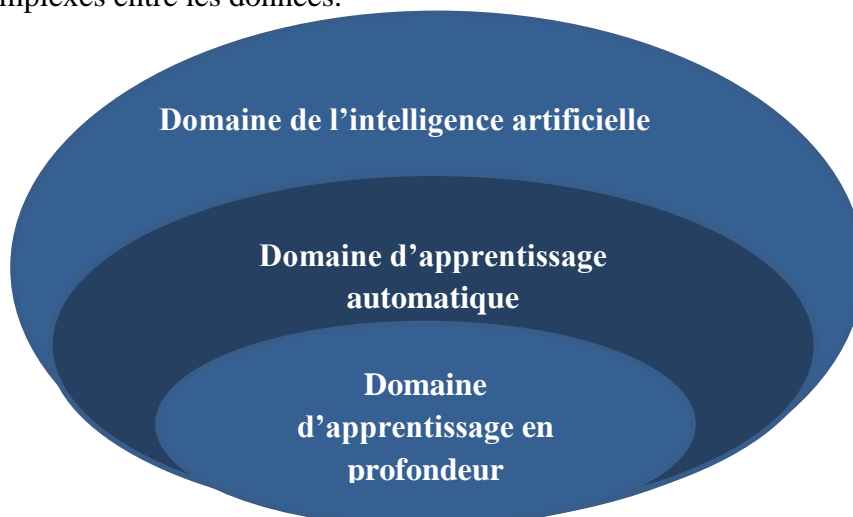


Figure 3. La relation entre l'intelligence artificielle, le ML et le deep Learning

4.1 Définition

L'apprentissage en profondeur est un ensemble d'algorithmes d'apprentissage automatique qui tentent d'apprendre à plusieurs niveaux, correspondant à différents niveaux d'abstraction. Il a la capacité d'extraire des caractéristiques à partir des données brutes grâce aux multiples couches de traitement composé de multiples transformations linéaires et non linéaires et apprendre sur ces caractéristiques petites à petit à travers chaque couche avec une intervention humaine minimale [28] [29] [30] [31] [32] [33] [34].

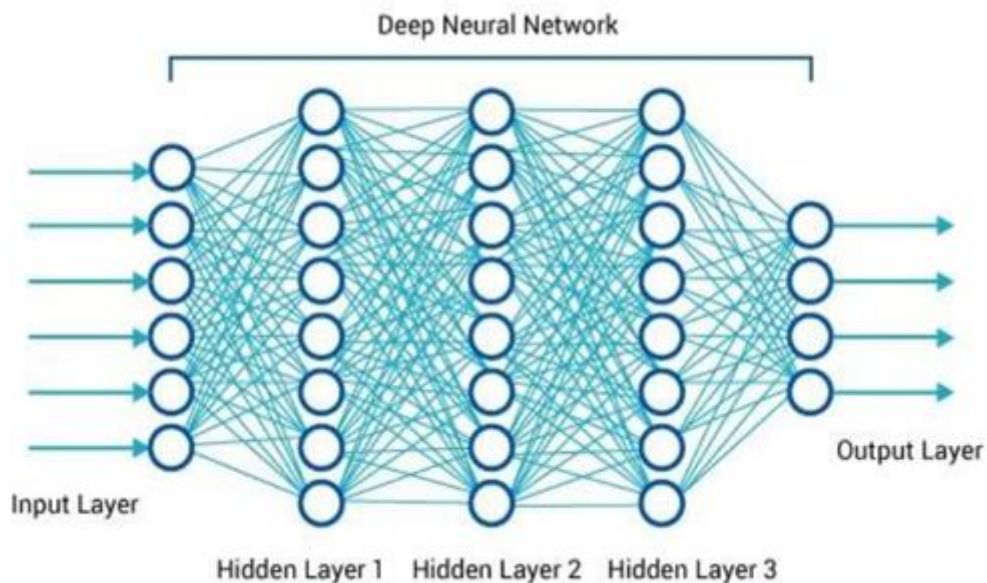


Figure 4. Schéma illustratif de DL avec plusieurs couches.

4.2 Historique

L'idée du Deep Learning n'est pas une idée récente, mais elle date en réalité des années 1980, plus particulièrement suite aux travaux de réseaux de neurones multicouches et aux travaux de certains pionniers du machine Learning et du Deep Learning comme le français Yann Le Cun. En collaboration avec deux autres informaticiens, Kunihiko Fukushima et Geoffrey Hinton, ils mettent au point un type d'algorithme particulier appelé Convolutional neural network.

Bien que cette approche donne des résultats, ses progrès et son évolution sont limités par les progrès technologiques en matière de micro-processeurs, de puissance de calculs, et du manque d'accessibilités à des données afin de pouvoir entraîner les neurones. Cependant certains chercheurs ont continué à travailler sur ce modèle pendant environ deux décennies et, avec l'aide des évolutions en matière de technologies, mais surtout avec la disponibilité toujours plus grande de données, ont pu améliorer cette technique.

Chapitre 01 : Expression faciale et Deep Learning

Afin de développer un système d'apprentissage performant, il faut pouvoir l'exercer et cela requiert un nombre important de données à tester. C'est dans ce contexte qu'en 2007 le STANFORD VISION LAB, avec Fei-Fei Li à sa tête, développent un agrégateur d'images où sont consignés et étiquetés quelques millions de photos : ImageNet. En 2010, ImageNet regroupe 15 000 000 d'images toutes catégorisées en fonction de leurs caractéristiques propres (véhicules, animaux, ...) [W3].

En 2012, le Deep Learning est remis au goût du jour avec un succès retentissant au ImageNet Large Scale Visual Recognition Challenge (ILSVRC) qui est un concours annuel de reconnaissance d'image fondée par l'université de Stanford, dans le cadre de son laboratoire STANFORD VISION LAB [W3]. Plusieurs équipes de chercheurs en informatique s'affrontent dans ce concours tous les ans afin de décerner la victoire au programme ayant eu le plus faible taux d'échec. Et alors que les algorithmes d'apprentissage profond sont absents de la compétition, en 2012 c'est bel et bien un algorithme de Deep Learning qui va remporter l'édition 2012 à la surprise générale [W4].

4.3 Pour quoi le choix deep Learning

Tout d'abord les différents algorithmes du deep Learning ne sont apparus qu'à l'échec de l'apprentissage automatique tentant de résoudre une grande variété de problèmes de l'intelligence artificielle (l'IA) :

- Afin d'améliorer le développement des algorithmes traditionnels dans de telles tâches de l'IA.
- De développer une grande quantité de données telle que les big data.
- De s'adapter à n'importe quel type de problème
- D'extraire les caractéristiques de façon automatique [35]

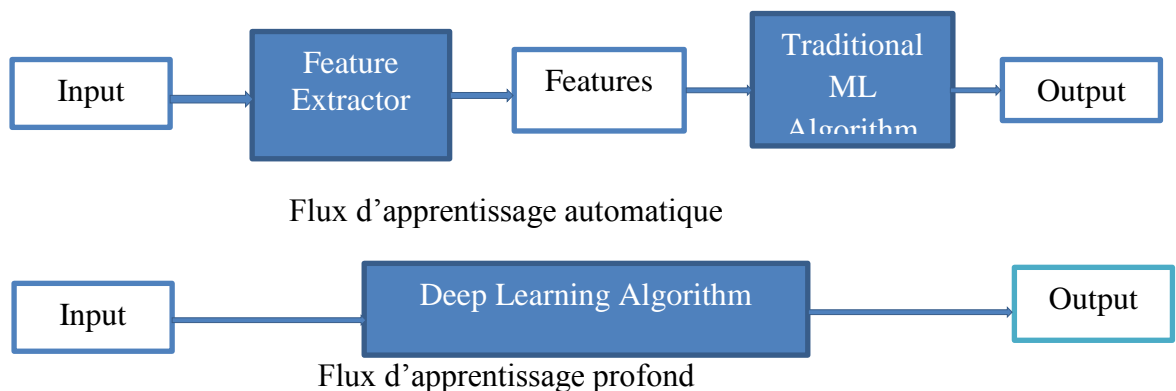


Figure 5. Comparaison entre la machine Learning et le Deep Learning.

4.4 Les différentes Architectures du Deep Learning

Bien qu'il existe un grand nombre de variantes d'architectures profondes. Il n'est pas toujours possible de comparer les performances de toutes les architectures, car elles ne sont pas toutes évaluées sur les mêmes ensembles de données. Le Deep Learning est un domaine à croissance rapide, et de nouvelles architectures, variantes ou algorithmes apparaissent toutes les semaines.

4.4.1 Les réseaux de neurones convolutifs

Convolutional Neural Network (CNN) (réseaux de neurones convolutifs) sont un type de réseau de neurones spécialisés pour le traitement de données ayant une topologie semblable à une grille. Qui se sont avérés très efficaces dans des domaines tels que la reconnaissance et la classification d'images et vidéos. CNN a réussi à identifier les visages, les objets, panneaux de circulation et auto-conduite des voitures [36]. Récemment, les CNN ont été efficaces dans plusieurs tâches de traitement du langage naturel (telles que la classification des phrases) [37]. [38] [39].

Dans le ML, un réseau convolutif est un type de réseau de neurones feed-forward, il a été inspiré par des processus biologiques [40]. Il existe quatre (4) principales opérations illustrées dans le CNN à savoir :

- La couche convolution
- La couche Rectified Linear Unit
- La couche Pooling
- La couche entièrement connectée

4.4.2 Réseau de neurones récurrents

L'idée derrière les RNN est d'utiliser des informations séquentielles. Dans un réseau neuronal traditionnel, nous supposons que toutes les entrées (et les sorties) sont indépendantes les unes des autres. Mais pour de nombreuses tâches, c'est une très mauvaise idée. Si on veut prédire le prochain mot dans une phrase, il faut connaître les mots qui sont venus avant. Les RNN sont appelés récurrents, car ils exécutent la même tâche pour chaque élément d'une séquence, la sortie étant dépendante des calculs précédents.

Une autre façon de penser les RNN est qu'ils ont une « mémoire » qui capture l'information sur ce qui a été calculé jusqu'ici. En théorie, les RNN peuvent utiliser des informations dans des séquences arbitrairement longues, mais dans la pratique, on les limite à regarder seulement quelques étapes en arrière. [41][42][43] [44]. Il est utilisé pour :

Chapitre 01 : Expression faciale et Deep Learning

- La modélisation du langage et génération de texte
- La traduction automatique
- La reconnaissance vocale
- Et la description des images

4.4.3 Modèle génératif

Si les modèles discriminatifs comme (CNN, RNN) sont utilisés pour prédire les données du label et de l'entrée, tant disque le modèle génératif décrit comment générer les données, il apprend et fait des prédictions en utilisant la loi de Bayes [45].

Cependant les modèles génératifs sont capables de bien plus que la simple classification comme par exemple générer de nouvelles observations.

Voici quelques exemples de modèle génératif :

- Boltzmann Machines[46]
- Restricted Boltzmann Machines [47][48]
- Deep Belief Networks[49]
- Deep Boltzmann Machines[50][51]
- Generative Adversarial Networks• Generative Stochastic Networks[52]
- Adversarial autoencoders[53]

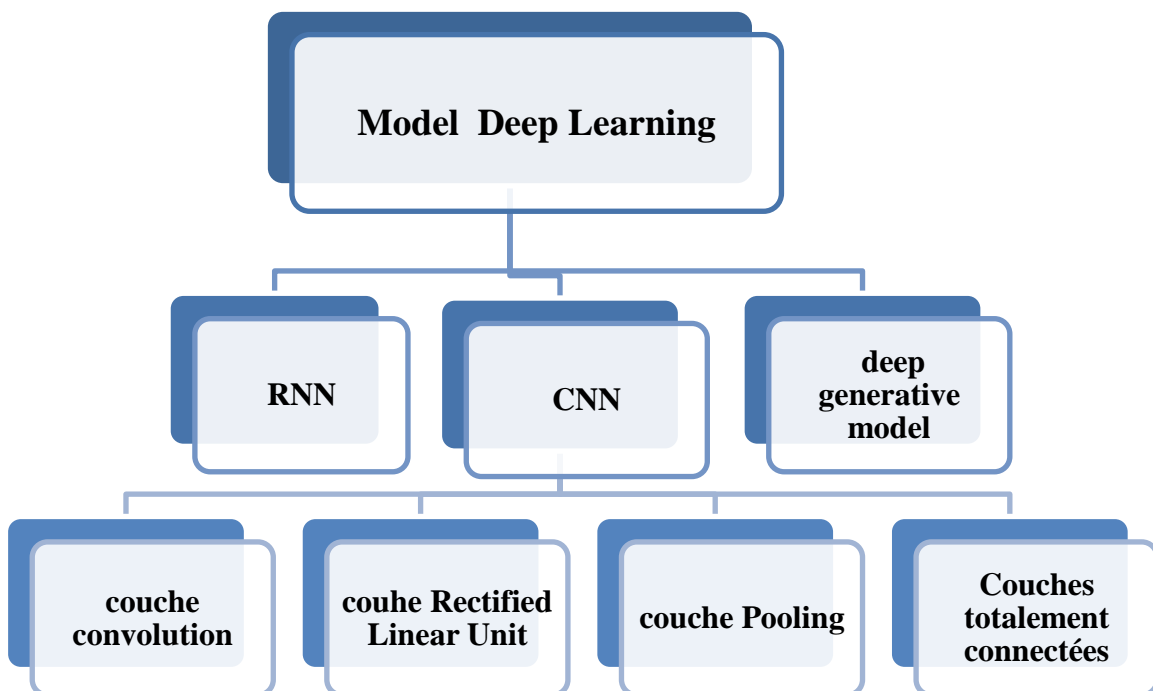


Figure 6. Différents modèles du Deep Learning.

Chapitre 01 : Expression faciale et Deep Learning

4.4.4 Exemples d'application de Deep Learning

Les applications du Deep Learning sont utilisées dans divers secteurs, de la conduite automatisée aux dispositifs médicaux. [W1]

Grace au deep Learning nous pouvons maintenant

- Faire une colorisation des images en noir et blanc. [54]
- Ajouter des sons à des films silencieux. [55]
- Faire de la traduction automatique. [56]
- Faire de classification des objets en photographies. [57]
- Générer d'écriture automatique. [58]
- Génération de légende d'image. [59]
- Jeu automatique [60]

Conclusion

Dans cette partie, nous avons présenté quelques méthodes existantes sur la reconnaissance de l'expression faciale. Ainsi nous avons eu à définir qu'est-ce qu'une expression faciale et qu'elles sont les plus utilisés, et qu'est-ce que le Deep Learning et comment il se différencie des algorithmes de ML traditionnelles.

Nous avons également parlé des trois familles majeures de modèle à savoir les réseaux convolutifs, les réseaux récurrents ainsi que les modèles génératifs, ces derniers font encore l'objet de recherches intensives.

Dans le second chapitre, nous rentrerons plus en détail sur les réseaux de neurones convolutifs (CNN).

Chapitre 02 : CNN pour la reconnaissance des expressions faciales

1. Introduction

Les réseaux de neurones à convolution (Convolutional Neural Networks, CNN) sont une catégorie de réseaux de neurones qui se sont avérés très efficaces dans des domaines tels que la reconnaissance et la classification d'images. CNN ont réussi à identifier les visages, les objets, panneaux de circulation et auto-conduite des voitures c' est raison pour laquelle que durant cette phase de formation de nous nous baserons uniquement sur le CNN.

2. Architecture du système

Nous allons commencer par entraîner un réseau de neurones convolutif des plus classiques, qui possède en entrée une base de données et paramètres entraînaables et il consiste en des couches convolutives, dont à chaque couche convolutives est placée une couche de Max pooling et enfin une couche totalement connectée. Pour un apprentissage plus rapide, ReLu sera la fonction d'activation.



Figure 7. Architecture générale d'un CNN.

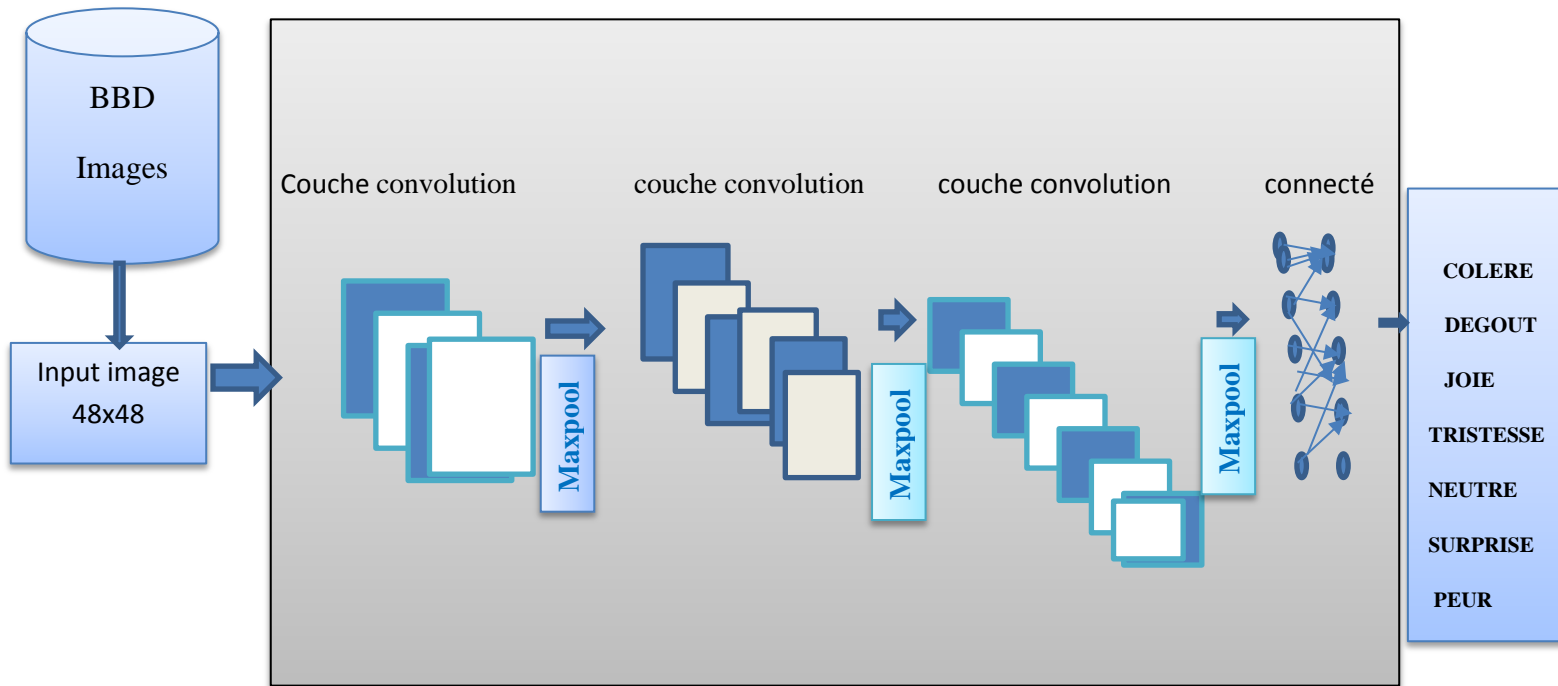


Figure 8. Architecture proposée.

3. Les différentes couches de CNN

Il existe quatre principales opérations dans le CNN, tout en ayant un rôle bien défini.

a) L'étape de convolution :

La **convolution** est le cœur du réseau de neurones convolutif, comme vous vous en doutez. À l'origine, une convolution est un outil mathématique (on parle de produit de convolution) très utilisé en retouche d'image, car il permet d'en faire ressortir l'extraction des caractéristiques à partir des images d'entrées, afin d'appliquer un bon **filtre**. En fait, une convolution prend simplement en entrée une image et un filtre (qui est une autre image), effectue un calcul, puis renvoie une nouvelle image (généralement plus petite). [61]

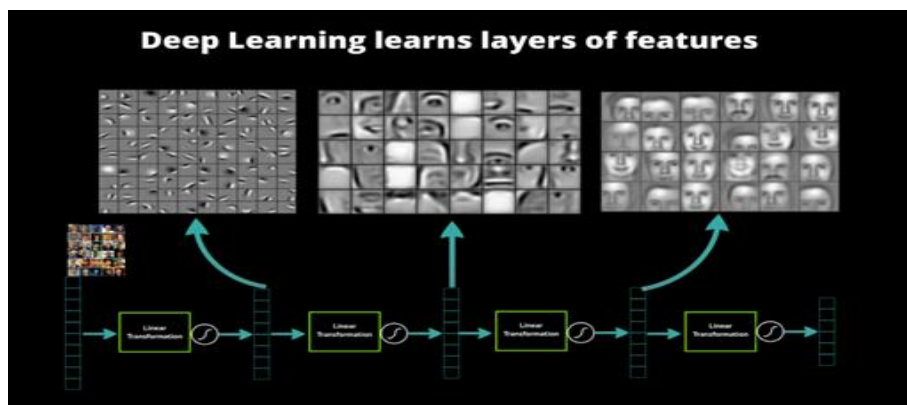


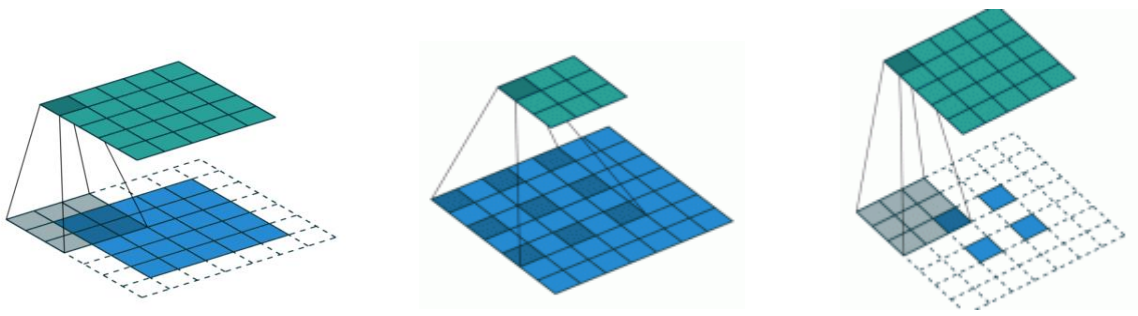
Figure 9. Phase de convolution.

Chapitre 02 : CNN pour la reconnaissance des expressions faciales

Les différentes convolutions

Il existe plusieurs types de convolutions, même si en général on utilise celle de base, il peut s'avérer utile de connaître les outils à notre disposition.

- La **convolution classique** qui représente le décalage du noyau entre chaque calcul, et le **padding** qui est la manière dont on peut « dépasser » de l'image pour appliquer la convolution.
- La **dilated convolution**, identique à la convolution à ceci près que le kernel est éclaté (on prend, par exemple, un pixel sur deux pour calculer la convolution). Il y a un paramètre supplémentaire : la **dilation rate**, qui est le nombre de pixels à ignorer.
- La **transposed convolution**, qui construit la sortie comme si on inversait une convolution sur l'image
- La **séparable convolution**, qui est une convolution décomposable en convolutions plus simples.



Une convolution de kernel

Une dilated convolution

Une transposed convolution

Figure 2.4 : différents types de convolutions.

Fonctions d'activation

Une fonction d'activation est une fonction mathématique appliquée à un signal en sortie d'un neurone artificiel. Il dérive de l'équivalent biologique qui signifie "**potentiel d'activation**, lorsque le seuil de stimulation aura été atteint entraîne une réponse du neurone. Son but principal est de pouvoir permettre aux réseaux de neurones d'apprendre des fonctions plus complexes qu'une simple régression linéaire, car le simple fait de multiplier les poids d'une couche cachée est juste une transformation linéaire :

- **Rectified Linear Unit (ReLU) :**

Elle est utilisée après chaque opération de convolution, ou toutes les valeurs de pixels négatifs sont mises à zéro. Le but de ReLU est d'introduire la non-linéarité dans notre CNN, puisque la

Chapitre 02 : CNN pour la reconnaissance des expressions faciales

plupart des données du monde réel, puisque la plupart de caractéristiques appliquées à l'une des cartes d'entrée donne une carte de sortie qui est également appelée carte de caractéristiques rectifiées. [62]

b) L'étape de pooling

Elle permet de réduire la dimension de chaque carte de caractéristiques, mais conserve l'information la plus importante. IL peut être de différents types : max, moyenne, somme, etc. En cas de Pooling max, on définit un voisinage spatial (par exemple, une fenêtre 2×2) et de prendre le plus grand élément dans cette fenêtre.

La fonction de Pooling consiste à réduire progressivement la taille de la carte de caractéristiques rectifiée, en particulier, pooling :

- Rend les représentations d'entrée plus petites et plus faciles à gérer,
- Réduit le nombre de paramètres et les calculs dans le réseau [63],
- rend le réseau invariant aux petites transformations, les distorsions et les translations dans l'image d'entrée (une faible distorsion en entrée ne change pas la sortie de la Pooling - car nous prenons le maximum / valeur moyenne dans un voisinage local).

Nous aide à arriver à une représentation presque invariante à l'échelle de notre image [64], [65].

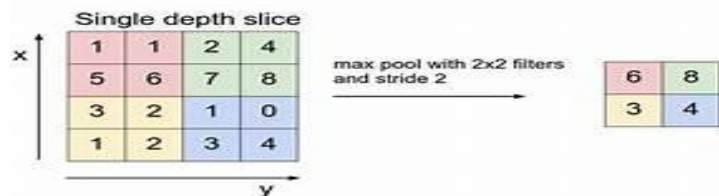


Figure 10. Exemple de description de la couche pooling.

Il existe plusieurs types de pooling :

- Le « max pooling », qui revient à prendre la valeur maximale de la sélection. C'est le type le plus utilisé, car il est rapide à calculer (immédiat), et permet de simplifier efficacement l'image
- Le « mean pooling » (ou average pooling), soit la moyenne des pixels de la sélection : on calcule la somme de toutes les valeurs et on divise par le nombre de valeurs. On obtient ainsi une valeur intermédiaire pour représenter ce lot de pixels
- Le « sum pooling », c'est la moyenne sans avoir divisé par le nombre de valeurs (on ne calcule que leur somme)

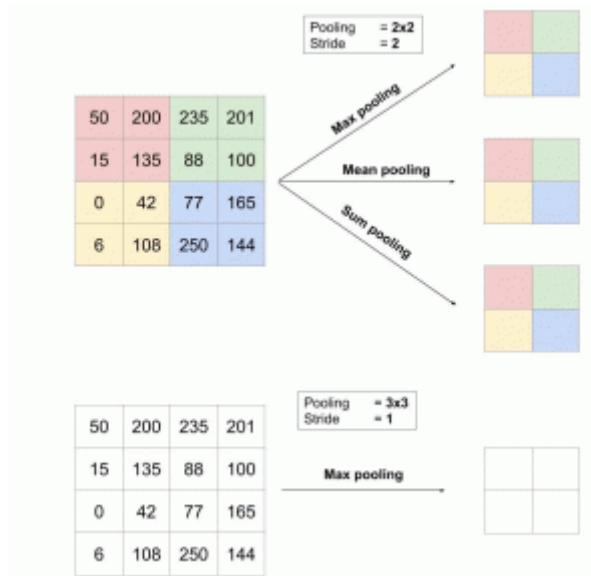


Figure 11. Calcul du pooling sur une image 4×4 .

La mise à plat

Il consiste simplement à mettre à bout toutes les images (matrices) que nous avons pour en faire un (long) vecteur. Les pixels (en réalité ce ne sont plus des images ou des pixels, mais des matrices de nombres, donc les pixels sont ces nombres) sont récupérés ligne par ligne et ajoutés au vecteur final.



Figure 12. Mise à plat des images.

c) La couche totalement connectée

La couche entièrement connectée est un traditionnel perceptron multicouche (Multi Layer Perceptron) utilisant une fonction d'activation notamment appelée « softmax » dans la couche de sortie (d'autres classificateurs comme SVM peuvent également être utilisés). Le terme «

entièrement connecté » implique que chaque neurone dans la couche précédente est connecté à chaque neurone sur la couche suivante.

La sortie des couches de convolution et de Pooling représente les fonctions de haut niveau de l'image d'entrée. Le but de la couche entièrement connectée est de pouvoir utiliser ces fonctions pour classer l'image d'entrée dans différentes classes en fonction de l'ensemble de données d'apprentissage. [66]

4. Optimisation pour l'apprentissage en Deep Learning

a. La descente de gradient

La descente de gradient est un algorithme d'optimisation souvent utilisé pour trouver les poids ou les coefficients des algorithmes d'apprentissage automatique, tels que les réseaux de neurones artificiels et la régression logistique.

Cela fonctionne en permettant au modèle de faire des prédictions sur les données d'apprentissage et en utilisant l'erreur sur les prédictions pour mettre à jour le modèle de manière à réduire l'erreur.

Le but de l'algorithme est de trouver des paramètres de modèle (par exemple, des coefficients ou des poids) qui minimisent l'erreur du modèle sur le jeu de données d'apprentissage. Pour ce faire, il modifie le modèle en le déplaçant le long d'une pente d'erreur vers une valeur d'erreur minimale. Cela donne à l'algorithme le nom de « descente de gradient ». [67]

Il existe trois variantes de cette méthode :

- Batch gradient descent
- Descente de gradient stochastique
- Mini-batch gradient descent [W5]

5. Algorithmes d'optimisation de la descente de gradient

a. Adagrad

Adagrad [68] est un algorithme d'optimisation basé sur la descente de gradient qui ne fait qu'adapter le taux d'apprentissage aux paramètres, effectuant de plus grandes mises à jour pour les caractéristiques peu fréquentes et de plus petites pour les caractéristiques plus fréquentes. Dean et al. [69] ont trouvé que Adagrad a grandement amélioré la robustesse de SGD et utilisé pour l'apprentissage de grands réseaux de neurones chez Google qui -entre autres- ont appris à reconnaître les chats dans les vidéos YouTube [70]. De plus Pennington et

al. [71] ont utilisé Adagrad pour l'apprentissage de GloVe word embeddings, comme les mots les moins fréquents demandent de plus grandes mises à jour que ceux qui sont plus fréquents.

b. RMSprop

RMSprop est une méthode de taux d'apprentissage adaptative qui a été proposée par Geoff Hinton [72]. Au lieu d'accumuler tous les carrés des gradients précédents, nous allons tout simplement restreindre la fenêtre des gradients accumulés à une taille fixée w . Au lieu de stocker les w carrés des gradients précédents, on applique une moyenne mobile exponentielle des carrés des gradients précédents.

6. Conclusion

Dans ce chapitre, nous avons tout d'abord défini l'architecture utilisée et aussi essayé d'expliquer les concepts principaux de réseau de neurone à convolution (CNN) en termes simples. Le CNN à quatre principales opérations : convolution, la fonction non-linéarité (ReLU), Pooling et couche entièrement connectée. Première opération est la convolution pour l'extraction de caractéristiques de l'image d'entrée. Deuxième opération est la fonction non-linéarité (ReLU) pour remplacer toutes les valeurs de pixels négatives par zéro. Troisième opération est la Pooling pour réduire progressivement la taille de la carte de caractéristiques rectifiée. Enfin une couche entièrement connectée pour la classification. Et aussi nous vous présentons quelques méthodes permettant la réduction des erreurs.

CHAPITRE 03 : Implémentation et résultats expérimentaux

Introduction

L'objectif de ce chapitre est de présenter les étapes de l'implémentation de l'approche proposée dans le cadre d'un système de reconnaissance des expressions faciales. Nous commençons tout d'abord par la présentation des ressources, du langage et de l'environnement de développement que nous avons utilisés. Puis les étapes de la réalisation du modèle et on termine par les tests effectués.

Ce chapitre est composé de deux parties, l'implémentation du système et les résultats expérimentaux des tests.

1. Environnement de développement

1.1. Google Colab

Google Colab est un service en nuage gratuit qui prend désormais en charge les GPU gratuits, permettant d'améliorer les compétences de codage en langage de programmation Python. De développer des applications d'apprentissage approfondi à l'aide de bibliothèques populaires telles que Keras, TensorFlow, PyTorch et Open CV.

La fonctionnalité la plus importante qui distingue Colab des autres services de cloud computing gratuits est la suivante : Colab fournit un GPU et est totalement gratuit.[W6]

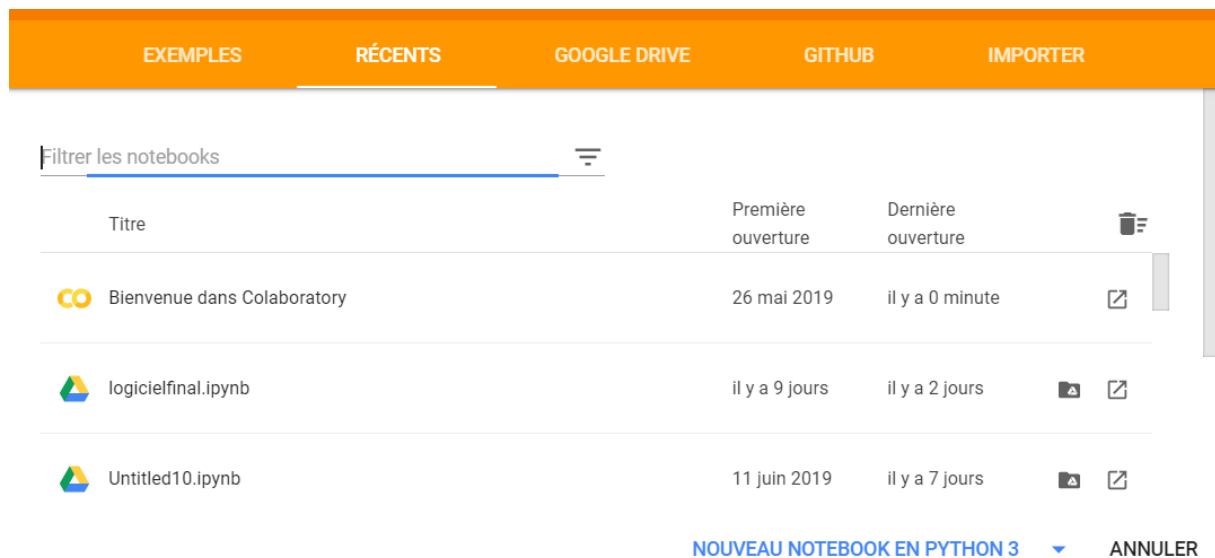


Figure 13. Environnement de Google Colab

1.2. Pourquoi les GPU ?

Pour la plupart des approches de l'apprentissage en machine ou l'apprentissage en profondeur, les GPU sont essentiels en raison de la quantité de données sur laquelle vous devez opérer. Le calcul mathématique de Deep Learning sur un CPU peut prendre des mois ! Mais ces calculs peuvent être assignés à des GPU pour un calcul plus rapide. Il n'est pas recommandé de former un réseau de neurones sur un processeur, un GPU est nécessaire pour un calcul élevé.

Comme nous le savons tous, les GPU sont chers, de même que les plates-formes cloud telles qu'AWS et GCP. Alors, où pouvons-nous obtenir un GPU ? Colab vient à la rescousse ! Colab vous fournit un **GPU Nvidia Tesla K80** gratuit dont le prix est d'environ 1,7 lakh Roupie et qui est une véritable bête dans la performance. Vous pouvez utiliser ce GPU pour former votre réseau de neurones sur le cloud et profiter des performances rapides et sans tracas. [W6]

1.3. Python

Python est un langage de programmation de haut niveau utilisé pour la programmation générale. Créé par Guido van Rossum et sorti en 1991, Python a une philosophie de conception qui met l'accent sur la lisibilité du code, notamment en utilisant des espaces importants.

Il fournit des constructions qui permettent une programmation claire à petite et à grande échelle. Python dispose d'un système de type dynamique et d'une gestion automatique de la mémoire. Il prend en charge de multiples paradigmes de programmation, y compris orientés objet, impératifs, fonctionnels et procéduraux, et dispose d'une bibliothèque standard vaste et complète. Les interpréteurs de Python sont disponibles pour de nombreux systèmes d'exploitation. [W7]



Figure 14. Le logo de python.

2. Bibliothèques utilisées

2.1. OpenCV (Open Source Computer Vision Library)

Est une bibliothèque proposant un ensemble de plus de 2500 algorithmes de vision par ordinateur spécialisé dans le traitement d'images, accessible au travers d'API pour les langages C, C++, et Python. Elle est distribuée sous une licence BSD (libre) pour les plateformes Windows, GNU/Linux, Android et MacOS [W8], nous avons utilisé cette bibliothèque pour la détection du visage à partir des images introduites.

2.2. Numpy :

Est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion facilitée des tableaux de nombres, des fonctions sophistiquées (diffusion), on peut aussi l'intégrer le code C / C ++ et Fortran. [W9][W10]

2.3. Matplotlib

est une bibliothèque de traçage pour le langage de programmation Python et son extension mathématique numérique NumPy . Il fournit une API orientée objet permettant d'incorporer des graphiques dans des applications à l'aide de kits d'outils d'interface graphique à usage général tels que Tkinter , wxPython , Qt ou GTK + .

2.4. Keras

Keras est une bibliothèque open source écrite en python et permettant d'interagir avec les algorithmes de réseaux de neurones profonds et de machine Learning, notamment Tensorflow et Theano. Elle a été initialement écrite par François Chollet .[W11]

2.5. Pandas

pandas est une bibliothèque open source sous licence BSD fournissant des structures de données hautes performances et faciles à utiliser, ainsi que des outils d'analyse des données pour le langage de programmation Python .

pandas est un projet sponsorisé par NumFOCUS . Cela contribuera au succès du développement de pandas en tant que projet open source de classe mondiale et permettra de faire un don au projet.[W12]

3. BDD

La base de données d'images des expressions faciales que nous avons eu à utiliser est celle de la base de données de Fer2013 elle contient 35887 images expressions faciales qui sont réparties en deux parties à savoir :

- 3589 images représentant les images de test
- Et 28709 représentent les images d'apprentissage.

Toutes ces images ont été déjà résolues par kaggle et contiennent chacune une taille de (48x48 pixels).

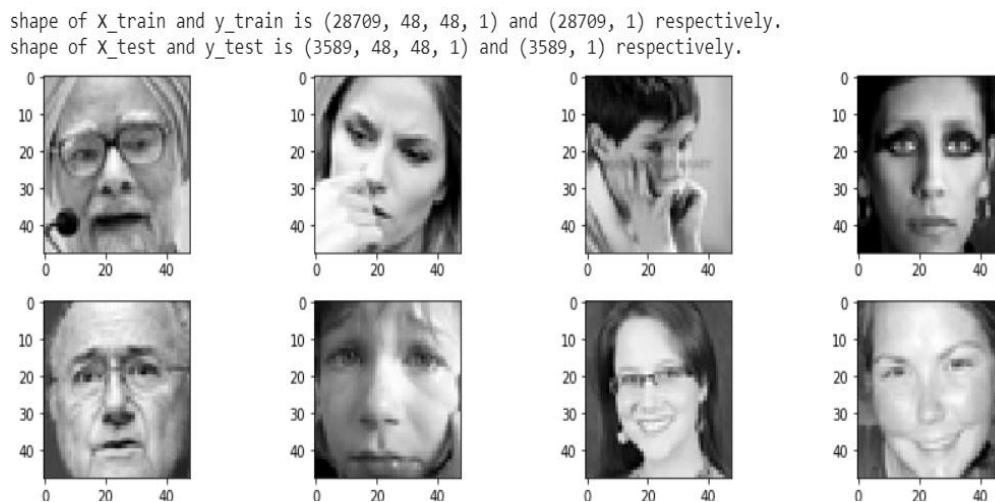


Figure 15.Échantillon d'images BDD de Fer2013.

4. Implémentation et résultats

Cette section décrit les expériences réalisées et les résultats expérimentaux de notre système sur la reconnaissance des expressions faciales. Elle contient deux parties pour examiner la performance de l'information.

Dans la **première expérience**, notre système est testé en utilisant une base de données fer2013 affichée ci-dessus.

Dans la **seconde expérience**, notre système est testé en utilisant des images externes afin de vérifier l'efficacité de notre système.

L'algorithme proposé est principalement réalisé en trois étapes comme ci-dessous :

Chapitre03 : Implémentation et résultat expérimentaux

- 1) Redimensionner les images d'entrée au format 32x32x3 ;
- 2) Construire une structure CNN avec trois couches constituées respectivement de : filtre convolutif, ReLu et de regroupement maximum (Max Pooling). Cette couche est répétée trois fois.
- 3) Après avoir extrait toutes les fonctionnalités, nous allons utiliser :

Flatten : elle permet d'aplatir les images voir la (figure12)

Dense : désigne le nombre que le veut sur une couche, elle prend toujours en entrée le résultat de l'image aplatit de la couche précédente.

Sofmax : qui est un classifieur qui désigne la distribution de la probabilité ou encore que la somme de toutes les porbalité de la sortie doit être égale à 1.

4.1. Formation du model avec keras

✚ La première des choses à faire consiste à tout d'abord configurer l'environnement

✚ Ensuite, installer toutes les bibliothèques citez ci-dessus

✚ Importer les bibliothèques installées

✚ Ensuite, nous allons importer le type de modèle séquentiel de Keras. Il s'agit simplement d'un empilement linéaire de couches de réseaux de neurones à travers la commande : **from keras. models import Sequential**

✚ Nous importerons les couches "principales" de Keras. Voici les couches utilisées dans presque tous les réseaux de neurones :

From keras. Layers import Dense, Dropout, Activation, Flatten

✚ Pour un entrainement plus efficace sur les données d'image nous allons importer les couches CNN de Keras. Ce sont les couches convolutives qui nous aideront à former notre modèle.**from keras. layers import Convolution2D, MaxPooling2D**

Notre modèle contient:

- (1, 48, 48) qui correspond à la (profondeur, largeur, hauteur) de chaque image de notre base de donnée de Fer2013 qui sera redimensionnée au format 32x32x3.
- Conv (3, 3) indique que nous avons utilisé une matrice à troisdimension

Initialization du model

```
model = Sequential ()
```

Adding Input Layer

```
model.add (Conv2D (32, (3, 3), padding='same', activation='relu',  
input_shape=input_shape))
```

Adding more layers

```
model.add (Conv2D (32, (3, 3), activation='relu'))
```

```
model.add (MaxPooling2D (pool_size= (2, 2)))
```

```
model.add (Dropout (0.25))
```

```
model.add (Conv2D (64, (3, 3), padding='same', activation='relu'))
```

```
model.add (Conv2D (64, (3, 3), activation='relu'))
```

```
model.add (MaxPooling2D (pool_size= (2, 2)))
```

```
model.add (Dropout (0.25))
```

```
model.add (Conv2D (128, (3, 3), padding='same',  
activation='relu'))
```

```
model.add (Conv2D (64, (3, 3), activation='relu'))
```

```
model.add (MaxPooling2D (pool_size= (2, 2)))
```

```
model.add (Dropout (0.25))
```

Flattening

```
model.add (Flatten ())
```

Adding fully connected layer

```
model.add (Dense (512, activation='relu'))
```

```
model.add (Dropout (0.4))
```

Adding Output Layer

2. Résultats expérimentaux :

Dans cette partie nous nous intéressons aux résultats d'exécution de l'approche développée.

Chapitre03 : Implémentation et résultat expérimentaux

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 32)	320
conv2d_2 (Conv2D)	(None, 46, 46, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 23, 23, 32)	0
dropout_1 (Dropout)	(None, 23, 23, 32)	0
conv2d_3 (Conv2D)	(None, 23, 23, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 11, 11, 64)	0
dropout_2 (Dropout)	(None, 11, 11, 64)	0
conv2d_4 (Conv2D)	(None, 11, 11, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 5, 5, 128)	0
dropout_3 (Dropout)	(None, 5, 5, 128)	0
flatten_1 (Flatten)	(None, 3200)	0
dense_1 (Dense)	(None, 512)	1638912
dropout_4 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 7)	3591

Total params: 1,744,423
Trainable params: 1,744,423
Non-trainable params: 0

Figure 17. résultat du modèle obtenu.

Une fois que le modèle est formé, nous allons essayer de lui attribuer certains paramètres :

Categorical_crossentropy : permet la réduction de l'erreur tout en utilisant l'algorithme de la descente de gradient.

Epoch : désigne le nombre d'itération dans notre base de données.

Optimiser : permet de réduire le poids des erreurs.

Loss : désigne le taux d'erreur.

Accuracy : désigne le taux de précision.

History : elle permet de donner au modèle toutes les images dont elle aura besoin en fonction du nombre d'itération (epochs) définit, dans le but de réduire le taux d'erreur.

```
batch_size = 256
epochs =200
model1.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accuracy'])
model1.summary()

history = model1.fit(train_data, train_labels_one_hot, batch_size=batch_size, epochs=epochs, verbose=1,
                    validation_data=(test_data, test_labels_one_hot))
model1.evaluate(test_data, test_labels_one_hot)
```

Figure 18.entrainement du modèle avec les images

Epoch 1/200

28709/28709 [=====] - 6s 192us/step - loss: 1.8165 -
acc: 0.2483 - val_loss: 1.7077 - val_acc: 0.2984

Epoch 200/200

28709/28709 [=====] - 4s 152us/step - loss: 0.3451 -
acc: 0.8903 - val_loss: 1.2742 - val_acc: 0.6124

Figure 19.Résultat de l'entrainement du modèle avec les images.

Si nous faisons une comparaison de la (**Figure19**) nous constatons qu'entre ces deux itérations nous constatons que le taux d'erreur baisse tant dis que notre précision augmente cela signifie que notre modèle a été bien entrainer et répons d'ailleurs à la définition du réseau de neurones qui disait que plus le réseau de neurones est profond meilleur sont ses performances.

```
#Defining labels
def get_label(argument):
    labels = {0:'Angry', 1:'Disgust', 2:'Fear', 3:'Happy',4:'Sad' , 5:'Surprise', 6:'Neutral'}
    return(labels.get(argument, "Invalid emotion"))

# resize plots
plt.figure(figsize=[6,4])

# Display random image from training data
plt.subplot(1,3,1)
plt.imshow(np.squeeze(X_train[54]), cmap='gray')
plt.title("entraînement : {}".format(get_label(int(y_train[54]))))

# Display random image from testing data
plt.subplot(1,3,3)
plt.imshow(np.squeeze(X_test[70]), cmap='gray')
plt.title("test: {}".format(get_label(int(y_test[70]))))
```

Figure 20. définissons les émotions de la BDD de test et d'entraînement.

La figure 21 désigne le résultat de la base de données et l'entraînement et phase de test.

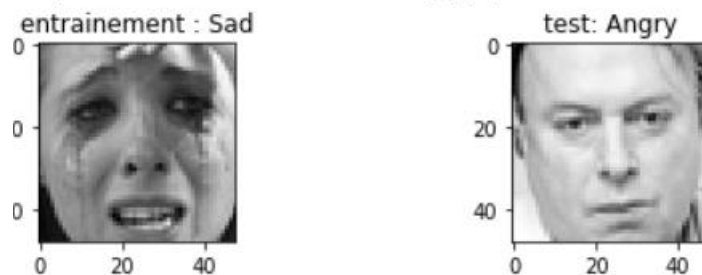


Figure 21. Les émotions obtenues de la BDD de test et d'entraînement.

Nous arrivons à la phase de quantification et valorisation des résultats de l'approche développée, nous se basant sur deux paramètres : le taux de reconnaissance (Accuracy) et le taux de perte ou erreur (Loss).

```
# Loss Curves
plt.figure(figsize=[8,6])
plt.plot(history.history['loss'], 'r', linewidth=3.0)
plt.plot(history.history['val_loss'], 'b', linewidth=3.0)
plt.legend(['Training loss', 'Validation Loss'], fontsize=18)
plt.xlabel('Epochs ', fontsize=16)
plt.ylabel('Loss', fontsize=16)
plt.title('Loss Curves', fontsize=16)
```

Figure 22. Affichage du Taux d'erreurs.

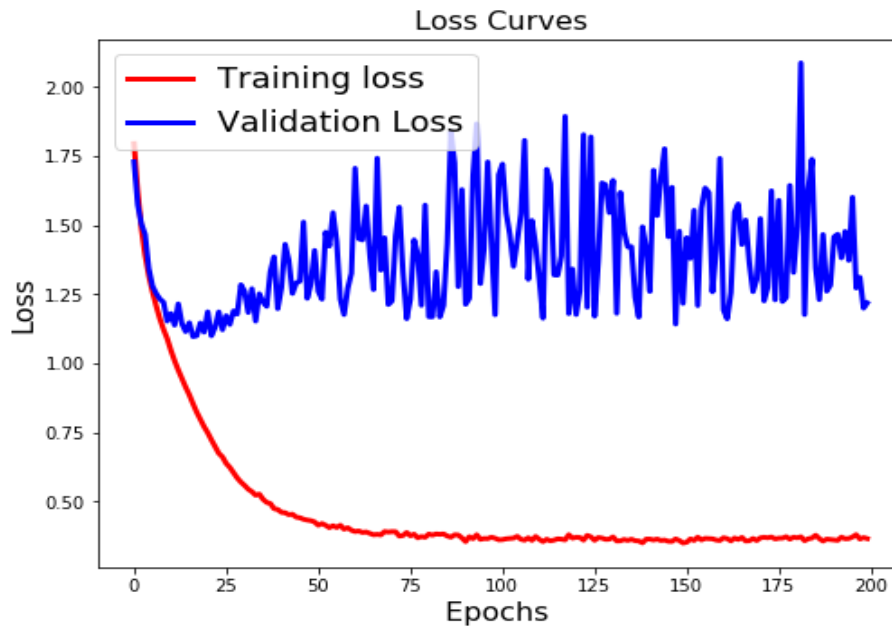


Figure 23.Taux d'erreurs

Les résultats obtenus sur le taux d'entraînement de la perte représenté dans le tableau ci-dessous nous ont donné **1.2742**

Et les résultats sur la perte nous ont donné un taux de **0.3451**

```
# Accuracy Curves
plt.figure(figsize=[8,6])
plt.plot(history.history['acc'],'r',linewidth=3.0)
plt.plot(history.history['val_acc'],'b',linewidth=3.0)
plt.legend(['Training Accuracy', 'Validation Accuracy'],fontsize=18)
plt.xlabel('Epochs ',fontsize=16)
plt.ylabel('Accuracy',fontsize=16)
plt.title('Accuracy Curves',fontsize=16)
```

Figure 24.Affichage du Taux de précision

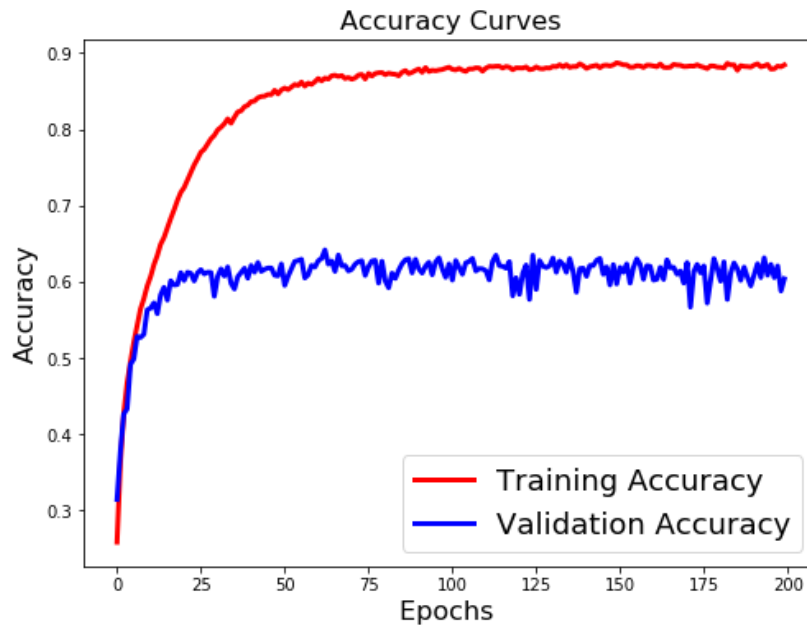


Figure 25. Le taux de précision.

Les résultats obtenus sur les données tests représentées dans le tableau ci-dessous nous ont donné un taux de **0.8903**.

Et les résultats sur les données d'entraînement nous ont donné un taux de **0.6124**.

Discussion : Les performances obtenues sont relativement acceptable compte tenu la taille réduite de l'échantillon d'images de la BDD et le fait qu'il s'agit d'images NG. Ceci s'explique par le fait que le CNN est plus performant dans le cas de BDD de grandes tailles. Toute fois vu la taille de la base de données ceci est tout de même acceptable même si on aurait voulu avoir beaucoup plus.

Comparaison aux travaux déjà existants :

Le tableau ci-dessous fait une comparaison des en termes du Training accuracy et la Validation accuracy avec la même base de données de Fer2013 nous constatons que les résultats obtenus avec les modèles CNN sont meilleurs.

Modele	Training accuracy	Validation accuracy
SVM	43.36%	38.61%
CNN	72.25%	63.86%
CNN	90.11%	67.68%
CNN	89,03%	61 ,24%

Tableau 4. Tableau comparatif de certains travaux déjà existant [26]

Deuxième expérience :

Pour mieux tester l'efficacité du modèle développé dans ce projet, nous allons essayer d'introduire des images externes. Pour cela nous avons créé un dossier et on a mis ces images dans le drive. Voir le code ci-dessous (Figure 26), ensuite nous avons fait appel au modèle déjà formé (Figure 27).

```
#Cropping image
from PIL import Image
import cv2
import matplotlib.pyplot as plt

faces = []

img = Image.open("/content/gdrive/My Drive/EmotionClassifier-master/trained_models/test_image/8.jpg")
face_cascade = cv2.CascadeClassifier('/content/gdrive/My Drive/Colab Notebooks/haarcascade_frontalface_alt.xml')
faces = face_cascade.detectMultiScale(np.asarray(img), 1.3, 5)
for (x, y, w, h) in faces:
    if len(faces) == 1:
        crop_img = img.crop((x,y,x+w,y+h))
    else:
        print("multiple faces detected, passing over image")

#Resizing image to required size and processing
test_image = crop_img.resize((48,48),Image.ANTIALIAS)

test_image = np.array(test_image)

gray = cv2.cvtColor(test_image, cv2.COLOR_BGR2GRAY)
```

Figure 26. Insertions des images externes.

Chapitre03 : Implémentation et résultat expérimentaux

```
#scale pixels values to lie between 0 and 1 because we did same to our train and test set

gray = gray/255

#reshaping image (-1 is used to automatically fit an integer at it's place to match dimension of original image)
gray = gray.reshape(-1, 48, 48, 1)

res = model1.predict(gray)

#argmax returns index of max value
result num = np.argmax(res)
```

Figure 27. Attribution des images au modèle.

Les résultats obtenus sont les figures ci-dessous.



Figure 28. Résultats de quelques images externes.

Conclusion

Nous avons présenté dans ce chapitre l'implémentation de l'approche de reconnaissance des expressions faciales basée sur les réseaux de neurones convolutionnels, L'approche proposée pour la reconnaissance des expressions faciales fonctionne pour de basses résolutions (les images sont de taille 32×32), intéressantes dans le cadre de futures extensions de la technique. L'approche utilise une architecture particulière de réseau de neurones convolutionnels. Celle-ci projette toutes les expressions du visage dans un espace de plus faible dimension, où la reconnaissance proprement dite est effectuée. Les tests effectués sur une base énormément grande étant encourageante.

Conclusion générale

Conclusion générale

Les démarches que nous avons eu à mener à savoir de la définition de la reconnaissance des expressions faciales jusqu'à l'implémentation de la méthode proposée nous prouvent que l'informatique est un outil de travail de masse. Durant cette phase de ce mémoire nous avons fait une généralité sur la reconnaissance des expressions faciales et du deep Learning,

La méthode que nous avons eu a utilisé est le réseau de neurones convolutifs, le modèle obtenu a été tester sur une très grande base de données qui est celle de Fer2013 le résultat obtenu nous a permis de faire une comparaison avec certains travaux déjà existants, qui nous a d'ailleurs donné de très bons résultats. Une des nouvelles encourageantes est que la plupart de ces progrès ne sont pas seulement le résultat d'un matériel plus puissant, de jeux de données plus volumineux et de modèles plus grands, mais principalement une conséquence de nouvelles idées, d'algorithmes et d'architectures réseau, améliorées.

Pour finir, avant de passer aux perspectives, ce travail nous a permis de mettre en pratique nos connaissances sur les réseaux de neurones et d'en acquérir d'autres et le temps passé à lire des articles nous a servi d'une bonne initiation à la recherche.

Perspectives :

Comme perspectives de recherches futures, nous envisageons de :

1. De tester sur notre modèle d'autre base de données autre que celle de Fer2013
2. D'augmenter de nouvelle couche afin de voir ce que cava donnée.
3. De travailler avec les modèles déjà entraînés tels que le VGG16 et faire une comparaison.

Bibliographie :

- [1] Fathallah, A., Abdi, L., & Douik, A. (2017, October). Facial expression recognition via deep learning. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)* (pp. 745-750)
- [2] Plutchik, R. (1980). A general psychoevolutionary theory of emotion. In *Theories of emotion* (pp. 3-33). Academic press
- [3] Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1), 71-86. 1991.
- [4] Padgett, C., & Cottrell, G. W. (1997). Representing face images for emotion classification. In *Advances in neural information processing systems* (pp. 894-900).
- [5] Lanitis, A., Taylor, C. J., & Cootes, T. F. (1997). Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and machine intelligence*, 19(7), 743-756.
- [6] Morizet, N. (2009). *Reconnaissance biométrique par fusion multimodale du visage et de l'iris* (Thèse de doctorat).
- [7] Brown, G. D., Yamada, S., & Sejnowski, T. J. (2001). Independent component analysis at the neural cocktail party. *Trends in neurosciences*, 24(1), 54-63.
- [8] Sagi, B., Nemat-Nasser, S. C., Kerr, R., Hayek, R., Downing, C., & Hecht-Nielsen, R. (2001). A biologically motivated solution to the cocktail party problem. *Neural Computation*, 13(7), 1575-1602.
- [9] Sagi, B., Nemat-Nasser, S. C., Kerr, R., Hayek, R., Downing, C., & Hecht-Nielsen, R. (2001). A biologically motivated solution to the cocktail party problem. *Neural Computation*, 13(7), 1575-1602
- [10] Kanade, T. (1973). *Picture processing system by computer complex and recognition of human faces*. Kyoto University, Japan (Thèse de doctorat).
- [11] Guo, G., Li, S. Z., & Chan, K. (2000, March). Face recognition by support vector machines. In *Proceedings fourth IEEE international conference on automatic face and gesture recognition (cat. no. PR00580)* (pp. 196-201).
- [12] Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6), 681-685.
- [13] Ahonen, T., Hadid, A., & Pietikäinen, M. (2004, May). Face recognition with local binary patterns. In *European conference on computer vision* (pp. 469-481). Springer, Berlin, Heidelberg.
- [14] Arca, S., Campadelli, P., & Lanzarotti, R. (2006). A face recognition system based on automatically determined facial fiducial points. *Pattern recognition*, 39(3), 432-443.

Références

- [15] Perlibakas, V. (2006). Face recognition using principal component analysis and log-gabor filters. *arXiv preprint cs/0605025*
- [16] Qi, C., Li, M., Wang, Q., Zhang, H., Xing, J., Gao, Z., & Zhang, H. (2018). Facial expressions recognition based on cognition and mapped binary patterns. *IEEE Access*, 6, 18795-18803.
- [17] Ghimire, D., Lee, J., Li, Z. N., & Jeong, S. (2017). Recognition of facial expressions based on salient geometric features and support vector machines. *Multimedia Tools and Applications*, 76(6), 7921-7946.
- [18] Lyons, M. J., Budynek, J., & Akamatsu, S. (1999). Automatic classification of single facial images. *IEEE transactions on pattern analysis and machine intelligence*, 21(12), 1357-1362.
- [19] Lyons, M., Akamatsu, S., Kamachi, M., & Gyoba, J. (1998, April). Coding facial expressions with gabor wavelets. In *Proceedings Third IEEE international conference on automatic face and gesture recognition* (pp. 200-205).
- [20] Valstar, M. F., Gunes, H., & Pantic, M. (2007, November). How to distinguish posed from spontaneous smiles using geometric features. In *Proceedings of the 9th international conference on Multimodal interfaces* (pp. 38-45). ACM.
- [21] Mlakar, U., Fister, I., Brest, J., & Potočnik, B. (2017). Multi-objective differential evolution for feature selection in facial expression recognition systems. *Expert Systems with Applications*, 89, 129-137.
- [22] Cohen, I., Sebe, N., Garg, A., Chen, L. S., & Huang, T. S. (2003). Facial expression recognition from video sequences: temporal and static modeling. *Computer Vision and image understanding*, 91(1-2), 160-187.
- [23] Davoine, F., Abboud, B., & Dang, V. M. (2004). Analyse de visages et d'expressions faciales par modèle actif d'apparence. *traitement du signal*, 1(3).
- [24] Pantic, M., & Rothkrantz, L. J. (2000). Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (12), 1424-1445.
- [25] Maalej, A., Amor, B. B., & Daoudi, M. (2011, June). Analyse locale de la forme 3D pour la reconnaissance d'expressions faciales.
- [26] M.Quinn, G.Sivesind, G.Reis," Real-time Emotion Recognition From Facial Expressions", Stanford University 2015.
- [27] Ding, H., Zhou, S. K., & Chellappa, R. (2017, May). Facenet2expnet: Regularizing a deep face recognition net for expression recognition. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)* (pp. 118-126).
- [28] Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3-4), 197-387.

Références

- [29] Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2(1), 1-127.
- [30] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1798-1828.
- [31] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.
- [32] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436.
- [33] Arel, I., Rose, D. C., & Karnowski, T. P. (2010). Deep machine learning-a new frontier in artificial intelligence research. *IEEE computational intelligence magazine*, 5(4), 13-18.
- [34] J. Schmid Huber, "Deep Learning,," Scholarpedia, vol. 10, no. 11, p. 32832, 2015.
- [35] Moualek, D. Y. (2017). *Deep Learning pour la classification des images* (Thèse de doctorat), Université Abou Bakr Belkaid Tlemcen.
- [36] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 38:1798–1828, 2013
- [37] Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., & Bengio, Y. (2017). Quantized neural networks: Training neural networks with low precision weights and activations. *The Journal of Machine Learning Research*, 18(1), 6869-6898.
- [38] Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In *Advances in neural information processing systems* (pp. 2643-2651).
- [39] Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning* (pp. 160-167).
- [40] Matsugu, M., Mori, K., Mitari, Y., & Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, 16(5-6), 555-559.
- [41] Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.
- [42] Liu, S., Yang, N., Li, M., & Zhou, M. (2014, June). A recursive recurrent neural network for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1491-1500).
- [43] Graves, A., & Jaitly, N. (2014, January). Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning* (pp. 1764-1772).

Références

- [44] Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3128-3137).
- [45] Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems* (pp. 841-848).
- [46] Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive science*, 9(1), 147-169.
- [47] Salakhutdinov, R., Mnih, A., & Hinton, G. (2007, June). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning* (pp. 791-798).
- [48] Hinton, G. E. (2012). A practical guide to training restricted Boltzmann machines. In *Neural networks: Tricks of the trade* (pp. 599-619). Springer, Berlin, Heidelberg.
- [49] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.
- [50] Salakhutdinov, R., & Hinton, G. (2009, April). Deep boltzmann machines. In *Artificial intelligence and statistics* (pp. 448-455).
- [51] Srivastava, N., & Salakhutdinov, R. R. (2012). Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems* (pp. 2222-2230).
- [52] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).
- [53] Denton, E. L., Chintala, S., & Fergus, R. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems* (pp. 1486-1494).
- [54] Zhang, R., Zhu, J. Y., Isola, P., Geng, X., Lin, A. S., Yu, T., & Efros, A. A. (2017). Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*.
- [55] Owens, A., Isola, P., McDermott, J., Torralba, A., Adelson, E. H., & Freeman, W. T. (2016). Visually indicated sounds. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2405-2413).
- [56] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- [57] Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014). Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2147-2154).

Références

- [58] Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.
- [59] Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3128-3137).
- [60] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- [61] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.
- [62] Bengio, Y., Laufer, E., Alain, G., & Yosinski, J. (2014, January). Deep generative stochastic networks trainable by backprop. In *International Conference on Machine Learning* (pp. 226-234).
- [63] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.
- [64] S. Raschka, Author of Python Machine Learning, researcher applying ML to computational bio .17 aout 2015 : What is the difference between deep learning and usual machine learning?
- [65] J. Da Rolt, PhD, Computer Engineer, Professor 26.12.2017: How is a convolutional neural network able to learn invariant features?
- [66] Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In *Advances in neural information processing systems* (pp. 2643-2651).
- [67] Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., & Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems* (pp. 2933-2941).
- [68] Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul), 2121-2159.
- [69] Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, Ng, A. Y. (2012). Large scale distributed deep networks. In *Advances in neural information processing systems* (pp. 1223-1231).
- [70] Clark, L. (2012). Google's artificial brain learns to find cat videos. *Wired UK*, www.wired.com.
- [71] Pennington, J., Socher, R., & Manning, C. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).
- [72] Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14, 8.

Webographie

- [W1] www.fr.wikipedia.org. Consultée le : 05 janvier 2019
- [W2] <https://www.futura-sciences.com/tech/dossiers/robotique-presentation-historique-reseaux-neuronaux-31/> consultée le : 15 janvier 2019
- [W3] www.image-net.org consultée le : 12 avril 2019
- [W4] www.wikiversity.org consultée le : 12 avril 2019
- [W5] <https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/c> consultée le : 25 Mai 2019
- [W6] <https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d> consultée le : 12 Mai 2019
- [W7] <https://www.python.org/> Consultée le 25/02/2019.
- [W8] <https://opencv.org/about/> consultée le : 12 avril 2019
- [W9] <http://www.numpy.org/> consultée 15 avril 2019
- [W10] <https://www.courspython.com/apprendre-numpy.html> 05 mai 2019. consultée 22 avril 2019
- [W11] <https://fr.wikipedia.org/wiki/Keras> consultée 20 avril 2019
- [W12] <https://pandas.pydata.org/> consultée 10 juillet 2019

