

Université A. MIRA-Bejaia
Faculté de Technologie
Département ATE

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Réseau et Télécommunication

Filière : Télécommunications

Spécialité: Réseaux et Télécommunications

Thème

*Reconnaissance des iris par des réseaux de
neurone convolutifs*

Présenté par :

LAHOUZI Dina

BOUHZILA Loubna

Soutenu 2022 devant le jury composé de :

Mme BOUNCER

M. DIBOUNE

Mme GHENNAM

Président

Encadrant

Examinatrice

Année universitaire: 2021/2022

Remerciement

Nous remercions tout d'abord le bon Dieu, le tout puissant de nous avoir accordées le savoir de nous avoir orientées vers le droit chemin.

Nous tenons à exprimer aussi nos reconnaissances à notre encadrant M. DIBOUNE, pour l'honneur qu'il nous a fait en assurant la direction et le suivi scientifique du présent mémoire.

Nous remercions aussi l'ingénieur d'informatique M.BEKHOCH pour son aide pratique et ses conseils.

Nous adressons également nos remerciements les plus profonds à nos enseignants qui nous ont enseignées tout au long de notre cursus universitaire.

Nous tenons aussi à remercier les membres du jury qui ont accepté d'examiner ce modeste travail.

Dédicace

Au nom de Dieu clément et miséricordieux

Je dédie ce modeste travail à :

Mes chères parentes : ma mère Nacera, mon père A/Aziz

Pour leurs soutiens, patiences et leurs

Sacrifices durant mes études et durant ce travail.

A mon frère Ameer

A ma petite sœur Malak

A mes chères grandes mères

A mon défunt oncle Mourad

Et à tous les membres de la famille

A ma chère amie Lydia

Pour leur soutien moral et sa sympathie

A mes chères amies : Ilhem, Yousra

Pour leur aide et leur support dans les moments difficiles.

Sans oublier ma binôme Loubna

Pour son soutien moral, sa patience tout au long de ce travail.

A tous mes professeures

A tous ceux qui ont contribué de près ou de loin à

La réalisation de ce travail.

Dina

Dédicace

Au nom de Dieu clément et miséricordieux

Je dédie ce modeste travail à :

Mon pilier et mon soutien et mon espoir dans cette vie

Qui est ma mère BAYA

Pour celui qui se fatigue et lutte la vie pour moi et pour mes études

Qui est mon père SAID

A mon Cher frère TAREK

AMa grand sœur, mon amour SOUMIA

A Ma petite sœur, mon âme AIDA

Mes grands-parents, Dieu les bénisse

A Mon fiancé SOFIANE qui a partagé ma fatigue

ET m'a toujours soutenu et ne m'a jamais laissé seul

Et à tous les membres de la famille

A mes tous chères amies

Pour leurs aide et leur support.

A mon binôme compréhensif Dina

Pour sa diligence et son soutien moral.

A tous mes professeures

A toutes les personnes qui m'avais aidé

Près ou de loin dans la réalisation de ce travail

Loubna

Résumé

Titre :Reconnaissance de l'iris par des réseaux des neurones convolutifs.

Mots-clés : la biométrie, l'iris, Deep Learning, Les Réseaux des neurones artificiels, Les réseaux des neurones covolutifs, classification, performance.

Résumer :

La biométrie est la science qui permet l'identification des individus à partir de leurs caractéristiques. La reconnaissance de l'iris est considérée comme la technique la plus précise, utilisée comme une forme unique d'identification. Il existe de nombreuses méthodes pour effectuer cette technique, parmi elles la méthode de réseaux de neurone convolutif qui représente une méthodologie similaire à celle des méthodes traditionnelles d'apprentissage. Cette méthode dépend sur deux grandes phases la première est la partie de convolution suivi par une partie de classification. Dans notre projet on va proposer un modèle CNN et étudier ces performances.

Abstract

Title: Iris recognition by convolutional neural networks.

Keywords:biometrics, iris, Deep Learning, Artificial Neural Networks, Covolutional Neural Networks,classification, performance.

Abstract:

Biometrics is the science that allows the identification of individuals from their characteristics. Iris recognition is considered the most accurate technique, used as a unique form of identification. There are many methods to perform this technique, among them the convolutional neural network method which represents a methodology similar to that of traditional learning methods. This method depends on two major phases the first is the convolution part followed by a classification part. In our project we will propose a CNN model and study these performances.

SOMMAIRE

Introduction générale	1
Chapitre 01 : Généralités sur la biométrie de l'iris	
Introduction.....	2
I.1. La biométrie.....	2
I.2. Les domaines d'applications de la biométrie	3
I.2.1. Applications commerciales.....	3
I.2.2. Applications gouvernementales.....	3
I.2.3. Applications légales	3
I.3. Le système biométrique.....	3
I.4. Les modalités biométriques	3
I.4.1. Biométrie physique.....	3
I.4.1.1. Empreintes Digitales.....	3
I.4.1.2. Visage	4
I.4.1.3. Empreinte palmaire.....	4
I.4.1.4. Iris.....	4
I.4.2. Biométrie comportementale	4
I.4.2.1. Voix	4
I.4.2.2. Signature manuscrite	4
I.4.2.3. Démarche.....	4
I.4.2.4. Frappe dynamique sur le clavier.....	5
I.4.3. Biométrie biologique :	5
I.4.3.1. Forme des veines de la main.....	5
I.4.3.2. Analyse de l'ADN	5
I.5. Facteurs à prendre en considération lors du choix d'une caractéristique biométrique pour une identification biométrique	5
I.5.1. L'universalité.....	5
I.5.2. L'unicité	5
I.5.3. La pérennité.....	5
I.5.4. Mesurabilité.....	5
I.5.5. La performance	5

1.5.6. L'acceptabilité	6
I.5.7. Le contournement.....	6
I.6. L'identification par l'iris	6
I.7. Principe général de reconnaissance de l'iris	7
I.8. Acquisition d'iris	7
I.9. Système de reconnaissance de l'iris :	8
I.9.1. Segmentations de l'iris	8
I.9.2. Normalisation.....	8
I.9.2.1. Inclinaison de la tête par le sujet	9
I.9.2.2. Mouvement du globe oculaire	9
I.9.2.3. Paramètre d'éclairage	9
I.9.2.4. Nature non concentrique de l'iris du sujet.....	10
I.9.2.5. Le flou d'image.....	10
I.9.3. Extraction des caractéristiques de l'iris	10
I.9.4. Correspondance de modèle :.....	10
I.9.4.1. Mode authentification.....	10
I.9.4.2. Mode identification.....	11
I.10. Technique de reconnaissance de l'iris utilisé	11
I.10.1. Modèle du SVM	11
I.10.2. Modèle du KNN	11
I.10.3. Un réseau de neurones	12
I.10.4. Modèle du Daugman	13
Conclusion	13

Chapitre 02:Architecture d'un réseau de neurones convolutif

Introduction.....	14
II.1. Les Réseaux de neurones artificiels	14
II.1.1. Le perceptron.....	15
II.1.2. Les Perceptrons Multicouches (MLP).....	15
II.1.3.Apprentissage de réseau de neurones	16
II.1.3.1. Apprentissage supervisé	16
II.1.3.2. Apprentissage non supervisé	16
II.1.4. Algorithme d'optimisation des paramètres dans un réseau de neurones	16

II.1.4.1. Initialisation des poids.....	16
II.1.4.2. L’admission des données	16
II.1.4.3. Propagation avant.....	17
II.1.4.4. Fonction de coût	18
II.1.4.5. Rétro propagation.....	18
II.1.4.6. Gradient descente	18
II.1.4.7. Mise à jour des poids.....	19
II.2. Les réseaux de neurones convolutifs.....	19
II.2.1. Architecture d’un réseau de neurones convolutif.....	19
II.2.1.1. Couche de convolution.....	20
II.2.1.2. Couche de mise en commun (pooling).....	22
II.2.1.3. Couche de vectorisation	23
II.2.1.4. Couche de classification.....	23
Conclusion	23

Chapitre 03:evaluation des performances d'un modèle CNN

Introduction.....	24
III.1. Modèle du réseau de neurones convolutif utilisé.....	24
III.2. Évaluations du modèle	27
III.2.1. Définition de la matrice de confusion.....	27
III.2.2. Les Métriques.....	27
III.3. L’environnement de développement.....	28
III.3.1. La Base de données utilisée	28
III.4. Évaluation des performances	29
III .4.1. Notions fondamentales.....	29
III.4.2. Étude des performances du modèle propose.....	29
III.4.2.1 Relation entre le nombre de couches de la convolution, la vitesse d’apprentissage et la précision:.....	29
III.4.3.Étude des métriques	35
III.4.4. Relation entre le nombre de couches de convolution est la précision	36
Conclusion	37
Conclusion Générale	39

Liste des Figures

Figure I.1: Anatomie et différentes caractéristiques de l'iris.....	7
Figure I.2 : Étapes typiques de la reconnaissance de l'iris.....	8
Figure I.3: Le bruit de prise partielle d'image de l'iris.....	9
Figure I.4: Exemple de rotation de l'iris.....	9
Figure I.5: le bruit de réflexion d'éclairage.....	9
Figure I.6: Exemple d'image floue.....	10
Figure I.7 : machine à vecteurs de support.....	11
Figure I.8: méthode des K plus proches vois.....	12
Figure I.9: réseaux neurones artificielles.....	13
Figure II.1 : la déférente couche de réseaux neurone artificiel.....	15
Figure II.2 : modèle de perceptron.....	16
Figure II.3 : l'architecture générale d'un réseau de neurones convolutif.....	21
Figure II.4: illustration de convolution.....	21
Figure II.5 : effet d'ajoutassions ou suppression de padding dans la convolution.....	22
Figure II.6: exemple de convolution en 3D.....	23
Figure II.7: Illustration de l'étape de sous échantillonnage ("pooling").....	23
Figure III.1: architecture du modèle CNN proposé.....	28
Figure III.2 : Précision et perte pour validation et Mini-Batch en utilisant une architecture à une couche de convolution.....	32
Figure III.3 : Précision et perte pour validation et Mini-Batch en utilisant une architecture à deux couches de convolution.....	33
Figure III.4 : Précision et perte pour validation et Mini-Batch en utilisant une architecture à trois couches de convolution.....	34

Figure III 5 : Précision et perte pour validation et Mini-Batch en utilisant une architecture à quatre couches de convolution.....35

Figure III 6 : Précision et perte pour validation et Mini-Batch en utilisant une architecture à cinq couches de convolution.....36

Figure III7 : comparaison des méthodes SVM et KNN et DecisionTree selon le nombre de couche de convolution.....39

Liste des Abréviations

Abréviation	Libelléenanglais	Libelléenfrançais
ADN	deoxyribonucleic acid	AcideDésoxyriboNucléique
ISO	International Organisation for Standardisation	Organisation International de Normalisation
SVM	Support Vector Machine	Machine à Vecteurs de Support
KNN	K-Nearest Neighbors.	/
ANN	ArtificielNeurone Network	Reseaux de NeuroneArtificiel
MLP	Multi-layer Perceptron	PerceptronsMulticouches
RNN	Recurrent Neural Network	RéseauNeuroneRécurant
CNN	ConvolutionalNeural Network	Réseau de NeuronesCovolutifs
ReLu	RectifiedLinear Unit	UnitéLinéaireRectifiée
GAN	Generativeadversarial networks	Réseauxadversariauxgénératifs
LSTM	Long-term and short-term memory networks	Réseaux de mémoire à long terme et à court terme

Liste des Tableaux

Tableau I.1: Comparaison des modalités biométriques.....	6
Tableau III .1: Les caractéristiques du modèle CNN proposé.....	27
Tableau III.2: Résultats des métriques de performances du modèle CNN proposé en utilisant les classifieurs SVM, KNN, Naif Bayes et DecisionTree.....	37
Tabelau III 3: La précision du modèle CNN proposé en le liant aux classifieurs traditionnels SVM , KNN Et DecisionTree dépendamment du nombre de couches de convolution.....	38

Introduction Générale

Introduction générale

Nous vivons actuellement une véritable révolution d'accès à l'information, dans tous les domaines de l'activité humaine. En fait, la sécurité des systèmes d'information est devenue un domaine de recherche d'une très grande importance, l'identification de l'individu est essentielle pour assurer la sécurité des systèmes et organisations. Dans ce sens, la biométrie est un exemple pratique parce qu'elle est de plus en plus présente dans la vie quotidienne: au travail des opérations bancaires, l'accès à certains endroits militaires ou industriels.

Un système de contrôle biométriques est un système automatique de mesure basé sur la reconnaissance des caractéristiques propres à un individu. Il existe différents moyens permettant la reconnaissance d'individus comme l'iris, elle s'agit de l'un des procédés biométriques les plus fiables, en raison d'un niveau d'unicité élevé.

Il existe plusieurs techniques de la reconnaissance d'iris nous citons (apprentissage automatique, partitionnement des graphes, etc.) : la méthode de SVM, la méthode de KNN, Algorithme de Daugman et les réseaux de neurones convolutifs.

La technique de Réseau de neurones convolutifs (Convolutional Neural Network) analyse les images et les vidéos afin de résoudre des problèmes complexes d'apprentissage, elle est avantageuse par rapport à les autres techniques classiques vue qu'elle extrait des caractéristiques propres à chaque image.

L'objectif principal de notre projet de fin d'étude, est l'étude des performances d'un modèle CNN pour la reconnaissance d'iris utilisant des différents classifieurs afin de savoir lesquels sont les plus performants.

Lorsque l'on fait de la reconnaissance d'iris par apprentissage profond, le plus important c'est d'avoir une grande base de données, pour cela on 'a utilisé la base de données IITD.

Pour ce faire, nous avons structuré notre mémoire en trois chapitres :

Le premier chapitre se repose sur la définition des systèmes et technologies biométriques d'une manière générale, et la biométrie par Iris en particulier. Aussi il définit les méthodes de segmentation et prétraitement utilisées sur les images Iris afin d'obtenir une base de données normalisées.

Le deuxième chapitre est consacré à la description de réseau de neurones et leurs paramètres d'optimisation, ainsi le chapitre décrit les réseaux de neurones convolutif et ses différentes couches.

Dans le troisième chapitre, nous allons montrer la partie expérimentale de notre travail, avec une évaluation des performances des différents classifieurs utilisés et on discute les résultats obtenus.

On termine par une conclusion générale

Chapitre 01

Généralités sur la biométrie de l'iris

Introduction

La biométrie est une technologie en pleine croissance qui est de plus en plus utilisée dans notre vie quotidienne. Elle vise à établir l'identité d'une personne de manière aussi fiable que possible en utilisant ses caractéristiques biologiques afin de garantir la sécurité des personnes dans les lieux publics [1].

La biométrie spécifie par des modalités qui sont les caractéristiques humaines physiques ou comportementales.

La reconnaissance de l'iris est considérée comme la technique la plus précise, utilisée comme une forme unique d'identification pour permettre d'identifier un individu, les caractéristiques morphologiques de l'iris sont propres à chaque individu et stables dans le temps [2].

Ce présent chapitre est consacré à la présentation de la biométrie de l'iris, nous donnons une description du système de reconnaissance d'iris en montrant l'acquisition et les différentes méthodes utilisées pour le traitement des données de la biométrie d'iris.

I.1. La biométrie

Le mot biométrie signifie "mesure du vivant", c'est la science permettant l'identification de l'individu à partir de leurs caractéristiques physiologiques ou comportementales. Ces caractéristiques doivent être universelles, uniques, enregistrables, et mesurables. Le premier exemple de la biométrie est l'enregistrement d'empreintes digitales par l'explorateur européen, au cours de XXème siècle, la biométrie a été utilisée exclusivement dans l'étude quantitative des êtres vivants [2].

En 1947 la formation de la Société internationale de Biométrie (Biometric Society) [3]. Durant les années soixante le développement des procédures de signature d'authentification. En 1970 l'utilisation de la biométrie est devenue systématique dans le domaine de détection des criminels.

La biométrie désigne la technique qui permet d'associer une identité à un individu grâce à la reconnaissance automatique d'une ou de plusieurs caractéristiques physiques ou comportementales de cette personne, qui sont préalablement enregistrées dans une base de données [4].

La question qui se pose est : pourquoi utiliser la biométrie ?

Traditionnellement, pour vérifier l'identité, on utilise surtout des codes, mots de passe, badges, cartes à puce et autres qui présentent plusieurs inconvénients :

- Il faut les mémoriser.
- Il y a un risque d'utilisation par des personnes non autorisées.

L'avantage de l'identification biométrique est que chaque individu a ses propres caractéristiques physiques qui ne peuvent être changées, perdues ou volées [5].

La méthode d'identification biométrique peut aussi être utilisée en complément ou en remplacement [6] des techniques classiques.

I.2. Les domaines d'applications de la biométrie

Le champ d'application de la biométrie est très vaste. En effet, tous les domaines qui nécessitent de vérifier ou déterminer l'identité d'une personne sont concernés, d'où les applications de la biométrie peuvent être divisées en trois groupes principaux :

I.2.1. Applications commerciales

Telles que l'ouverture d'un réseau informatique, la sécurité des données électroniques, l'e-commerce, l'accès Internet, les cartes de crédit, le contrôle d'accès physique, le téléphone cellulaire, la gestion des registres médicaux, E-learning...etc [7].

I.2.2. Applications gouvernementales

Telles que la carte d'identité nationale, le permis de conduire, la sécurité sociale, le contrôle des frontières, le contrôle des passeports...etc [7].

I.2.3. Applications légales

Telles que l'identification de corps, la recherche criminelle, l'identification de terroristes...etc[7].

I.3. Le système biométrique

Un système biométrique est un système de reconnaissance des formes et des caractéristiques propres à l'individu, qui fonctionne sur l'analyse des données prises sur un être humain et comparant ces caractéristiques avec des données stockées sur une base de données selon le contexte d'application.

Il existe plusieurs types d'applications qui nécessitent d'identifier l'utilisateur. Ces applications peuvent être séparées en deux catégories qui fonctionnent en mode de vérification ou mode d'identification.

I.4. Les modalités biométriques

La biométrie est basée sur les caractéristiques biométriques de l'individu, ces caractéristiques sont classées en trois grandes catégories :

I.4.1. Biométrie physique

La biométrie physique fait référence aux caractéristiques physiologiques du corps humain qui peuvent servir d'identification, comme une empreinte digitale ou une analyse de la rétine [8].

I.4.1.1. Empreintes Digitales

La reconnaissance des individus par empreintes digitales est la technique biométrique la plus utilisée. Les empreintes digitales sont composées de lignes localement parallèles présentant des points singuliers (minuties) et constituent un motif unique, universel et permanent [9], ce type de

technique biométrique est utilisé par les institutions financières pour leurs clients et se trouve en même temps dans les hôpitaux, les écoles, les aéroports...etc[10] [11].

I.4.1.2. Visage

Nos visages sont des objets complexes avec des traits qui peuvent varier dans le temps. L'écart entre les deux yeux, l'écartement des narines ou encore la largeur de la bouche peuvent permettre d'identifier un individu. Cette méthode doit pouvoir tenir compte de certains changements de la physionomie (lunettes, barbe, chirurgie esthétique) et de l'environnement (conditions d'éclairage). Parfois, il est impossible de différencier deux jumeaux [12] [13].

I.4.1.3. Empreinte palmaire

Cette technique utilise la surface intérieure de la paume pour l'identification et/ou la vérification des personnes. Elle est bien adaptée pour les systèmes à moyenne sécurité telle que le contrôle d'accès physique ou logique [14].

I.4.1.4. Iris

L'iris est une région sous forme d'anneau, située entre la pupille et le blanc de l'œil, elle est unique. L'iris a une structure extraordinaire et offre de nombreuses caractéristiques de texture qui sont uniques pour chaque individu [12].

I.4.2. Biométrie comportementale

I.4.2.1. Voix

La voix humaine varie d'une personne à une autre et peut se constituer de composantes physiologiques et comportementales. L'identification par la voix est sur les traits acoustiques tels que la fréquence fondamentale, le spectre de fréquence, le timbre, le pitch, l'intensité et utilisée dans la synthèse du son [15]. La reconnaissance des locuteurs est plus utilisée par les téléphones, Les corps policiers, les hôpitaux, etc.

I.4.2.2. Signature manuscrite

C'est une écriture personnelle d'un individu. Basée sur deux modes

- Mode statique: la vérification de la signature statique met l'accent sur les formes géométriques de la signature, dans ce mode, en général, la signature est normalisée à une taille connue ensuite décomposée en éléments simples [16].
- Mode dynamique: il utilise les caractéristiques dynamiques telles que l'accélération, la vitesse et les profils de trajectoire de la signature [16].

I.4.2.3. Démarche

Chaque personne a une façon particulière de marche, nous pouvons identifier les individus à partir de la nature du mouvement des jambes, des bras et des articulations ou le mouvement spécial obtenu par une caméra vidéo afin de l'envoyer à un ordinateur pour l'analyse afin de déterminer la vitesse et l'accélération de chaque individu[15].

I.4.2.4. Frappe dynamique sur le clavier

C'est un système de reconnaissance d'un individu basé sur la manière de ses écritures par un dispositif logiciel qui calcule la vitesse de la frappe, la suite des lettres, le temps de frappe et la pause entre chaque mot [15].

I.4.3. Biométrie biologique :

I.4.3.1. Forme des veines de la main

Les veines de la main sont uniques pour chaque individu, il s'agit ici d'analyser le dessin formé par le réseau des veines pour garder quelques points caractéristiques [2].

I.4.3.2. Analyse de l'ADN

L'ADN présent dans les cellules du corps, il est spécifique pour chaque individu et permet de l'identifier de manière certaine à partir d'un simple fragment de peau, d'une trace de sang ou d'un cheveu ou d'ongle. On l'utilise pour identification des criminels ou reconnaître des corps anonymes. Le résultat d'un test ADN n'est pas immédiat dans les délais de 10 à 14 jours avec un coût cher mais exact et sûr à 100%. Il existe des nouvelles façons d'analyses des caractéristiques d'un individu, nous citons l'odeur, la salive, l'urine [2].

I.5. Facteurs à prendre en considération lors du choix d'une caractéristique biométrique pour une identification biométrique

On distingue sept particularités pour chaque attribut biométrique permettant de les comparer les uns aux autres et ainsi déterminer les plus appropriés à une situation donnée [17]:

I.5.1. L'universalité

Détermine si la modalité existe et si elle est présente quel que soit l'individu.

I.5.2. L'unicité

Définit la probabilité de ne pas trouver de similitudes entre les mesures d'une même modalité sur des personnes différentes.

I.5.3. La pérennité

Indique si la caractéristique reste inchangée sur une période donnée.

I.5.4. Mesurabilité

Détermine le degré de facilité de l'acquisition, de la mesure et de l'exploitation de la modalité.

I.5.5. La performance

Caractérise la robustesse, la fiabilité et la vitesse de la mesure.

1.5.6. L'acceptabilité

Indique si la modalité est acceptée sans objection par l'utilisateur.

1.5.7. Le contournement

Représente la difficulté de contourner le système, par usurpation d'identité ou d'autres techniques de fraude.

Tableau I.1: Comparaison des modalités biométriques
 (***) le facteur le plus approprié.

1.6. L'identification par l'iris

L'iris est une région sous forme d'anneau, situé entre la pupille et le blanc de l'œil avec des

Modalité	Universalité	Unicité	pérennité	Mesurabilité	Performance	Acceptabilité	contournement
Visage	***	*	**	***	*	***	*
Empreintes digitales	**	***	***	**	***	**	***
Paume de la main	**	***	***	**	***	**	***
Réseau vasculaire	**	**	**	**	**	**	***
Géométrie de la main	**	**	**	***	**	**	**
Rétine	***	***	**	*	***	*	***
Iris	***	***	***	**	***	*	***
Oreille	**	**	***	**	**	***	**
Démarche	**	*	*	***	*	***	**
Signature	*	*	*	***	*	***	*
Frappe au clavier	*	*	*	**	*	**	**
Voix	**	*	*	**	*	***	*

motifs uniques : sillons, des anneaux, des taches et cryptes. L'iris a une couleur distincte, immuable et invariante dans le temps.

L'identification des personnes par l'iris a le meilleur potentiel de développement et un avenir radieux en raison de ses avantages qui sont son invariabilité, sa stabilité, et son accessibilité, même si son l'histoire n'est que d'environ 20 ans, la biométrie de l'iris est particulièrement avantageuse pour une utilisation dans les systèmes de reconnaissance biométrique.

L'iris est connu pour être particulièrement stable tout au long de la vie d'une personne. Les motifs de l'iris de l'homme sont fixés à environ une année d'âge et restent constants [19].

L'iris et la rétine ont démontré qu'ils avaient un degré élevé de caractères distinctifs encore plus que la main ou la géométrie du doigt. Les motifs de l'iris ne sont pas déterminés par la

génétique et ils sont tellement aléatoires, les motifs de l'iris de l'œil gauche et droit sont aussi différents les uns des autres que l'iris d'un individu à un autre individu. Même les jumeaux identiques ont des caractéristiques de l'iris complètement différents (Il en existe plus de 266 caractéristiques).

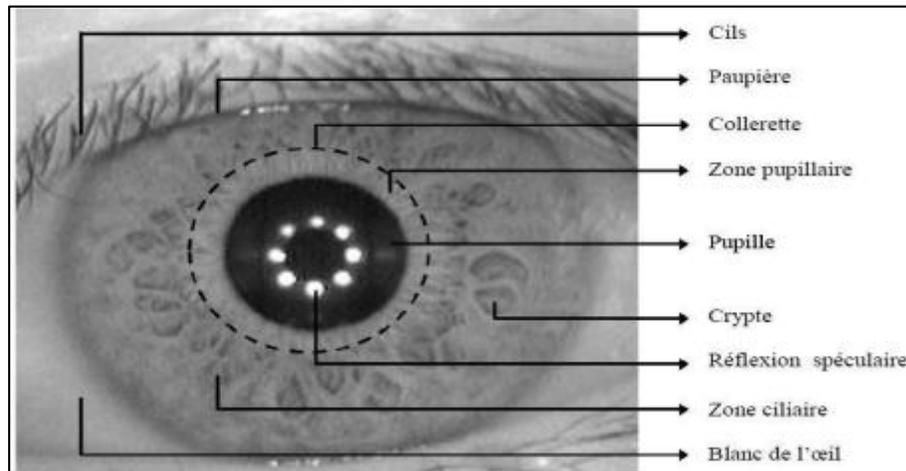


Figure I.1: Différentes caractéristiques de l'iris humain[19].

I.7. Principe général de reconnaissance de l'iris

Le processus de système de reconnaissance de l'iris contient deux phases principales :

- Une unité de vision optique pour capturer l'image de l'iris.
- Une unité de traitement des données qui contient : segmentation de l'image de l'iris, normalisation, extraction des caractéristiques de l'iris et correspondance de modèle.

I.8. Acquisition d'iris

Les images peuvent être recueillies à partir de deux types de caméras : la caméra traditionnelle et la caméra infrarouge. La caméra traditionnelle capte les longueurs d'onde du spectre visible que l'œil humain sait reconnaître. La caméra infrarouge de son côté, saisit les ondes infrarouges dans la lumière naturelle. Avec ce type de caméra, la luminosité des images produites est médiocre de telle sorte que l'on doit les soumettre à une source infrarouge supplémentaire afin d'améliorer la qualité [2].

L'acquisition des images d'iris est ainsi implémentée d'une manière non-invasive par un système de photographie [20]. Les systèmes d'acquisitions des images d'iris qui se conforment à la norme ISO utilisent une ou plusieurs sources infrarouges pour éclairer l'iris [21].

Pour capturer l'image d'un iris, la caméra numérique doit être fixée à quelque distance de l'utilisateur (de 30 cm à 60 cm), et acquérir directement l'image. L'iris étant un organe sensible, de petite taille et il est obscurci par les cils, les paupières ou les lentilles. De plus, les utilisateurs ont tendance à bouger. Il est donc assez difficile d'avoir une bonne qualité de l'image, donc il faut que ce soit rapide et précis.

En l'absence de conditions contrôlées, différentes irrégularités peuvent se présenter dans l'image de l'iris tel que l'obtention d'une image floue, de mauvais contraste, la présence des reflets et l'éclairage sur la surface de l'iris.

I.9. Système de reconnaissance de l'iris :

Après avoir une image de l'iris acquise, on doit passer au traitement des données obtenues par l'acquisition, ce système composé de plusieurs phases comprenant:

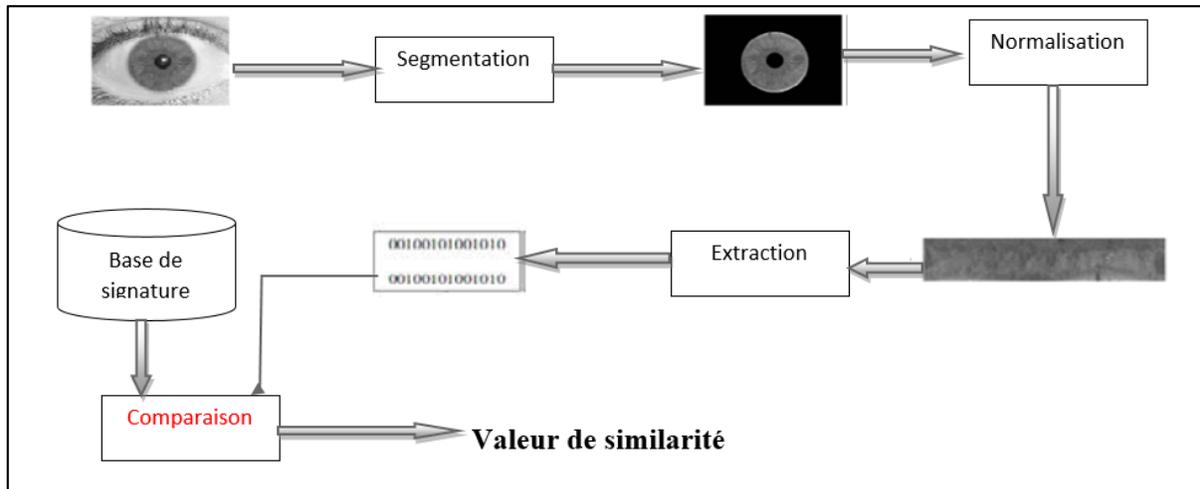


Figure I.2 : Étapes typiques de la reconnaissance de l'iris.

I.9.1. Segmentations de l'iris

La segmentation de l'iris est la première étape dans le processus de traitement, avec la présence d'autres informations dans l'œil humain, il est nécessaire d'isoler la zone de l'iris du blanc et des paupières de l'œil, la réalisation de cette étape dépend d'un algorithme pour détecter la frontière iris/pupille et frontière d'iris/sclérotique.

Elle est considérée comme l'étape la plus difficile du système de reconnaissance et son degré de fiabilité affecte très significativement la performance du système [21].

I.9.2. Normalisation

L'iris segmenté peut être un iris complet ou une fraction d'iris complet, de petite taille ou de grande taille. Ainsi une comparaison entre deux iris de tailles différentes ne sera possible qu'à partir d'une représentation consistante entre toutes les images. Les opérations à appliquer doivent garantir la même dimensionnalité de tous les iris [22].

Les causes des incohérences dans les dimensions de la région de l'iris :

I.9.2.1. Inclinaison de la tête par le sujet

La capture d'images de la tête lors de mouvement du corps peut produire un manque de quantité d'informations comme c'est montré dans la Figure I.3 [2].



Figure I.3: Le bruit de prise partielle d'image de l'iris [2].

I.9.2.2. Mouvement du globe oculaire

En raison de la rotation de la tête et des yeux, il est possible de capturer des images qui ne sont pas alignées avec la direction du capteur [2].



Figure I.4: Exemple de rotation de l'iris [2].

I.9.2.3. Paramètre d'éclairage

Ce type de bruit correspond généralement à des réflexions des sources de lumière artificielle à proximité de l'objet ou bien dans les conditions d'éclairage naturel d'environnement, ils peuvent apparaître avec une large gamme et localisées dans des parties distinctes de l'iris, ces zones ont des valeurs d'intensité proche du maximum [2].

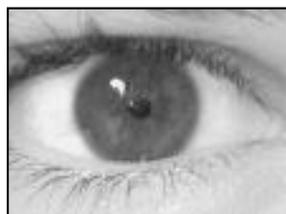


Figure I.5: Le bruit de réflexion d'éclairage [2].

I.9.2.4. Nature non concentrique de l'iris du sujet

Ce bruit apparaît lorsque la segmentation de frontières de pupille/iris et iris / sclérotique n'est pas exacte, certaines parties de pupille et sclérotique sont considérées comme un bruit dans l'iris[2].

I.9.2.5. Le flou d'image

Une fois de plus, plusieurs pièces mobiles qui interagissent lors de la capture de l'image, produisent le flou [2].

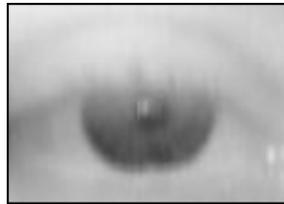


Figure I.6: Exemple d'image floue [2].

I.9.3.Extraction des caractéristiques de l'iris

Dans cette étape, l'iris est bien segmenté et normalisé. L'encodage consiste à extraire de l'iris les caractéristiques les plus discriminantes et les plus pertinentes, nécessaires et utiles pour son identification ou authentification [22] [23].

Un modèle d'iris a près de 266 caractéristiques dans ses modèles, 173 sont utilisées pour la création de modèle.

I.9.4.Correspondance de modèle :

Pour que le système de reconnaissance d'iris soit utilisé pour révéler l'identité, une phase d'entraînement doit précéder l'utilisation du système. Dans cette phase, une base de données contenant une signature de l'iris spécifique pour chacune des personnes à reconnaître est constituée.

Le processus revient à attribuer à chaque personne un profil d'iris, qui sera utilisé lors d'une comparaison avec un nouveau profil d'iris d'identité inconnue [2]. La correspondance des modèles est appariée à l'aide de plusieurs techniques.

Un système de reconnaissance d'iris peut fonctionner selon deux modes :

I.9.4.1. Mode authentification

Il va comparer les informations des utilisateurs autorisés stockées dans une base de données (en local ou sur un serveur d'authentification) à celles fournies. L'accès sera autorisé seulement si les informations sont identiques [24].

I.9.4.2. Mode identification

Permet de connaître l'identité d'une entité. Le vérificateur vérifie l'information révélée par l'entité par rapport à celles de toutes les entités connues. L'identification doit être unique.

I.10. Technique de reconnaissance de l'iris utilisé

Il existe plusieurs techniques de la reconnaissance d'iris nous citons :

I.10.1. Modèle du SVM

Le SVM permet de résoudre les problèmes de classification ou de régression (prédiction d'une valeur numérique). Le principe de base consiste à déterminer une fonction h qui permet de correspondre pour chaque vecteur d'entrée x (un échantillon, un vecteur décrivant les valeurs des caractéristiques de l'iris) à une sortie $y \in \{-1, +1\}$ l'iris de la personne à identifier ou pas. Dans le cas linéaire, $h(x) = W^T x + W_0$ le but est d'apprendre W, W_0 du meilleur séparateur, grâce aux échantillons étiquetés dans l'ensemble d'apprentissage (training dataset): $\{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\} \subset R^n \times \{-1, +1\}$. Sachant que la marge est la distance entre l'hyperplan $x \mapsto h(x) = w^T x + w_0$ et les vecteurs du support (les échantillons les plus proches, voir Figure I.7). Le but principal est de chercher l'hyperplan qui permet de maximiser cette marge [25].

$$\operatorname{argmax}_{w, w_0} \min_k \{\|x - x_k\| : x \in R^n, w^T x + w_0 = 0\}.$$

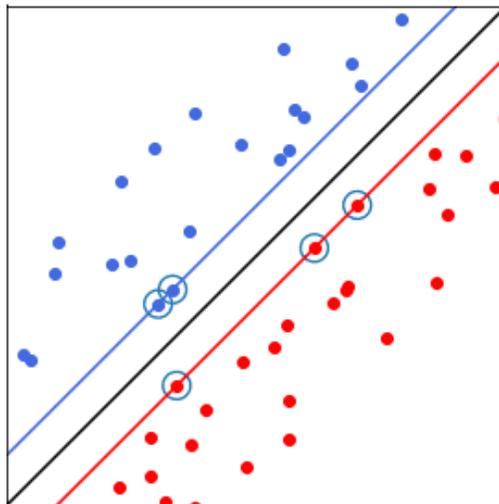


Figure I.7 :Machine à vecteurs de support [26].

I.10.2. Modèle du KNN

L'idée de base est la suivante : à partir d'un ensemble de données d'apprentissage (échantillons étiquetés), on prédit la classe d'une nouvelle donnée en l'étiquetant par la classe

majoritaire des k plus proches échantillons. Le seul hyperparamètre à fixer est donc le nombre de voisins à considérer (k) [27].

Dans la figure suivante k = 3 (la classe du point ? est Δ).

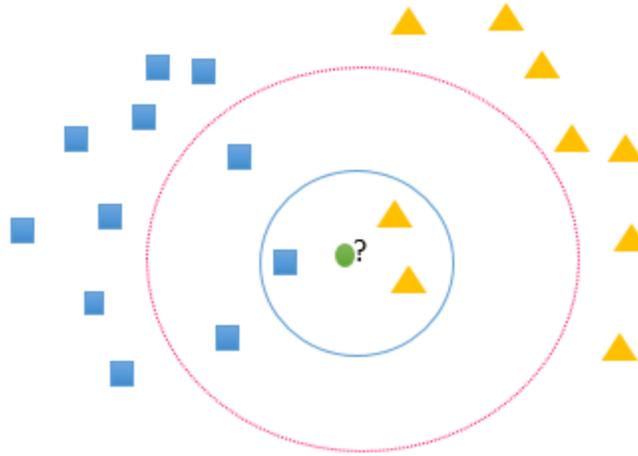


Figure I.8: Méthode des K plus proches voisins.

I.10.3. Un réseau de neurones

Un réseau de neurones consiste en une couche d'entrée, les couches cachées et enfin une couche de sortie. Un poids réel $W_{ji} \in R$ est associé à chaque connexion entre deux neurones i et j de couches consécutives. La sortie d'un neurone i est donnée par la formule suivante :

$$h_i = \sigma \left(\sum_{j=1}^N W_{ij} x_j + b_i \right)$$

Tels que σ est la fonction d'activation (de transfert ou de seuillage), N est le nombre de neurones dans la couche précédente, b_i est le biais.

Le but de la fonction d'activation est d'introduire le non linéarité dans le réseau de neurones. La fonction de transfert la plus utilisée est la ReLU (rectified linear unit) défini par :

$$\sigma(x) = \max(x, 0)$$

Le réseau de neurones est entraîné par l'ensemble d'apprentissage, comme les sorties désirées sont connues, l'objectif est de minimiser la fonction d'erreur entre les sorties obtenues et les sorties effectives en ajustant les poids des interconnexions (backward propagation) [28].

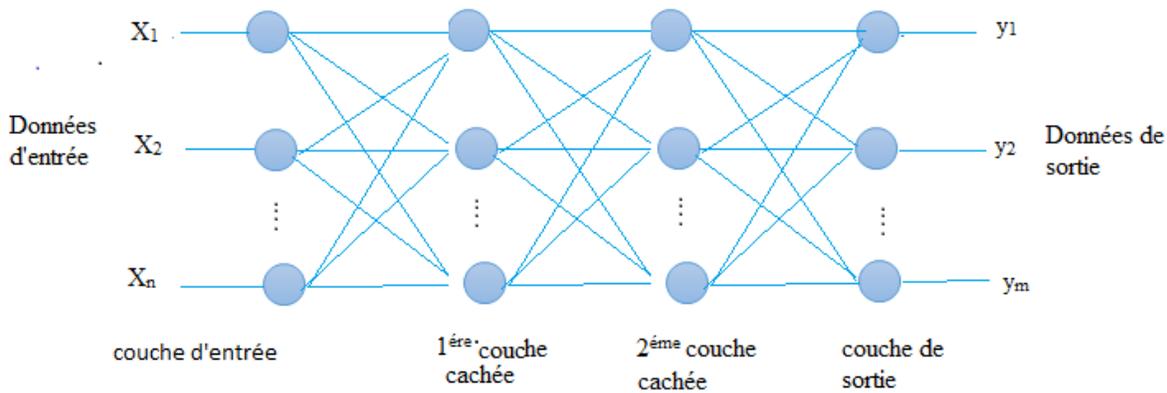


Figure I.9: Réseaux neurones artificielles.

I.10.4. Algorithme de Daugman

Elle est basée sur plusieurs avancées majeures et originales : d'abord, Daugman a proposé une méthode de détection de l'iris dans l'image de l'œil. Détecter l'iris revient à détecter au moins les pixels formant la frontière entre l'iris et la pupille, et l'iris et la sclérotique par la méthode d'opérateur intégro-différentiel. Il a aussi développé une méthode pour normaliser la forme de l'iris « RubberSheet » qui pourrait être décrite comme une tentative d'étendre le disque de l'iris comme du caoutchouc.

Elle est dite pseudo-polaire car les cercles d'iris et de pupille ne sont pas concentrique (n'ont pas le même centre). Pour la troisième phase, il a utilisé un banc des filtres de Gabor qui permet d'extraire des informations pertinentes relatives à la texture d'image de l'iris normalisé [2].

Conclusion

Aujourd'hui, la biométrie est la méthode la plus avancée de sécurisation de la confiance. Elle est en effet de plus en plus appliquée en raison de ses bienfaits. Ses applications sont diverses : applications de contrôle d'accès et applications e-commerce.

Dans ce chapitre nous avons introduit les facteurs d'identification biométrique, un bref aperçu sur quelques techniques biométriques. Nous avons aussi expliqué le système de reconnaissance de l'iris avec ses étapes et quelques techniques de reconnaissance de l'iris utilisées.

Chapitre 02

Architecture d'un réseau de neurones convolutif

Introduction

Les ingénieurs s'efforcent à chaque fois de développer le procédé de reconnaissance de l'iris, il existe de nombreuses façon pour effectuer cette procédure nous citons : Algorithme duDaugman, méthode SVM, méthode KNN et réseau neurone. Qui sont présenté dans le premier chapitre.

Avec le récent développement de la technologie et l'émergence de deeplearning, les réseaux neurones convolutifs sont apparus pour compenser les méthodes traditionnelles parmilesquels le CNN a une architecture spécifique au traitement d'image.

Un réseau de neurones convolutif est un type très particulier de réseau neurones multicouche et conçu pour reconnaître les motifs visuels directement à partir d'images avec un traitement minimal [29].

Dans ce chapitre, nous étudierons le réseau de neurones artificiel ainsi que leurs algorithmes d'optimisation et nous allons aborder l'architecture des réseaux de neurones convolutifs.

II.1. Les Réseaux de neurones artificiels

Les réseaux de neurones artificiels sont connus trois grandes phases de développement, leur premier succès est venu en 1940 avec les travaux de McCulloch et Pitts. La deuxième phase vient dans les années 60 avec le théorème de convergence de Rosenblatt et le travail de Minsky et Papert démontrant les limitations d'un perceptron. Ceci a conduit à une démotivation de la communauté scientifique qui a conduit à une stagnation dans le domaine qui a duré 20 ans. Depuis 1980, Les réseaux des neurones ont reçu un intérêt vif [30].

Les réseaux de neurones artificiels sont inspirés du neurone biologique, est un système qui se base sur le fonctionnement du cerveau humain pour l'adapter à des ordinateurs équipés de fonctions d'intelligence artificielle. Avec les réseaux de neurones artificiels l'ordinateur arrive à résoudre des problèmes de manière autonome.

Les réseaux de neurones artificiels sont composés d'une couche de nœud, contenant une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. Chaque nœud, ou neurone artificiel, se connecte à un autre et a un poids et un seuil associé.

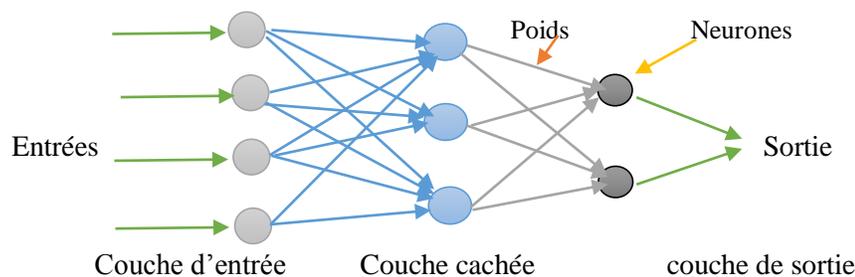


Figure II.1 : Les différentes couches des réseaux de neurones artificiels

II.1.1. Le perceptron

Le perceptron est la version la plus ancienne et simple des réseaux de neurones. Capable de résoudre uniquement les problèmes linéairement séparés. Il est composé d'une seule sortie y à laquelle toutes les entrées x_i sont connectées avec. Ses entrées et sa sortie étant booléennes [31].

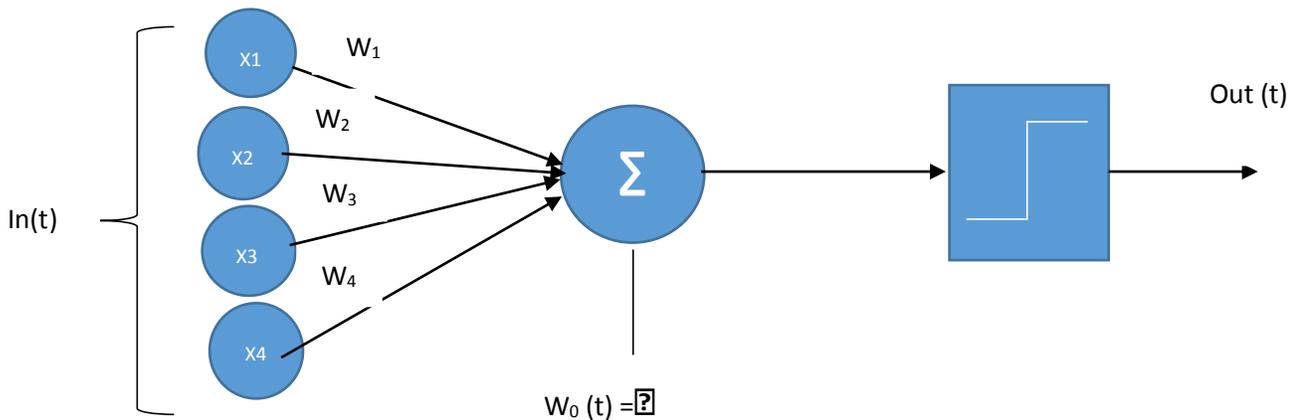


Figure II.2 : Modèle de perceptron.

II.1.2. Les Perceptrons Multicouches (MLP)

Les perceptrons multi-couches, appelés aussi MLP (Multi-layer Perceptron), sont des réseaux de neurones plus généraux que le perceptron [32] et sont non bouclés dont les nœuds sont organisés en trois niveaux ou plus appelés « couches ». Les couches voisines sont complètement connectées, c'est-à-dire, les nœuds de chaque couche sont liés à tous les nœuds de la couche inférieure et à tous ceux de la couche supérieure. En revanche, aucune connexion n'existe entre les unités d'une même couche. Un MLP est constitué de trois types de couches, une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie [33].

La couche d'entrée est la première couche du réseau. Les activations de cette couche reçoivent l'information fournie par les vecteurs d'entrée de chaque instance. Cette couche ne comporte donc pas de connexions en entrée venant d'autres nœuds. Elle est par contre complètement connectée à la première couche cachée :

- Dans un MLP contenant N ($N \geq 1$) couches cachées, chacune des $(N - 1)$ couches cachées inférieures est complètement connectée à celle qui y est supérieure. La N -ième et dernière couche cachée est complètement connectée à la couche de sortie.
- Les activations des neurones de la couche de sortie représentent les valeurs du vecteur de

- sortie du MLP.

II.1.3.Apprentissage de réseau de neurones

Toute l'information que peut contenir un réseau de neurones réside dans les poids synaptiques. L'apprentissage consiste donc à ajuster ces derniers de telle façon qu'il puisse générer correctement la sortie correspondante à chaque point de l'espace d'entrée. Ainsi, l'apprentissage peut être défini comme une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré [34].

II.1.3.1.Apprentissage supervisé

En apprentissage supervisé, la sortie attendue ou correcte du réseau à une entrée donnée est connue a priori. L'apprentissage du réseau consiste alors à mesurer l'écart entre son comportement actuel et un comportement de référence, et à corriger ses poids pour réduire cette erreur.

II.1.3.2.Apprentissage non supervisé

En apprentissage non supervisé, seules les valeurs d'entrée sont disponibles et le réseau est libre de converger vers n'importe quel état final. Aucune connaissance préalable de la sortie souhaitée n'est requise et le processus d'apprentissage est basé uniquement sur les valeurs d'entrée. Le réseau s'organise de manière à optimiser une fonction de coût spécifique.

II.1.4. Algorithme d'optimisation des paramètres dans un réseau de neurones

Pour effectuer l'optimisation des paramètres de réseau de neurones, il faut passer par les phases suivantes :

II.1.4.1.Initialisation des poids

En termes d'annotations, si l'on considère le neurone $n^{\circ 3}$ de la couche IV, le poids w associé à sa connexion au neurone $n^{\circ 1}$ de la couche III précédente est noté : $w_{3,1}^{IV}$. Si on généralise: w_{ij}^l est le poids qui relie le $i^{\text{ème}}$ neurone de la $l^{\text{ème}}$ couche au $j^{\text{ème}}$ neurone de la $(l - 1)^{\text{ème}}$ couche. Lorsqu'on évoque la somme de tous les poids du $i^{\text{ème}}$ neurone de la $l^{\text{ème}}$ couche, on signifie que c'est la somme de tous les poids des connexions synaptiques existantes entre les neurones de la couche $(l - 1)$ et ce $i^{\text{ème}}$ neurone de la $l^{\text{ème}}$ couche. Cela peut s'écrire comme : $P_i^l = \sum_j w_{ij}^l$

Le choix des poids initiaux des couches cachées du réseau est un point fondamental de la phase d'apprentissage. Pourquoi ? L'initialisation avec les poids appropriés peut faire la différence entre un réseau convergent dans un délai raisonnable et un réseau non convergent malgré un nombre d'itérations qui implique plusieurs jours voire des semaines d'entraînement [35].

II.1.4.2.L'admission des données

Les données sont connectées directement avec la couche d'entrées des réseaux de neurones.

II.1.4.3. Propagation avant

L'information se propage dans une seule direction vers l'avant, de la couche d'entrées passant par les couches cachées vers les couches de sorties, cet algorithme basé sur des fonctions mathématiques : la fonction d'agrégation et la fonction d'activation.

- **La fonction d'agrégation**

$$z_i^l = \sum_{j=1}^n w_{ij}^l a_j^{l-1} + b_i^l$$

z_i^l : La valeur d'agrégation de l^{ème} neurones de la l^{ème} couche.

l : Le numéro de couches du réseau de neurones.

i : Le numéro de neurones.

w_{ij}^l : Le poids qui relie le i^{ème} neurones de la l^{ème} couche au j^{ème} neurones de la (l - 1)^{ème} couche.

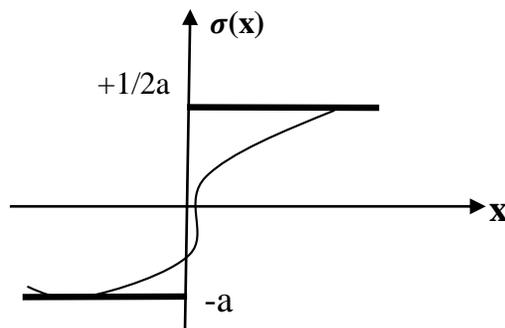
a_j^{l-1} : La valeur d'activation du j^{ème} neurones de la (l - 1)^{ème} couche.

b_i^l : Le biais associé au le i^{ème} neurones de la l^{ème} couche.

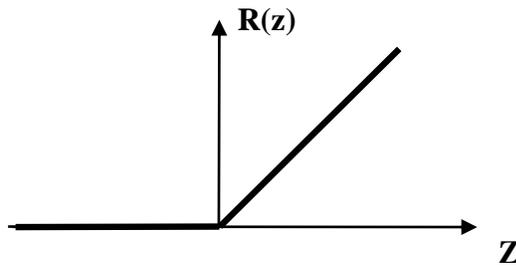
- **La fonction d'activation**

Le but de la fonction d'activation est de transformer le signal d'entrée d'un nœud à un signal de sortie. Cette transformation peut introduire propriétés non linéaires de MLP afin qu'il puisse résoudre des problèmes de non-linéarité et traitement de données complexes comme les images et vidéo. Il existe deux types de fonctions d'activation : des fonctions linéaires (identité) et des fonctions non linéaires (sigmoïde, softmax, tangente hyperbolique et unités linéaires rectifiées). En général, les couches cachées utilisent la même fonction d'activation (tangente hyperbolique, sigmoïde ou unité linéaire rectifiée) et la couche de sortie est basée sur la fonction softmax ou la fonction sigmoïde, selon type de classement [36].

Fonction sigmoïdale : $\sigma(x) = \frac{1}{1+e^{-x}}$



Fonction Relu : $\mathbf{R}(z) = \max(0, z)$



II.1.4.4.Fonction de coût

Une fonction de coût est une fonction qui permet de quantifier les erreurs effectuées par un modèle.

C'est une fonction qui mesure les distances entre les sorties $a(z)$ et les données y , ce calcul se fait avec la fonction de coût.

$$L = \frac{-1}{m} \sum_{i=1}^m y_i \log(a_i) + (1 - y_i) \log(1 - a_i)$$

m : Nombre de données.

y_i : Données d' $i^{\text{ème}}$ neurones.

a_i : Sortie d' $i^{\text{ème}}$ neurones.

II.1.4.5.Rétro propagation

Consiste à déterminer comment la sortie du réseau varie en fonction des paramètres (w , b), présents dans chaque couche du modèle, pour ça on calcule une chaîne du gradient indiquant comment la fonction varie en fonction de la dernière couche. Puis comment la dernière couche varie en fonction de l'avant-dernière couche jusqu'à arriver à la première couche de notre réseau de neurone.

II.1.4.6.Gradient descente

Consiste à ajuster les paramètres W et b de façon à minimiser les erreurs du modèle, c'est-à-dire à minimiser la fonction coût.

Pour ça, il faut déterminer comment cette fonction varie en fonction des différents paramètres.

$$w_{t+1} = w_t - \alpha \frac{\partial L}{\partial w_t}$$

w_{t+1} : Paramètre W à l'instant $t+1$.

w_t : Paramètre W à l'instant t .

α : pas d'apprentissage positif.

$\frac{\partial L}{\partial w_t}$: Gradient à l'instant t.

II.1.4.7. Mise à jour des poids

Entraîner un réseau de neurones consiste à répéter la mise à jour des poids w_{ij}^l et biais b_j^l jusqu'à la convergence du réseau. Pour ça on s'aide à l'algorithme de descente de gradient [35].

II.2. Les réseaux de neurones convolutifs

La méthode de deep learning est un sous-ensemble d'apprentissage automatique composé des réseaux de neurones qui exécutent le processus d'apprentissage sans intervention humaine. La plupart des méthodes de deep learning utilisent des architectures des réseaux de neurones, ce qui explique pourquoi il est souvent question des réseaux de neurones profonds pour désigner des modèles de deep learning. L'apprentissage profond contient de nombreux types, notamment : RNN, CNN, GAN, LSTM et MLP...etc.

Le réseau de neurones convolutifs est l'un des types populaires des réseaux de neurones profonds, sont utilisées avec succès dans un grand nombre d'applications. La tâche de reconnaissance de l'écriture manuscrite a été l'une des premières applications de l'analyse d'image par réseaux de neurones convolutifs en plus de fournir des bons résultats sur des tâches de détection d'objet et de classification d'images [35].

Les CNN utilisent des couches convolutifs pour extraire les caractéristiques directement des données. Son architecture est idéale pour travailler avec des données raster, telles que des images. Les réseaux de neurones convolutifs éliminent le besoin d'effectuer une extraction manuelle des caractéristiques, qui est souvent une tâche fastidieuse.

Les fonctionnalités utilisées ne sont pas pré-entraînées, c'est-à-dire que le réseau les apprend en s'entraînant sur des images. Cette extraction automatique des caractéristiques permet aux CNN d'atteindre des performances élevées dans la classification des objets.

II.2.1. Architecture d'un réseau de neurones convolutif

Un réseau de neurones convolutif est composé de quatre types de couche : couche de convolution, couche de mise en commun (pooling), Couche de vectorisation et couche de classification.

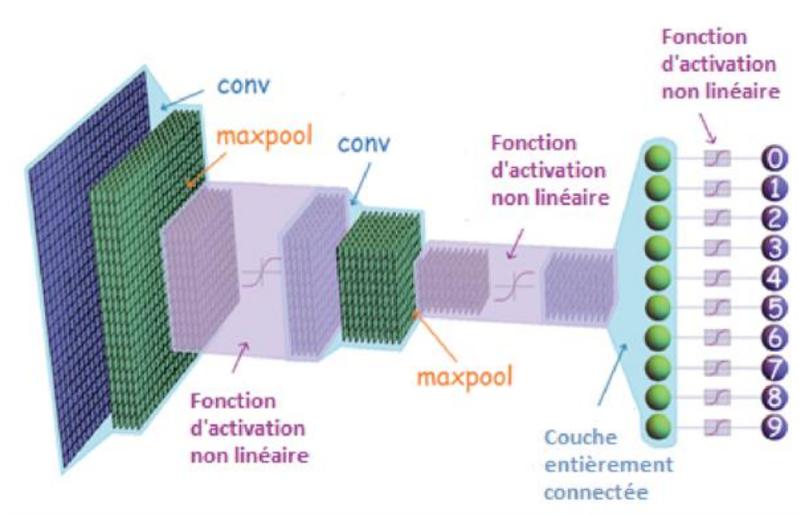


Figure II.3 : L'architecture générale d'un réseau de neurones convolutif [37].

II.2.1.1. Couche de convolution

Le but de la convolution est d'extraire les caractéristiques pertinentes des images en entrées. Pour le faire, cette couche effectue la convolution entre la première matrice qui est une sous-partie des données d'entrées avec une deuxième matrice qui représente le filtre d'apprentissage.

Une opération de sommation de ces pixels calculés de la convolution des deux matrices précédentes pour obtenir un seul résultat représenté sous forme d'une case. Le filtre se glisse en avant avec un pas S sur le reste du champ respectons les dimensions de la matrice d'entrées. Cette opération sera répéter pour tout l'image jusqu'à le dernier pixel pour obtenir une nouvelle image avec des nouvelles dimensions.

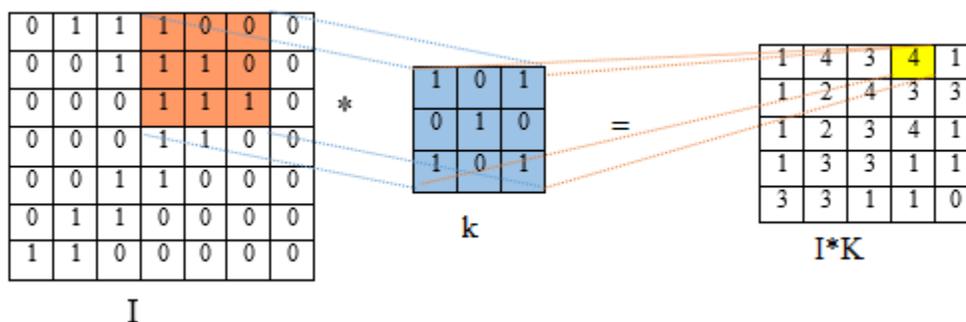


Figure II.4: Illustration de convolution.

Le pas S de déplacement de la matrice des données d'entrées a pour objectif de gérer la dimension de la matrice de l'image de sortie, donc il faut choisir un pas de tel sort qu'il produise une image aux dimensions souhaitées.

Pour ne pas perdre les données qui existent sur les bords de l'image, le CNN utilise une technique de « padding », qui sert à ajouter des pixels aux frontières de l'image, ces pixels sont fixés à « 0 ».

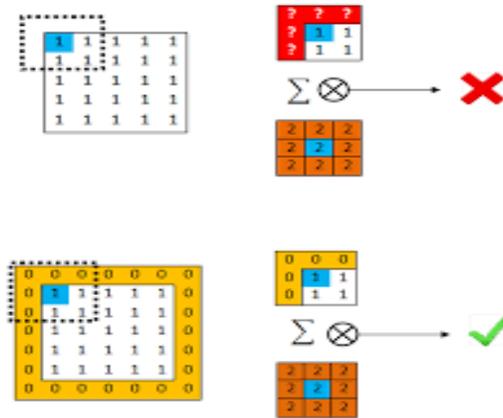


Figure II.5 : Effet de l'ajout ou de la suppression du padding dans la convolution [38].

Les pixels qui forment l'image de l'entrée sont confinés entre « 0 » et « 255 » pour une image à 8 bits, donc le résultat de la convolution peut comporter des valeurs supérieures à 255, à fin de normaliser ces valeurs, on leur applique des fonctions d'activation non linéaires pour obtenir des nouvelles valeurs à partir des actuelles valeurs, Ces fonctions ne modifient pas la résolution spatiale des featuremaps, parmi ces fonctions nous citons fonction sigmoïde, tangente hyperbolique et ReLU.

La fonction la plus utilisée est la fonction « ReLU » qui consiste à rendre les valeurs négatives à zéro et conservant les autres valeurs.

Les paramètres sont fixés pendant la convolution :

- Le nombre de filtres (kernel).
- Le pas de glissement (S).
- Le padding (P).
- La dimension de filtre ($D_f \times D_f \times m$ élément).

Noter que $m=1$: image au niveau de gris ; $m=3$: image en couleur (RVB)

Chaque pixel comporte plusieurs canaux la Figure II.6 la profondeur de filtre (3D pour le cas RVB) est augmentée pour correspondre à celle de la taille du canal.

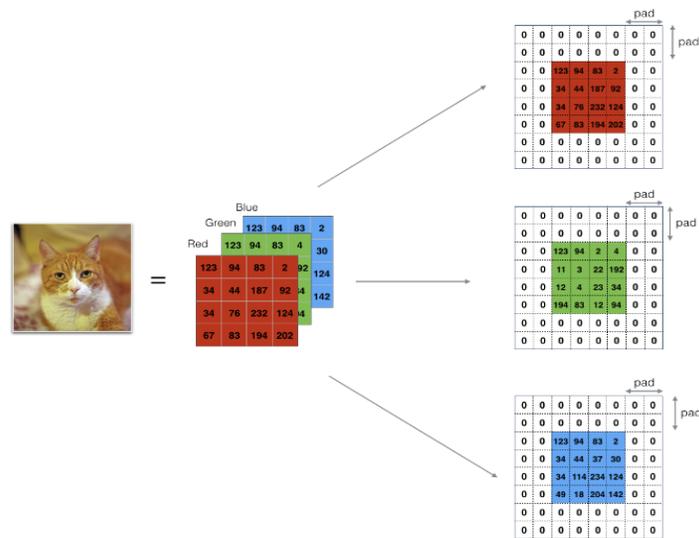


Figure II.6: Exemple de convolution en 3D (RVB) [39].

II.2.1.2. Couche de mise en commun (pooling)

Cette étape est insérée après chaque couche de convolution. Elle sert à réduire la taille spatiale de l'image intermédiaire, donc diminue les nombres des paramètres de réseau.

Une fenêtre glissante sur l'image avec des paramètres fixes durant cette couche (le pas S et la dimension de fenêtre de pooling $D_p \times D_p$), cette opération opère sur chaque canal de l'image indépendamment et ne requiert aucun paramètre. La figure II.7 montre les différentes fonctions utilisées : moyen pooling et max pooling.

A la sortie de couche de convolution, une image de taille $W \times H \times m$ est passée à la couche pooling qui produit à la fin une matrice $W_c \times H_p \times D_p$, où :

$$W_c = \frac{W - D_p}{S} + 1 \qquad H_p = \frac{H - D_p}{S} + 1 \qquad D_p = m$$

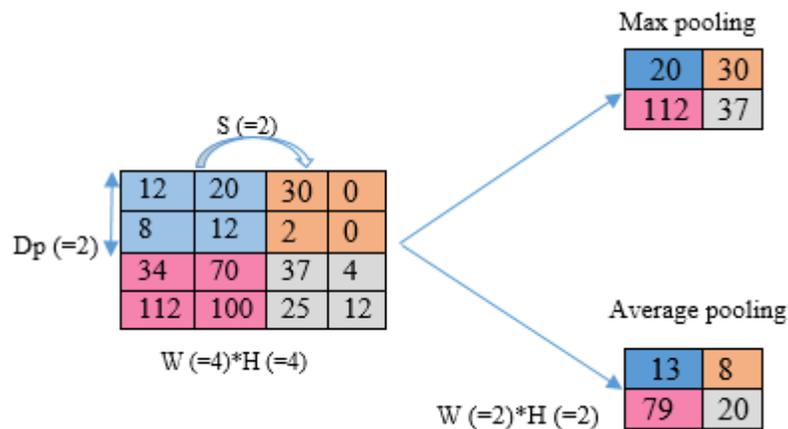


Figure II.7: Illustration de l'étape de sous échantillonnage (pooling).

II.2.1.3. Couche de vectorisation

Après l'enchaînement d'un certain nombre de couches de convolutions/pooling, l'image d'entrée du réseau se transforme à une featuremap dont la largeur et la hauteur sont très petites par rapport à l'image d'entrée, et la profondeur (nombre de canaux) très grande, Il n'est donc plus nécessaire de conserver les informations sous la forme d'une featuremap .Ainsi, une étape de vectorisation est nécessaire afin de combiner les caractéristiques locales détectées par les couches précédentes, les éléments de chaque canal de la dernière featuremap sont empilés en un vecteur.

II.2.1.4. Couche de classification

Après la vectorisation les featuremaps sont inclus dans la première couche de réseau de neurones. Une classification des caractéristiques s'effectue par le réseau géré. La partie de classification s'effectue dans la dernière couche du réseau de neurones, qui utilise une fonction d'activation « softmax », qui sert à normaliser les valeurs associées aux classes pour avoir des valeurs entre « 0 » et « 1 », la somme des résultats associés à chaque classe est égale à 1, donc représente une probabilité associée à chacune de ces classes [17].

$$f_j(z) = \frac{e^{z_j}}{\sum_{k=1}^k e^{z_k}} \quad \forall j[1; k] \text{ avec } \sum_{j=1}^k f_j(z) = 1$$

Conclusion

L'apprentissage et la classification basés sur les images utilisent le plus souvent des filtres de convolution pour identifier les informations pertinentes sur les images, par exemple les réseaux de neurones convolutifs. Dans ce chapitre, nous avons parlé d'algorithme d'optimisation des paramètres dans un réseau de neurones. Nous avons aussi expliqué et détaillé les couches de la convolution

CHAPITRE 03

Évaluation des performances d'un modèle CNN

Introduction

Dans ce chapitre, on effectue une étude des performances d'un modèle CNN qui sera utilisé avec différents classifieurs SVM, KNN, Naïf Bayes et DecisionTree simulés avec le Matlab.

Pour valider notre travail nous avons testé ces modèles sur une base de données IITD afin de savoir lequel est le modèle le plus performant.

III.1. Modèle du réseau de neurones convolutif utilisé

Dans cette section nous proposons un modèle de réseau de neurones convolutif pour la reconnaissance de l'iris.

L'architecture de ce modèle se compose de :

- Couche en entrée : images normalisées de taille 3 X 48 X 42.
- Cinq couches+ de convolution et une couche de réseau de neurones totalement connectée, permettant de détecter les caractéristiques cachées des images de l'iris.
- Réseau entièrement connectée (fully connecte) : avec une seule couche cachée, permettant la classification des images selon les caractéristiques extraites par les couches de convolution.
- Les valeurs des composantes du vecteur de sortie (prédiction) sont normalisées en utilisant l'équation softmax :

$$F_{\theta}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

z_i : est la $i^{\text{ème}}$ sortie

- La métrique à optimiser est cross entropy loss, qui est définie par la formule suivante :

$$CEL = - \sum_i y_i \times \log(\hat{y}_i)$$

Telles que y_i et \hat{y}_i sont respectivement la valeur réelle et la valeur estimée par le modèle (probabilité) de la composant i du vecteur de sortie.

	Filtres / Kernel neurons	Kernel	Stride	Padding	Activation function
Conv1	16	8	2	2	RELU
Max Pool1		4	2		
Conv2	16	4	1	2	RELU
Max Pool2		4	2		
Conv3	32	3	1	2	RELU
Max Pool3		3	2		
Conv4	32	3	1	2	RELU
Max Pool4		3	2		
Conv5	64	3	1	2	RELU

Tableau III.1 : Les caractéristiques du modèle CNN proposé.

Le modèle que nous présentons dans la figure III.1 est composé de cinq couches de convolution et quatre couches de max pooling et une couche entièrement connectée.

L'image normalisée en entrée est de taille 3 X 48 X432, l'image passe d'abord à la première couche de convolution. Cette couche est composée de 16 filtres, chaque filtre est de taille 8*8. Chacune de nos couches de convolution est suivie d'une fonction d'activation ReLU, cette ; maps seront créés. Le max pooling est appliqué après pour réduire la taille de l'image et des paramètres. À la sortie de cette couche, nous aurons 16 featuremaps de taille 110*110.

Ensuite, les 16 featuremaps qui sont obtenus sont passés en entrée à la deuxième couche de convolution qui est composée aussi de 16 filtres, chaque filtre est de taille 4*4 après la convolution et maxpooling nous aurons des featuresmaps 16 de taille 54*54.

On répète la même chose avec les couches de convolutions trois, quatre et cinq, La fonction d'activation ReLU est appliquée toujours sur chaque convolution. Une couche de max pooling est appliquée après la couche de convolution. À la sortie de la dernière couche, nous aurons 64 featuremaps, chaque feature est de taille 3*3. Le vecteur de caractéristiques issu des convolutions a une dimension de 64 * 9*9.

Après ces cinq couches de convolution, nous utilisons un réseau de neurones composé d'une seule couche fully connecte de 1 024 neurones, une couche de softmax qui permet de calculer la distribution de probabilité des 224 classes.

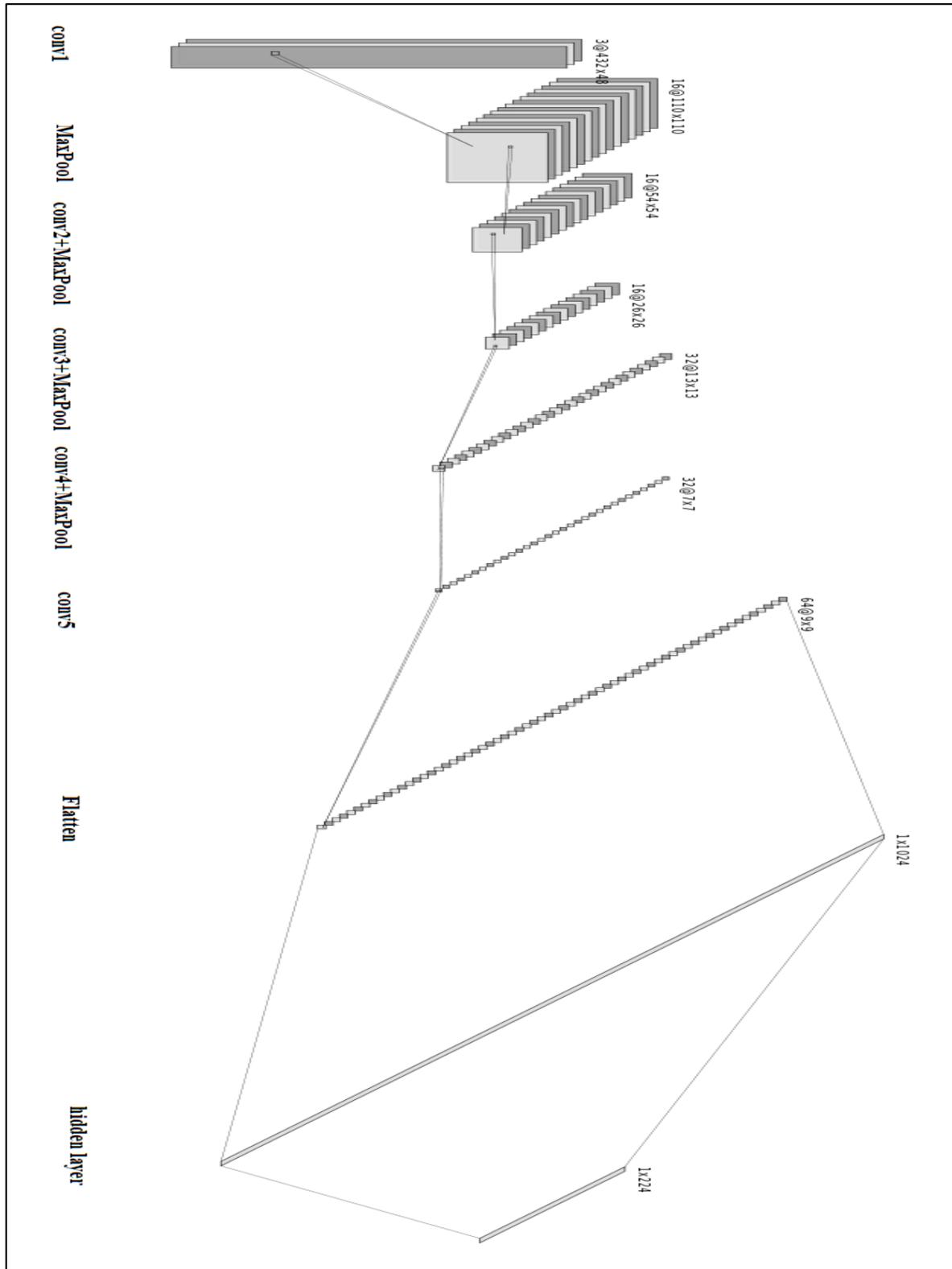


Figure III.1: Architecture du modèle CNN proposé.

III.2.Évaluations du modèle

III.2.1.Définition de la matrice de confusion

La matrice de confusion est en quelque sorte un résumé des résultats de prédiction pour un problème particulier de classification. Elle compare les données réelles pour une variable cible à celles prédites par un modèle. Les prédictions justes et fausses sont révélées et réparties par classe, ce qui permet de les comparer avec des valeurs définies [40]

Les matrices de confusion dans notre cas sont de taille 224 X 224 avec $U(i, i)$ est le nombre d'instances correctement classées et $U(i, j)$ telle que $i \neq j$ est le nombre d'instances de classe i qui sont classé par le modèle comme étant de classe j ,

Les 4 valeurs sont :

- **True Positive (TP)** : la prédiction et la valeur réelle sont positives.
- **TrueNegative (TN)** : la prédiction et la valeur réelle sont négatives.
- **False Positive (FP)** : la prédiction est positive alors que la valeur réelle est négative.
- **False Negative (FN)** : la prédiction est négative alors que la valeur réelle est positive.

III.2.2.Les Métriques

Différentes métriques peuvent être calculées à partir du tableau de contingence afin d'en faciliter l'interprétation. Ces indicateurs permettent de mieux apprécier la qualité de précision du modèle.

- **Accuracy** :

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

- **Error** :

$$\text{Error} = \frac{FN+FP}{TP+TN+FP+FN}$$

- **Sensitivity** :

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

- **Specificity** :

$$\text{Spécificité} = \frac{TN}{TN+FP}$$

- **Kappa** : Il s'agit essentiellement d'une proportion de la qualité de l'exécution du classificateur par rapport à la qualité de ses performances, essentiellement par une certaine coïncidence.

$$\text{Kappa (k)} = \frac{VP \times VN + FP \times FN}{n_1 n_2 + n'_1 n'_2}$$

n_1 : VP+FP.

n_2 : FN+VN.

n'_1 : VP+FN.

n'_2 : FP+VN.

III.3.L'environnement de développement

On a utilisé le Matlab R2018b pour les simulations et l'étude des performances de notre modèle CNN, et une base de données IITD contient des images d'iris .

Dans cette section, nous présenterons l'environnement matériel la base de données de notre travail.

- Pour l'implémentation et l'évaluation du modèle proposé, nous avons utilisé un ordinateur portable personnel possédant les caractéristiques suivantes:
 - ✓ Processeur 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
 - ✓ Mémoire vive de 4 Go
 - ✓ Disque dur hybride SSD de capacité 167Go
 - ✓ Système d'exploitation Windows 10 x64 bits

III.3.1.La Base de données utilisée

La base de données IIT Delhi Iris se compose principalement des images d'iris collectées auprès des étudiants et du personnel de l'IIT Delhi, New Delhi, Inde. Cette base de données a été acquise au Biometrics Research Laboratory de janvier à juillet 2007 (toujours en cours) à l'aide de JIRIS, JPC1000, caméra numérique CMOS [41].

La base de données actuellement disponible provient de 224 utilisateurs, toutes les images sont au format bitmap (*.bmp). Tous les sujets de la base de données sont dans le groupe d'âge 14-55 ans composé de 176 hommes et 48 femmes. La base de données de 1120 images est

organisée en 224 classes. La résolution de ces images normalisées est de 48 X432 pixels. La normalisation est obtenue en utilisant le modèle de Daugman's Rubber.

III.4. Évaluation des performances

III .4.1. Notions fondamentales

- **Ensemble de données d'apprentissage** : Utilisé lors de la phase d'entraînement, le système s'entraîne sur ces données pour effectuer la tâche attendue de lui.
- **Ensemble de validation** : il est plus petit que l'ensemble d'apprentissage et est utilisé pour évaluer les performances des modèles avec différentes valeurs d'hyperparamètres. Il est également utilisé pour détecter le surajustement pendant les phases d'entraînement.
- **Ensemble de test** : cet ensemble est utilisé pour avoir une idée de la performance finale d'un modèle après le réglage des hyperparamètres.
- **Epoch** : représente une itération sur l'ensemble du jeu de données.
- **Batch** : nous ne pouvons pas transmettre l'intégralité de l'ensemble de données au réseau de neurones en une seule fois. Nous divisons donc l'ensemble de données en nombre de batch.

III.4.2. Étude des performances du modèle propose

III.4.2.1 Relation entre le nombre de couches de la convolution, la vitesse

d'apprentissage et la précision:

Notre modèle d'apprentissage consiste en des caractéristiques comme le batch de taille 64, de plus le taux d'apprentissage (learning rate) qui est réduit de 20% après chaque 5 epochs, et le gradient descent avec momentum qui représente l'algorithme d'optimisation.

- Architecture à une seule couche de convolution

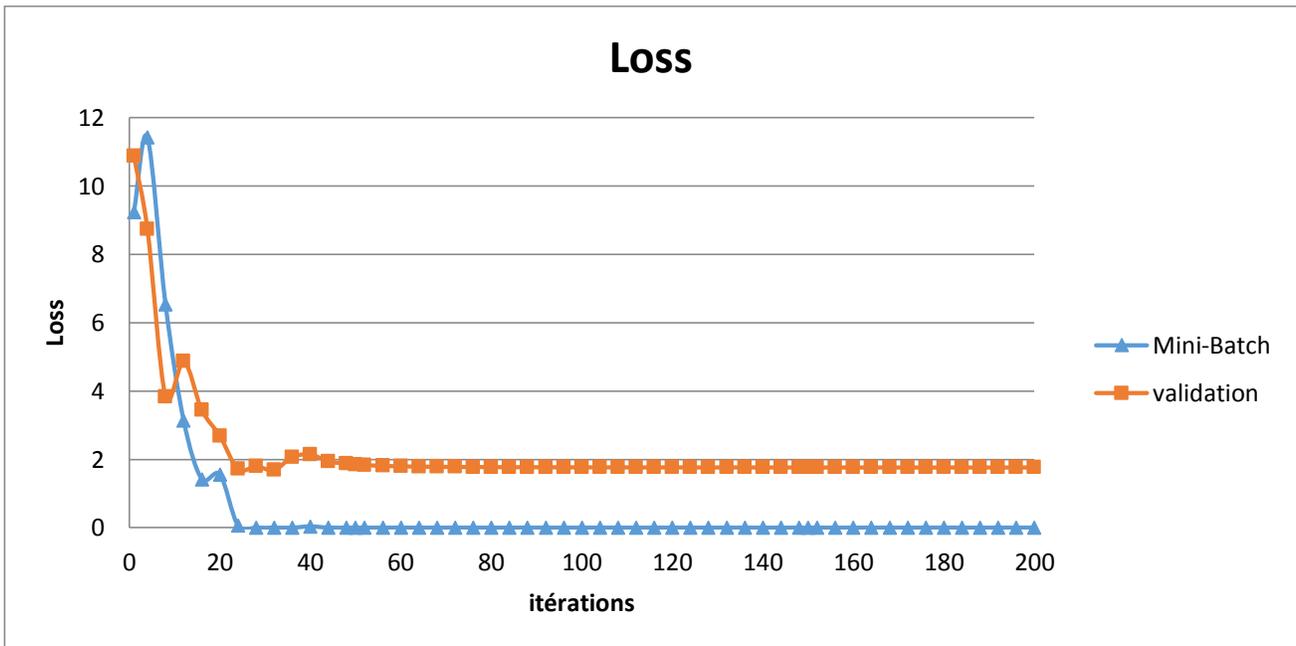
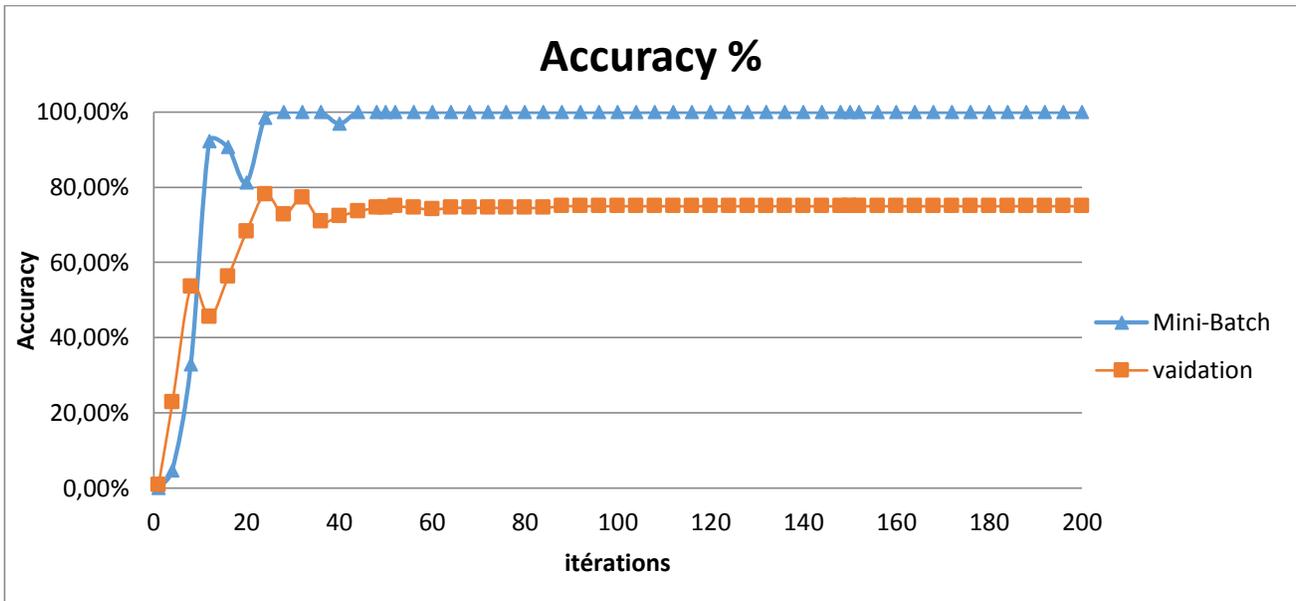


Figure III2 : Précision et perte pour validation et Mini-Batch en utilisant une architecture à une couche de convolution.

- Architecture à deux couches de convolution

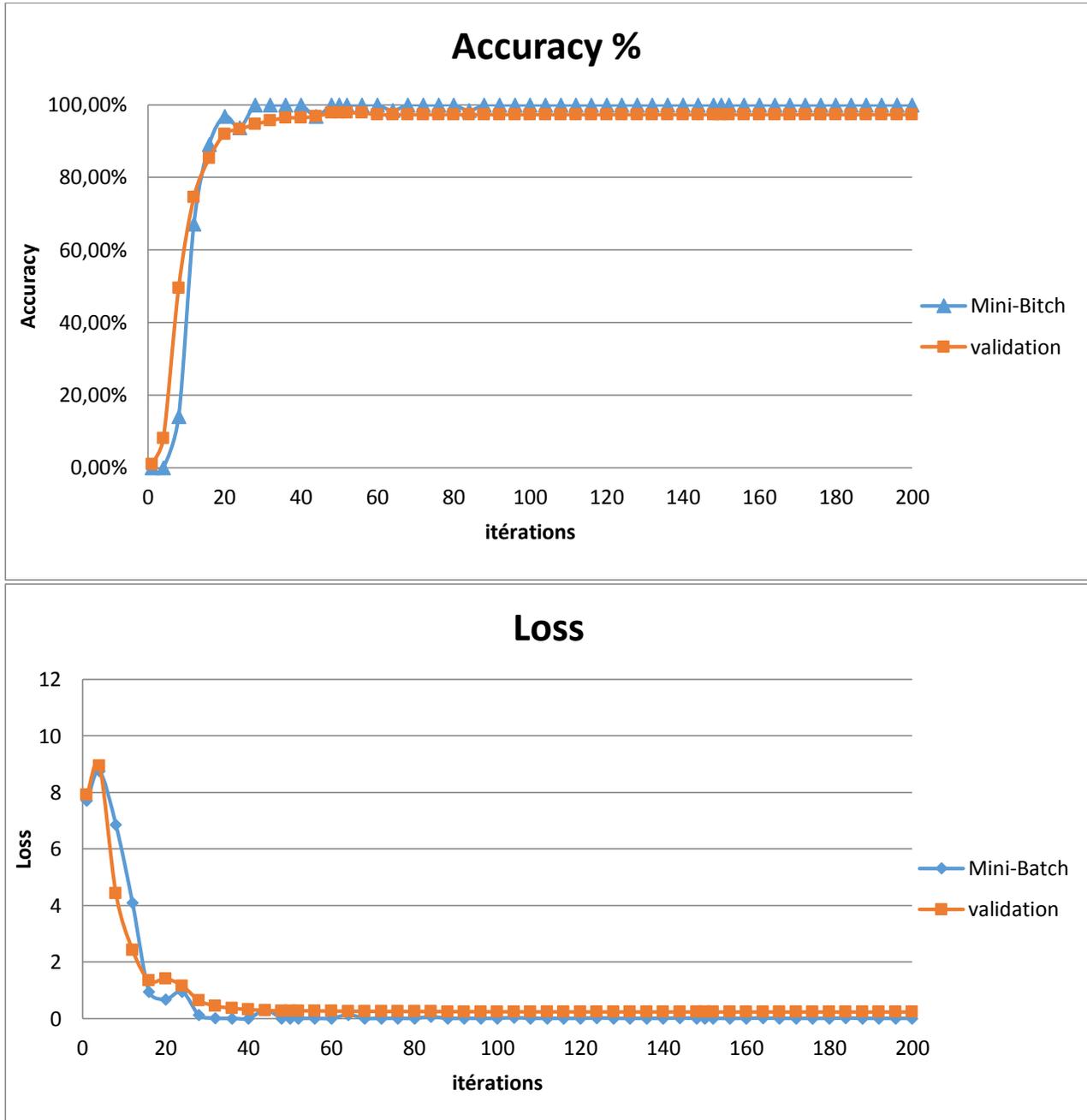


Figure III 3 : Précision et perte pour validation et Mini-Batch en utilisant une architecture à deux couches de convolution.

- Architecture à trois couches de convolution

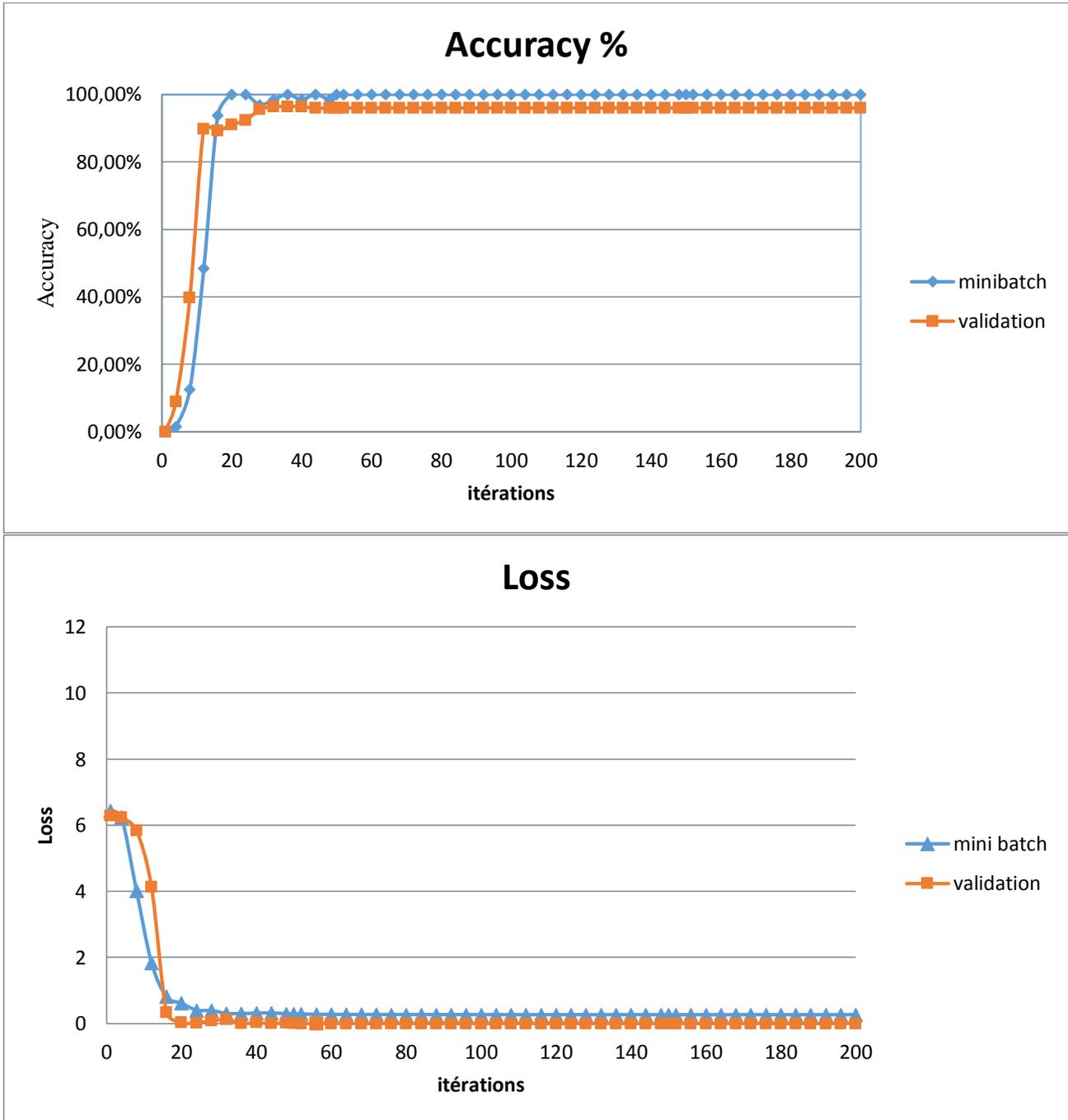


Figure III 4 : Précision et perte pour validation et Mini-Batch en utilisant une architecture à trois couches de convolution.

- Architecture à quatre couches de convolutions

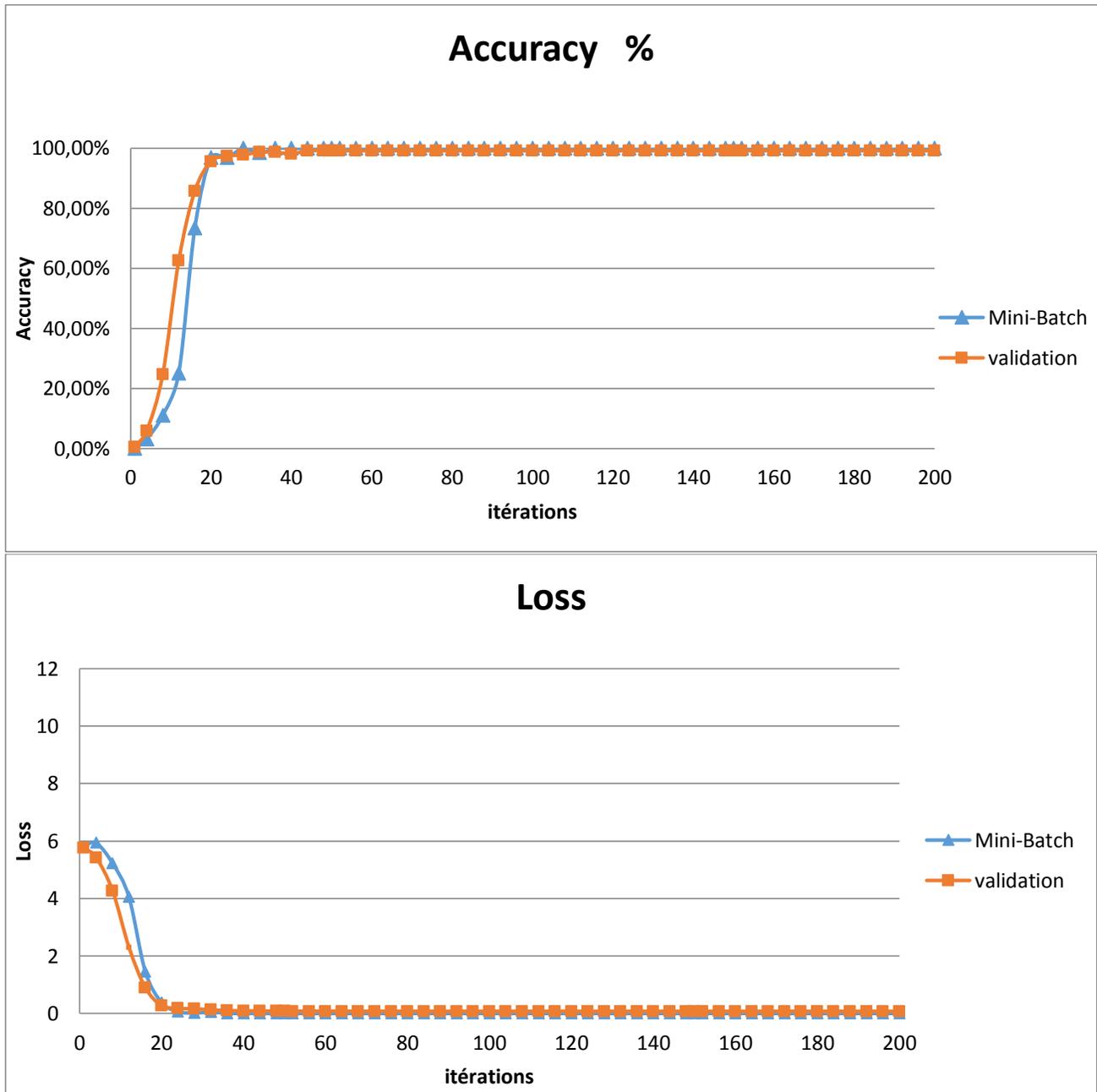


Figure III 5 : Précision et perte pour validation et Mini-Batch en utilisant une architecture à quatre couches de convolution.

- Architecture à cinq couches de convolution

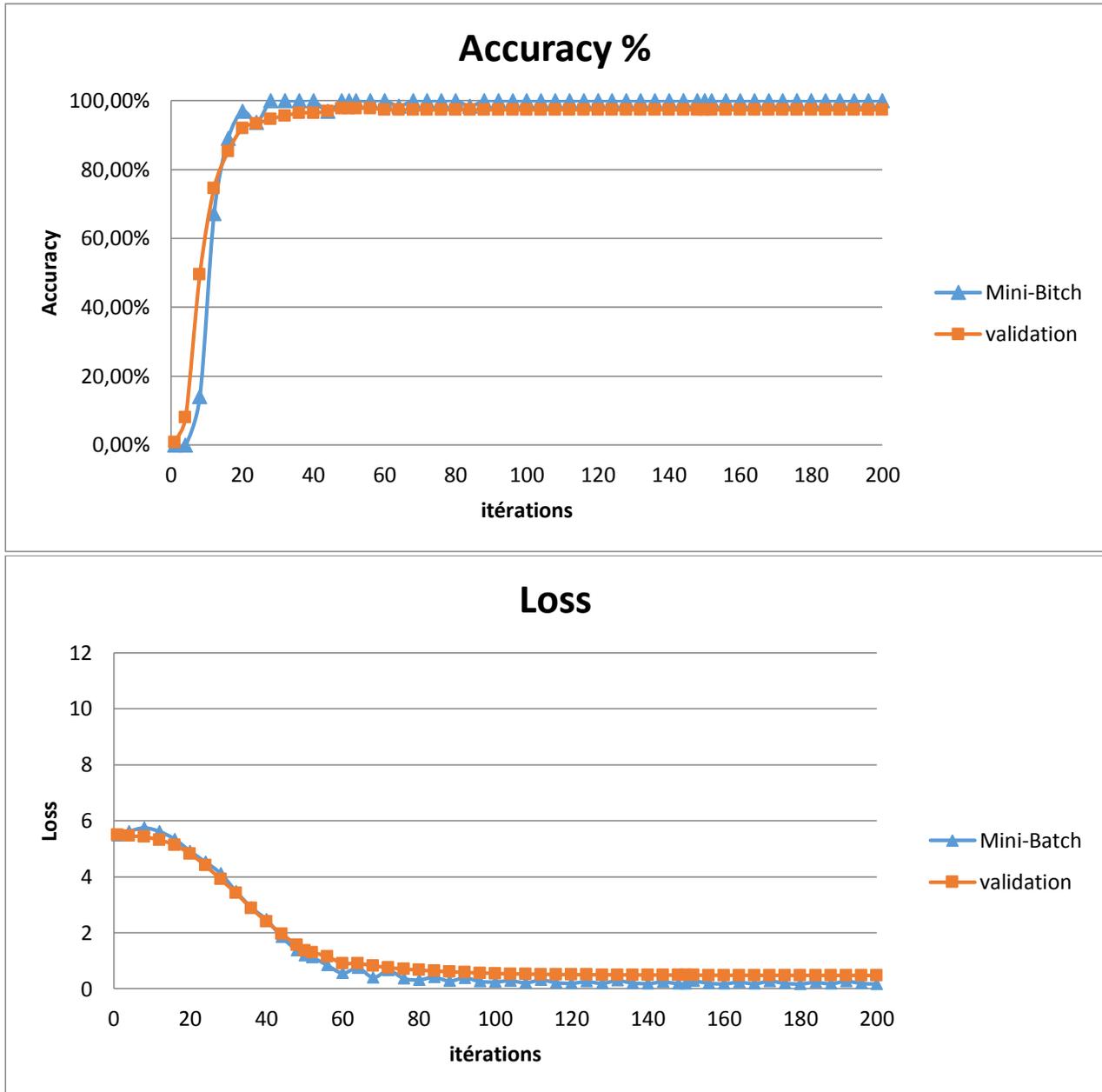


Figure III 6 : Précision et perte pour validation et Mini-Batch en utilisant une architecture à cinq couches de convolution.

Analyse :

- De manière globale, notre modèle permet une bonne précision dans la reconnaissance des iris avec un temps d'entraînement acceptable (la 5^{ème} couche : précision > 90% à partir de la 50^{ème} itération, à la 3^{ème} couche : précision >95% à partir de la 32^{ème} itération, à la première couche : précision >74% à partir de la 50^{ème} itération).
- Avec l'augmentation du nombre de couches on constate une lenteur dans la convergence de l'algorithme d'apprentissage, vu que le modèle de réseau de neurones nécessite l'optimisation d'un plus grand nombre des paramètres (valeurs des filtres, poids des liens, etc.) ce qui induit à temps plus long d'exécution de l'algorithme d'optimisation (qui reste de complexité linéaire). En contrepartie la valeur finale de la perte de précision (Loss) diminue avec l'augmentation de nombre des couches de convolution ce qui reflète une bonne précision de modèle.

III.4.3. Étude des métriques

Notre réseau entièrement connecté qui se compose d'une seule couche cachée permet la classification des images selon les caractéristiques extraites par les cinq couches de convolution (vecteur de caractéristique).

On relie la dernière couche cachée qui représentent les caractéristiques extraites à un classifieur automatique (SVM, KNN, Naif Bayes, DecisionTree). Le tableau ci-dessous résume notre résultat obtenu sur ces métriques (Accuracy, Error, Sensitivity, Specificity, Kappa) pour les quatre méthodes utilisées :

	Accuracy	Error	Sensitivity	Specificity	Kappa
SVM	0.9509	0.0491	0.9509	0.9998	0.8190
KNN	0.9018	0.0982	0.9018	0.9996	0.9095
Naif Bayes	0.1250	0.8750	0.1250	0.9961	0.9898
DecisionTree	0.2500	0.2500	0.9966	0.0034	0.9881

Tableau III.2 : Résultats des métriques de performances du modèle CNN proposé en utilisant les classifieurs SVM, KNN, Naif Bayes et DecisionTree

Analyse :

D'après les résultats obtenus, on remarque la variation des valeurs des métriques d'une méthode à une autre, La précision a de grandes valeurs pour les méthodes SVM et KNN.

Les méthodes SVM et KNN ont des taux d'erreurs plus faibles que les autres méthodes, ce qui donne à ces deux méthodes un avantage sur les autres méthodes Naif Bayes et DecisionTree. Ensuite pour la sensibilité et la spécificité, on remarque que les valeurs de SVM et KNN sont proches de 1, ce qui signifie qu'en revanche, le test de ces deux méthodes est parfait, car les autres méthodes ont des valeurs inférieures à 0,5 .Ce qui signifie que les deux classificateurs Naif Bayes et DecisionTree ont donnés des mauvais résultats dans notre étude.

En général, dans notre cadre de modèle, KNN et SVM sont bien meilleurs que Naif Bayes et DecisionTree, car KNN et SVM sont des classificateurs non linéaires, tandis que Naif Bayes et DecisionTree sont des classificateurs linéaires qui permettent uniquement la classification des instances linéairement séparables.

III.4.4. Relation entre le nombre de couches de convolution est la précision

Nous projetons la dernière couche cachée de notre modèle aux différents classifieurs (SVM, KNN et DecisionTree) et nous étudions les performances des modèles selon leurs précisions. Les différents résultats sont résumés dans le tableau.

Pour cette partie on a fait un test ou on a varié le nombre de couches pour avoir plusieurs valeurs de précision pour les trois méthodes SVM , KNN Et DecisionTree. On a résumé les résultats dans le tableau suivant :

	Couche 2	Couche 3	Couche 4	Couche 5
SVM	0.9643	0.9821	0.9955	0.9509
KNN	0.9999	0.9911	0.9687	0.9018
Decision Tree	0.1652	0.2143	0.3170	0.2500

Tableau III 3: La précision du modèle CNN proposé en le liant aux classifieurs traditionnels SVM , KNN Et DecisionTree dépendamment du nombre de couches de convolution.

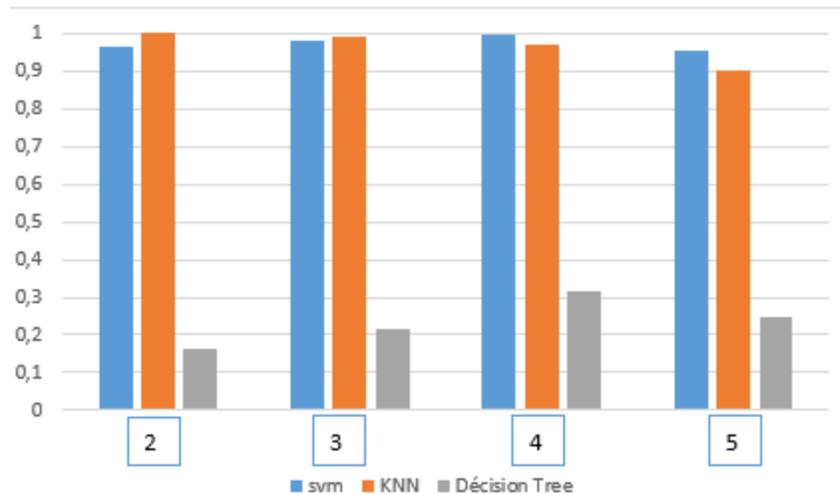


Figure III7 : Comparaison des méthodes SVM et KNN et DecisionTree selon le nombre de couches de convolution.

Analyse :

D’après les résultats de tableau obtenu pour les trois méthodes, les valeurs de précision varient, alors on remarque qu’avec SVM la valeur de précision la plus élevée est avec 4 couches, et pour le KNN la valeur de précision la plus élevée est avec 2 couches.

Le nombre de couches dépend du modèle utilisé, car l’augmentation du nombre de couches cachées permet l’extraction des caractéristiques différentes, or le modèle de distribution des vecteurs de caractéristiques influe considérablement sur les performances des classifieurset est considéré comme facteur clé du choix du classifieur dans n’importe quel apprentissage automatique.

On conclut que l’augmentation du nombre de couches permet d’améliorer la précision.

Conclusion

Dans ce chapitre nous avons proposé un modèle CNN, puis nous avons interprété les résultats de l’étude des performances dudit modèle. Avec des différents classifieurs (KNN, SVM, Naïf Bayes, DecisionTree) selon leurs précisions.

On conclut que le modèle proposé CNN donne des résultats plus performant avec les classifieurs SVM et KNN.

Conclusion Générale

Conclusion Générale

Le réseau de neurones convolutifs est la révolution du deeplearning. Utilisé en reconnaissance d'image d'iris. Grâce à l'essor des réseaux de neurones convolutifs, la classification d'images a connu une avance majeure en termes de performance vu qu'est utilisé pour la détection des caractéristiques des images.

Dans le cadre de ce PFE nous avons proposé un modèle qui contient cinq couches de convolution, suivies par quatre couches de max pooling et une couche entièrement connectée avec une seule couche cachée.

On a étudié les performances de ce modèle CNN utilisant des différents classifieurs : SVM, KNN, Naif Bayes et DecisionTree, afin d'obtenir de meilleurs résultats en termes de précision avec les deux modèles SVM et KNN.

Comme perspective, il serait intéressant de développer une application interactive basée sur le modèle proposé afin de permettre la détection des iris à partir des images des yeux prises des utilisateurs. Ces iris sont ensuite normalisées via l'algorithme de Daugman et soumis au modèle proposé afin de détecter la classe à laquelle l'iris appartient.

Bibliographie

- [1] M. Lemmouchi, « Reconnaissance Biométrique par Fusion Multimodale », Thèse de Doctorat, Traitement de signal, Université De Batna 2, 2020.
- [2] S. Djebbar, « Système de vérification de l'identité de personne par reconnaissance de l'iris », diplôme de Master, Sytème distribue, Université De Oum El Bouaghi, 2011.
- [3] <https://fr-academic.com/dic.nsf/frwiki/216284> juin 2022.
- [4] <https://www.larousse.fr/encyclopedie/divers/biom%C3%A9trie/27110> mai 2022.
- [5] <https://www.biometrie-online.net/biometrie/pourquoi-utiliser-la-biometrie> mai 2022.
- [6] http://www.unitek-france.fr/unitek_new/pourquoi-utiliser-la-biometrie mai 2022.
- [7] S. GuerfiAbabsa, « Authentification d'individus par Reconnaissance de caractéristiques Biométriques liées aux visages 2D/3D », Thèse de Doctorat, Université D'every Val D'essonne ,2008.
- [8] <https://recfaces.com/articles/types-of-biometrics#:~:text=Physical%20biometrics%20analyze%20data%20such,the%20wrist%2C%20and%20heat%20pattern> mai 2022.
- [9] H. Goumaziane, D.Laribi, « Développement d'un Système Biométrique pour la Reconnaissance de visage, Basé sur l'opérateur Binaire local(LBP) et ses variantes », Réseau et Mobilité et Système Embraqués, diplôme de Master, université de Tizi-Ouzou, 2018.
- [10] S. Boudjelal, « Détection et identification de personne par méthode biométrique », Mémoire de magister en électronique, Université de Tizi Ouzou, 2014.
- [11] L. Menssoura, « identification des visages humains par réseaux de nuerons », mémoire de magister, université de Batna, 2013.
- [12] A. Benagga et L. Telib, « Reconnaissance des personnes basée sur l'empreinte de l'articulation de doigt », diplôme de master académique, université de Ouargla, 2016.
- [13] F. Davoine, B. Abboud et V. Mo dang, » Face and facial expression analysisbased on an active appearance model », Traitement du signal, Vol.21, No.3, 2004.
- [14] A.Chachoua, I.Kahlaoui, « Reconnaissance de Personne en utilisant l'empreinte palmaire multi-spectral basée sur l'apprentissage approfondi », Mémoire Master, Automatique, université Ouargla, 2016.

Bibliographie

[15] F. Louiba et R. Hadj, « Système de contrôle d'accès physique basé sur le visage et la Java Card », d'ingénierie en informatique, Institut National de formation en Informatique (I.N.I), 2010.

[16] A. Meraoumia, « Modèle de Markov caché appliqué à la Multi Biométrie en électronique », Thèse de Doctorat, Sciences et de la Technologie, Houari Boumediene, 2014.

[17] P. Bonazza, « Système de Sécurité Biométrique Multimodal par Imagerie, Dédié au Contrôle d'accès », Thèse de Doctorat, Informatique et Instrumentation de l'Image, Université de Bourgogne, 2019.

[18] K. Draoui, « Identification des Personnes par l'Iris en Utilisant la Méthode des Paquets d'Ondelettes », Diplôme de Magister, Système de Communication Moderne, université d'Oran, 2011.

[19] S. Ouchene, « Une Approche Neuronale pour la Reconnaissance D'iris basé sur la Méthode LBP », Diplôme de Master, Vision Artificielle, Université Oum El Bouagui, 2019.

[20] D. Zhang, X. Jing, J. Yang, « Biometric Image Discrimination Technologies », Idea Group Inc. publishing, 2006.

[21] A. Hilal, « Système d'identification à partir de l'image d'iris et détermination de la localisation des informations », Thèse de Doctorat, Optimisation et Sécurité des Systèmes, université de Troyes, 2013.

[22] L. Masek, « Recognition of Human Iris Patterns for Biometric Identification », Thèse de Master présentée à l'Université de Western Australia, Australie, 2003.

[23] N. Morizet, « Reconnaissance Biométrique par Fusion Multimodale du Visage et de l'Iris », thèse de Doctorat présentée à l'École Nationale Supérieure des Télécommunications, France, 2009.

[24] <https://www.syloe.com/glossaire/authentication/> avril 2022.

[25] BOSER, Bernhard E., GUYON, Isabelle M., et VAPNIK, Vladimir N. A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory. 1992. p. 144-152

[26] <https://dataanalyticspost.com/Lexique/svm/> avril 2022.

[27] EMRULLAH, A. C. A. R. extraction of texture features from local Iris areas by GLCM and Iris recognition system based on KNN. European Journal of Technique (EJT), 2016, vol. 6, no 1, p. 44-52

[28] SIBAI, Fadi N., HOSANI, Hafsa I., NAQBI, Raja M., et al. Iris recognition using artificial neural networks. Expert Systems with Applications, 2011, vol. 38, no 5, p. 5940-5946.

[29] m. Sewak, Md.R. Karim, p. Pujari, "Practical Convolutional Neural Networks", Packt Publishing, 2018

[30] A.H. Sabry, A. Bacha, J. Benhra. « Utilisation des réseaux de neurones pour le tuning des Algorithmes d'optimisation par colonies de fourmis Application aux chaînes logistiques », Xème Conférence Internationale : Conception et Production Intégrées, Tanger, Maroc, 2015.

Bibliographie

-
- [31] A.Labiad, «Sélection Des Mots Clés Basée Sur La Classification Et L'extraction Des Règles D'association », Mathématiques Et Informatique Appliquées,Diplôme De Master, L'université Du Québec À Trois-Rivières, 2017.
- [32] H.Corentin, « Contribution Au Développement De l'Apprentissage Profond Dans Les Systèmes Distribués »,Thèse de Doctorat, Mathématique et Sciences et Technologies de l'Information et de la Communication, Université De Rennes 1,2019.
- [33] M.Bouaziz,»« Réseaux de Neurones Récurents pour la Classification de Séquences dans des Flux Audiovisuels Parallèles », Thèse de Doctorat, Sciences et Agrosiences, 'Université d'Avignon et des Pays de Vaucluse, 2017.
- [34] G. Dreyfur. « Réseaux de neurones : Méthodologie et application », édition Eyrolles, 2004.
- [35] C.Etienne, «Apprentissage Profond Appliqué A La Reconnaissance Des Emotions Dans La Voix », Thèse de Doctorat, Université Paris-Saclay, 2019.
- [36] N.Dif, «L'apprentissage profond pour le traitement d'images », Thèse de Doctorat, Technologie de l'information, Université Sidi Bel Abbés, 2022.
- [37] https://leonardoaraujosantos.gitbook.io/artificial-intelligence/machine_learning/deep_learning/convolutional_neural_networks juin 2022.
- [38] https://www.l2s.centralesupelec.fr/wp-content/uploads/gac-nicolas/rapport_stage_pencrech_L2S_2010.pdf avril 2022.
- [39] https://datascience-enthusiast.com/DL/Convolution_model_Step_by_Stepv2.html mai 2022.
- [40] <https://www.jedha.co/formation-ia/matrice-confusion>septembre 2022.
- [41] https://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Iris.htmseptembre 2022.

Code Source

CNN-Naïve Bayes

```
NMCLS = 224
PTH = './IITD'
Sz = imsize('./IITD\001\001_1.BMP')

% Chargement du dataset

disp('chargement du dataset')

imds = imageDatastore(PTH, 'LabelSource', 'foldernames', 'IncludeSubfolders',true);
[trainingSet,validationSet, testSet] = splitEachLabel(imds,0.6, 0.2, 0.2 ...
    , 'randomized');

layers = [

    imageInputLayer(Sz)
    convolution2dLayer(8,16,"Padding",1,"Stride",2)
    batchNormalizationLayer
    reluLayer('Name','relu_1')

    maxPooling2dLayer(4,"Stride",2)
    convolution2dLayer(4,16,"Padding",2)
    batchNormalizationLayer
    reluLayer('Name','relu_2')

    % maxPooling2dLayer(4,"Stride",2)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% convolution2dLayer(3,32,"Padding",2)
% batchNormalizationLayer
% reluLayer('Name','relu_3')
%
% maxPooling2dLayer(3,"Stride",2)
% convolution2dLayer(3,32,"Padding",2)
% batchNormalizationLayer
% reluLayer('Name','relu_4')
%
% maxPooling2dLayer(3,"Stride",2)
% convolution2dLayer(3,64,"Padding",2)
% batchNormalizationLayer
% reluLayer('Name','relu_5')

fullyConnectedLayer(NMCLS)
softmaxLayer('Name','softmax_1')
classificationLayer];

disp ('break point 1')

options = trainingOptions("sgdm", 'LearnRateSchedule', "piecewise", ...
    'LearnRateDropFactor', 0.2, ...
    'LearnRateDropPeriod', 5, ...
    'MaxEpochs',10, ...
    'ValidationData',validationSet, ...
    'ValidationFrequency',4, ...
    'MiniBatchSize', 64, ...
    'Plots', "training-progress")

disp ('break point 2 ')
```

```
convnet = trainNetwork(trainingSet, layers, options);

disp('break point 3')

disp('inspecter la premiere couche')

convnet.Layers(1)

disp('inspecter la derniere couche')

convnet.Layers(end)

% Display conv layers

w1 = convnet.Layers(2).Weights;
w1 = mat2gray(w1);
w1 = imresize(w1, 5);
figure
montage(w1)
title('Premiere couche de convolution')

featureLayer = 'softmax_1';
trainingFeatures = activations(convnet, trainingSet, featureLayer, ...
    'MiniBatchSize', 64, 'OutputAs', 'columns');

% Get training labels from the trainingSet
trainingLabels = trainingSet.Labels;
```

```
% Train NB
classifier = fitcnb(trainingFeatures', trainingLabels)

% Extract test features using the CNN
testFeatures = activations(convnet, testSet, featureLayer, ...
    'MiniBatchSize', 32, 'OutputAs', 'columns');

% Pass CNN image features to trained classifier
predictedLabels = predict(classifier, testFeatures);

% Get the known labels
testLabels = testSet.Labels;

% Tabulate the results using a confusion matrix.
confMat = confusionmat(testLabels, predictedLabels);

[Result,RefereceResult]=confusion.getValues(confMat);
```

Result

CNN-SVM

```
NMCLS = 224
```

```
PTH = './IITD'
```

```
Sz = imsize('./IITD\001\001_1.BMP')
```

```
% Chargement du dataset
```

```
disp('chargement du dataset')
```

```
imds = imageDatastore(PTH, 'LabelSource', 'foldernames', 'IncludeSubfolders',true);
```

```
[trainingSet, validationSet, testSet] = splitEachLabel(imds,0.6, 0.2, 0.2 ,'randomized');
```

```
% Affichage du nombre d'instances dans chaque catégorie
```

```
layers = [
```

```
    imageInputLayer(Sz)
```

```
    convolution2dLayer(8,16,"Padding",1,"Stride",2)
```

```
    batchNormalizationLayer
```

```
    reluLayer('Name','relu_1')
```

```
    maxPooling2dLayer(4,"Stride",2)
```

```
    convolution2dLayer(4,16,"Padding",2)
```

```
    batchNormalizationLayer
```

```
    reluLayer('Name','relu_2')
```

```
    maxPooling2dLayer(4,"Stride",2)
```

```
    convolution2dLayer(3,32,"Padding",2)
```

```
batchNormalizationLayer
reluLayer('Name','relu_3')
```

```
maxPooling2dLayer(3,"Stride",2)
convolution2dLayer(3,32,"Padding",2)
batchNormalizationLayer
reluLayer('Name','relu_4')
```

```
maxPooling2dLayer(3,"Stride",2)
convolution2dLayer(3,64,"Padding",2)
batchNormalizationLayer
reluLayer('Name','relu_5')
```

```
fullyConnectedLayer(NMCLS)
softmaxLayer('Name','softmax_1')
classificationLayer];
```

```
disp ('break point 1')
```

```
options = trainingOptions("sgdm", 'LearnRateSchedule', "piecewise", ...
    'LearnRateDropFactor', 0.2, ...
    'LearnRateDropPeriod', 5, ...
    'MaxEpochs',20, ...
    'ValidationData',validationSet, ...
    'ValidationFrequency',4, ...
    'MiniBatchSize', 64, ...
    'Plots', "training-progress")
```

```
disp ('break point 2')
```

```
convnet = trainNetwork(trainingSet, layers, options);

disp('break point 3')

disp('inspecter la premiere couche')

convnet.Layers(1)

disp('inspecter la derniere couche')

convnet.Layers(end)

% Display conv layer

w1 = convnet.Layers(2).Weights;
w1 = mat2gray(w1);
w1 = imresize(w1, 5);
figure
montage(w1)
title('Premiere couche de convolution')

featureLayer = 'softmax_1';
trainingFeatures = activations(convnet, trainingSet, featureLayer, ...
    'MiniBatchSize', 64, 'OutputAs', 'columns');

% Get training labels from the trainingSet
trainingLabels = trainingSet.Labels;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
% Train multiclass SVM classifier  
classifier = fitcecoc(trainingFeatures, trainingLabels, ...  
    'Learners', 'Linear', 'Coding', 'onevsall', 'ObservationsIn', 'columns');  
  
% Extract test features using the CNN  
testFeatures = activations(convnet, testSet, featureLayer, ...  
    'MiniBatchSize', 32, 'OutputAs', 'columns');  
  
% Pass CNN image features to trained classifier  
predictedLabels = predict(classifier, testFeatures, 'ObservationsIn', 'columns');  
  
% Get the known labels  
testLabels = testSet.Labels;  
  
% Confusion Matix  
confMat = confusionmat(testLabels, predictedLabels);  
  
[Result,RefereceResult]=confusion.getValues(confMat);  
  
Result
```

CNN-Binary Tree

```
NMCLS = 224
```

```
PTH = './IITD'
```

```
Sz = imsize('./IITD\001\001_1.BMP')
```

```
% Chargement du dataset
```

```
disp('chargement du dataset')
```

```
imds = imageDatastore(PTH, 'LabelSource', 'foldernames', 'IncludeSubfolders',true);
```

```
[trainingSet,validationSet, testSet] = splitEachLabel(imds,0.6, 0.2, 0.2 ...
```

```
...%399,49,49 ...
```

```
...%100,10,10 ...
```

```
, 'randomized');
```

```
layers = [
```

```
    imageInputLayer(Sz)
```

```
    convolution2dLayer(8,16,"Padding",1,"Stride",2)
```

```
    batchNormalizationLayer
```

```
    reluLayer('Name','relu_1')
```

```
    maxPooling2dLayer(4,"Stride",2)
```

```
    convolution2dLayer(4,16,"Padding",2)
```

```
batchNormalizationLayer
reluLayer('Name','relu_2')

maxPooling2dLayer(4,"Stride",2)
convolution2dLayer(3,32,"Padding",2)
batchNormalizationLayer
reluLayer('Name','relu_3')

maxPooling2dLayer(3,"Stride",2)
convolution2dLayer(3,32,"Padding",2)
batchNormalizationLayer
reluLayer('Name','relu_4')

maxPooling2dLayer(3,"Stride",2)
convolution2dLayer(3,64,"Padding",2)
batchNormalizationLayer
reluLayer('Name','relu_5')

fullyConnectedLayer(NMCLS)
softmaxLayer('Name','softmax_1')
classificationLayer
];

disp ('break point 1')

options = trainingOptions("sgdm", 'LearnRateSchedule', "piecewise", ...
    'LearnRateDropFactor', 0.2, ...
    'LearnRateDropPeriod', 5, ...
    'MaxEpochs',10, ...
    'ValidationData',validationSet, ...
```

```
'ValidationFrequency',4, ...
'MiniBatchSize', 64, ...
'Plots', "training-progress")

disp ('break point 2 ')

convnet = trainNetwork(trainingSet, layers, options);

disp ('break point 3')

disp('inspecter la premiere couche')

convnet.Layers(1)

disp('inspecter la derniere couche')

featureLayer = 'softmax_1';
trainingFeatures = activations(convnet, trainingSet, featureLayer, ...
    'MiniBatchSize', 64, 'OutputAs', 'columns');

% Get training labels from the trainingSet
trainingLabels = trainingSet.Labels;

% Train multiclass BT
classifier = fitctree(trainingFeatures', trainingLabels)

% Extract test features using the CNN
testFeatures = activations(convnet, testSet, featureLayer, ...
```



```
'MiniBatchSize', 32, 'OutputAs', 'columns');

% Pass CNN image features to trained classifier
predictedLabels = predict(classifier, testFeatures);

% Get the known labels
testLabels = testSet.Labels;

% Tabulate the results using a confusion matrix.
confMat = confusionmat(testLabels, predictedLabels);

[Result,RefereceResult]=confusion.getValues(confMat);
```

Result

CNN-KNN

```
NMCLS = 224
```

```
PTH = './IITD'
```

```
Sz = imsize('./IITD\001\001_1.BMP')
```

```
disp('chargement du dataset')
```

```
imds = imageDatastore(PTH, 'LabelSource', 'foldernames', 'IncludeSubfolders',true);
```

```
[trainingSet,validationSet, testSet] = splitEachLabel(imds,0.6, 0.2, 0.2 ...
```

```
...%399,49,49 ...
```

```
...%100,10,10 ...
```

```
, 'randomized');
```

```
layers = [
```

```
    imageInputLayer(Sz)
```

```
    convolution2dLayer(8,16,"Padding",1,"Stride",2)
```

```
    batchNormalizationLayer
```

```
    reluLayer('Name','relu_1')
```

```
%    maxPooling2dLayer(4,"Stride",2)
```

```
%    convolution2dLayer(4,16,"Padding",2)
```

```
%    batchNormalizationLayer
```

```
%    reluLayer('Name','relu_2')
```

```
%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   maxPooling2dLayer(4,"Stride",2)
%   convolution2dLayer(3,32,"Padding",2)
%   batchNormalizationLayer
%   reluLayer('Name','relu_3')
%
%   maxPooling2dLayer(3,"Stride",2)
%   convolution2dLayer(3,32,"Padding",2)
%   batchNormalizationLayer
%   reluLayer('Name','relu_4')
%
%   maxPooling2dLayer(3,"Stride",2)
%   convolution2dLayer(3,64,"Padding",2)
%   batchNormalizationLayer
%   reluLayer('Name','relu_5')

fullyConnectedLayer(NMCLS)
softmaxLayer('Name','softmax_1')
classificationLayer];

disp ('break point 1')

options = trainingOptions("sgdm", 'LearnRateSchedule', "piecewise", ...
    'Momentum', 0.9000,...
    'InitialLearnRate', 0.0100, ...
    'LearnRateSchedule', 'piecewise', ...
    'LearnRateDropFactor', 0.2000, ...
    'LearnRateDropPeriod', 5, ...
    'L2Regularization', 1.0000e-04, ...
    'GradientThresholdMethod', 'l2norm', ...
    'LearnRateDropFactor', 0.2, ...
```

```
'LearnRateDropPeriod', 5, ...
'MaxEpochs',20, ...
'ValidationData',validationSet, ...
'ValidationFrequency',1, ...
'MiniBatchSize', 64, ...
'Plots', "training-progress")

disp ('break point 2 ')

convnet = trainNetwork(trainingSet,layers,options);

disp ('break point 3')

disp('inspecter la premiere couche')

convnet.Layers(1)

disp('inspecter la derniere couche')

convnet.Layers(end)

% display conv layers

w1 = convnet.Layers(2).Weights;
w1 = mat2gray(w1);
w1 = imresize(w1,5);
figure
montage(w1)
```

```
title('Premiere couche de convolution')

featureLayer = 'softmax_1';
trainingFeatures = activations(convnet, trainingSet, featureLayer, ...
    'MiniBatchSize', 64, 'OutputAs', 'columns');

% learned labels
trainingLabels = trainingSet.Labels;

% Knn model
classifier = fitcknn(trainingFeatures', trainingLabels, 'NumNeighbors', 1, 'Standardize', 1)

% extraction des caract 
testFeatures = activations(convnet, testSet, featureLayer, ...
    'MiniBatchSize', 32, 'OutputAs', 'columns');

% prediction using cnn labels
predictedLabels = predict(classifier, testFeatures);

% observed labels
testLabels = testSet.Labels;

% Confusion Matix
confMat = confusionmat(testLabels, predictedLabels);

[Result, RefereceResult]=confusion.getValues(confMat);

Resultat
```