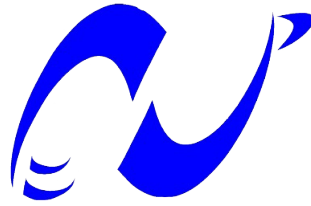


RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

ÉCOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

DÉPARTEMENT D'ÉLECTRONIQUE
MÉMOIRE DE PROJET DE FIN D'ÉTUDES
AU VU DE L'OBTENTION DU DIPLOME D'INGÉNIEUR D'ÉTAT EN
ÉLECTRONIQUE

**Apport de l'apprentissage profond pour la prédiction
du genre à partir de l'écriture manuscrite**

Ryad Lotfi MAHTAL

Khalil MEFTAH

Sous la supervision de

**Dr. Nesrine BOUADJENEK, Maître de Conférence B à l'Ecole
Nationale Polytechnique**

Présenté et soutenu publiquement le 31 Août 2020

Composition du Jury :

Rachida TOUHAMI	Professeur	ENP	Présidente
Nesrine BOUADJENEK	MCB	ENP	Superviseur
Nour El Houda BENALIA	MCB	ENP	Examinatrice

ENP 2020

École Nationale Polytechnique (ENP)
10, Avenue des Frères Oudek, Hassen Badi, BP. 182, 16200 El Harrach, Alger, Algérie

www.enp.edu.dz

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

ÉCOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

DÉPARTEMENT D'ÉLECTRONIQUE
MÉMOIRE DE PROJET DE FIN D'ÉTUDES
AU VU DE L'OBTENTION DU DIPLOME D'INGÉNIEUR D'ÉTAT EN
ÉLECTRONIQUE

**Apport de l'apprentissage profond pour la prédiction
du genre à partir de l'écriture manuscrite**

Ryad Lotfi MAHTAL

Khalil MEFTAH

Sous la supervision de

**Dr. Nesrine BOUADJENEK, Maître de Conférence B à l'Ecole
Nationale Polytechnique**

Présenté et soutenu publiquement le 31 Août 2020

Composition du Jury :

Rachida TOUHAMI	Professeur	ENP	Présidente
Nesrine BOUADJENEK	MCB	ENP	Superviseur
Nour El Houda BENALIA	MCB	ENP	Examinatrice

ENP 2020

École Nationale Polytechnique (ENP)
10, Avenue des Frères Oudek, Hassen Badi, BP. 182, 16200 El Harrach, Alger, Algérie

www.enp.edu.dz

ملخص

الكتابة اليدوية هي مقياس حيوي شخصي يميز كل فرد بطريقة فريدة، فهي تسمح بالحصول على العديد من المعلومات الخاصة بكتابتها مثل الجنس. في هذا العمل أنشأنا نظاما آليا للتنبؤ بالجنس من الكتابة اليدوية. تحقيقا لهذه الغاية، نقترح أولا إنشاء صور تكستريز لكل جنس إنطلاقا من عينات الكتابة اليدوية. نقوم بعد ذلك بربط هذه الصور بشبكة عصبية تلافيفية (CNN) من أجل استخراج الخصائص الخاصة بكل جنس. أخيرا، نقترح استخدام مرشحات Gabor مع الشبكات العصبية التلافيفية لاستخراج الميزات الخاصة بكل نوع بشكل أفضل. يتم إجراء الاختبارات المختلفة على قاعدتي بيانات عامة. النتائج التي تم الحصول عليها مرضية للغاية وواعدة. الكلمات المفتاحية : التنبؤ بجنس الكاتب، الكتابة اليدوية، التكستريز، CNN، مرشحات Gabor.

Abstract

Handwriting is a personal behavioral biometry that characterizes each individual in a unique way. It allows to obtain several information specific to its author such as gender. In this work we propose an automatic gender prediction system from handwriting using deep learning. Therefore, we first propose to generate textured images by gender from handwriting's samples. We will then associate these images to a Convolutional Neuron Network (CNN) in order to extract the characteristics specific to each gender. Finally, we propose to use Gabor filters with the CNNs for a better feature extraction. The different tests are performed on two benchmark datasets. The obtained results are very satisfactory and promising.

Keywords : Writer's gender prediction, handwriting, texture, CNN, Gabor filters.

Résumé

L'écriture manuscrite est une biométrie comportementale personnelle qui caractérise l'individu d'une façon unique. Elle permet d'obtenir plusieurs informations propre à son auteur tel que le genre. Dans ce travail, nous proposons un système de prédiction automatique du genre à partir de l'écriture manuscrite en utilisant l'apprentissage profond. A cet effet, nous proposons en premier lieu de générer des images textures par genre à partir des échantillons d'écriture manuscrite. Nous associerons ensuite ces images à un Réseau de Neurones Convolutifs (CNN) afin d'extraire les caractéristiques propres à chaque genre. Enfin, nous proposons d'utiliser les filtres de Gabor avec les CNNs pour une meilleure extraction de caractéristiques. Les différents tests sont effectués sur deux bases de données standards. Les résultats obtenus sont très satisfaisants et prometteurs.

Mots-clés : Prédiction du genre de l'écrivain, écriture manuscrite, texture, CNN, filtres de Gabor.

Remerciement

Tout d'abord nous voudrions remercier Dieu, le tout-puissant, pour nous avoir donné la force, la patience, la volonté et toutes ces bénédictions.

Nous remercions tout particulièrement Madame Nesrine Bouadjnek, Docteur à l'École Nationale Polytechnique, d'avoir accepté de nous encadré pendant toute la longue période de préparation de ce mémoire. Nous la remercions vivement pour son aide, sa patience et ses précieux conseils qui nous ont grandement aidés.

Nous voulons remercier Madame Rachida TOUHAMI, professeur à l'École Nationale Polytechnique d'avoir accepté de présider notre jury, ainsi que Madame Nour El Houda BENALIA, maître conférence B à l'École Nationale Polytechnique d'avoir accepté d'être membres du Jury.

Nous remercions tous les professeurs qui nous ont enseignés pendant ces trois années de spécialités ainsi qu'au cycle préparatoire. C'est grâce au savoir qu'ils nous ont transmis que nous avons pu accomplir ce travail.

Enfin, nous remercions toutes personne qui nous a aidé de loin ou de près et de n'importe quelle manière à réaliser ce travail.

Dédicaces

Je dédie ce travail particulièrement à la lumière de ma vie, ma mère, qui m'a soutenu depuis le début de ma scolarité, m'a accompagné durant toute épreuve, qui a toujours su me remonter le moral et qui a su inculquer en moi des valeurs qui font aujourd'hui l'homme que je suis.

A mon père, l'homme fort qui m'a toujours dirigé vers la bonne voie, qui a toujours su me remonter le moral et m'apprendre le sens réel de la vie tout en y apportant un équilibre qui fait de moi l'homme que je suis aujourd'hui. C'est mon premier professeur de mathématique, il m'a enseigné les bases de la matière depuis mon jeune âge.

A ma chère soeur Tinhinene, avec qui j'ai passé toute mon enfance et celle qui m'a inspiré de par son parcours académique et professionnel. Elle a toujours su m'apporter les conseils d'une grande soeur, a toujours été là pour m'épauler, je te souhaite beaucoup de bonheur.

A ma petite soeur adorée Lyna, celle qui ne cesse de me taquiner et m'apporter sa joie de vivre. Je te souhaite tout le bonheur et la réussite dans ton parcours scolaire.

A mon oncle Sidali, un homme de principe, celui qui répond toujours présent en cas de besoin, il continue de m'inspirer par son excellent parcours professionnel.

A ma tante Anissa, ma deuxième maman, celle qui m'a élevé, par sa gentillesse et son amour.

A mes amis, Mohamed (Bouguerra et Saieh), Adel, Raien, Abderrahman, Abdou, Zinou, Khaled, Chakib, Tarek, et Cherif, ceux avec qui j'ai passé des moments inoubliables et incroyables. Ceux que leur présence a été un facteur essentiel pour que je puisse passer à travers les moments difficiles. Ils étaient toujours présents dans les meilleurs moments comme des les pires.

A mon amie d'enfance Sonia, celle que je connais depuis plusieurs années, merci d'être là pendant toute cette période. Merci de m'avoir toujours aidé dans les moments difficiles de ma vie.

A mes amis du CAP, Adlene, Aimad, Ishak, Dalil, Yanis, Anis, Malek, Zaki, Anes, merci de m'avoir permis de vivre des expériences uniques et inoubliables, merci pour toute la joie que vous m'avez apporté.

A mes camarades de classes, Lina, Rayane, Yacine, Oussama, Imene, Sarah, Khadidja, Hacene, Abdou, Cylia, Hanane, merci pour ces trois belles années.

A mes amis de la prépa, Kahina, Madina, Badis, Lamine, Moh, Khoukha, Abir, Houda, Chakib, Cyrine, Aghiles, Rokho, et mohamed, grâce à vous, j'ai aimé la prépa.

Et enfin, à mon binôme, Khalil, qui a su me soutenir tout au long de ce travail, qui a fait preuve de persévérance et d'assiduité. Merci d'avoir contribué grandement à ce travail.

Ryad Lotfi

Dédicaces

Je dédie ce travail

A ma mère ...

A mon père ...

A mes frères ...

A toute la famille MEFTAH ...

A tous mes amis ...

A tous les membres de l'ENP.

Khalil

Table des matières

Liste des figures

Liste des tableaux

Liste des abréviations

Introduction	13
1 La prédiction du genre à partir de l'écriture manuscrite	15
1.1 Introduction	16
1.2 L'écriture manuscrite comme caractéristique biométrique comportementale	16
1.3 Pourquoi la prédiction du genre à partir de l'écriture manuscrite?	18
1.4 Différences géométriques et visuelles entre l'écriture masculine et féminine .	18
1.5 Architecture d'un système de prédiction du genre à partir de l'écriture manuscrite	20
1.6 Aperçu sur l'état de l'art	22
1.7 Conclusion	23
2 Prétraitement des données : De l'image brute à l'image texture	25
2.1 Introduction	26
2.2 L'analyse de texture, quel intérêt?	26
2.3 Segmentation des lignes en mots	27
2.3.1 Binarisation des images	28
2.3.2 Complément d'image	29
2.3.3 Erosion et dilatation	29
2.3.4 Histogramme de projection horizontale et verticale	32
2.3.5 Le filtrage des petits segments	33
2.3.6 Redimensionnement des mots	33
2.4 Génération des images texturées	34
2.4.1 Centre de masse	34
2.4.2 Procédure de génération de l'image texture	35
2.5 Segmentation et filtrage des images textures	36

2.6	Conclusion	37
3	Réseaux de neurones convolutifs	39
3.1	Introduction	40
3.2	Historique	40
3.2.1	Notion de neurone formel	41
3.2.2	Perceptron mono-couche	41
3.2.3	Perceptron multi-couche	42
3.3	Réseaux de neurones convolutifs (CNNs)	44
3.3.1	Bloc d'extraction de caractéristiques	44
3.3.2	Bloc de classification	49
3.3.3	L'entraînement	50
3.4	Régularisation	51
3.5	Les hyperparamètres	52
3.6	Architecture adoptée	53
3.7	Extreme Learning Machine	55
3.7.1	Filtre de Gabor	55
3.8	Implémentation du Filtre de Gabor dans l'architecture CNN	57
3.9	Conclusion	58
4	Tests et résultats expérimentaux	59
4.1	Introduction	60
4.2	Description des bases de données	60
4.2.1	La base KHATT	60
4.2.2	La base IAM	61
4.3	Logiciels, bibliothèques et matériel utilisés	61
4.3.1	Python	62
4.3.2	OpenCV	62
4.3.3	TensorFlow	62
4.3.4	Keras	62
4.3.5	Google Colab	63
4.3.6	Matériel utilisé	63
4.4	Critères d'évaluation	63
4.4.1	Le taux de bonne prédiction	63
4.4.2	Matrice de confusion	64
4.4.3	Courbe ROC	64
4.4.4	AUC : aire sous la courbe ROC	65
4.5	Protocole expérimental	65
4.5.1	Répartition des données	66

4.5.2	Influence de la taille du bloc texture et l'espace inter-lignes sur les performances du modèles	67
4.5.3	Application des filtres de Gabor sur les images textures	69
4.5.4	Le réglage des hyperparamètres	70
4.6	Résultats obtenus	73
4.6.1	CNN sans filtre de Gabor	73
4.6.2	CNN avec filtre de Gabor	76
4.6.3	Comparaison avec les résultats de l'état de l'art	81
4.7	Conclusion	82

Conclusion **83**

Table des figures

1.1	Classification des différents types de biométrie	17
1.2	Échantillons d'écriture féminine et masculine en langue Anglaise et Arabe .	19
1.3	Architecture d'un système de prédiction du genre à partir de l'écriture manuscrite	20
2.1	Blocs textures	27
2.2	Processus de segmentation de la ligne en mots	28
2.3	Binarisation d'un mot par la méthode Otsu	29
2.4	Complément d'image	29
2.5	Éléments structurants ou masques de convolution	30
2.6	Élément structurant rectangulaire de taille 3 x 5	31
2.7	Dilatation morphologique	31
2.8	Érosion morphologique	32
2.9	Segmentation d'une ligne en mots	33
2.10	Exemples d'images aberrantes	33
2.11	Histogramme des hauteurs des segments	34
2.12	Exemple de redimensionnement d'un mot	34
2.13	Centre de masse d'un mot	35
2.14	Remplissage de l'image canevas	35
2.15	Images texture	36
2.16	Blocs de texture	36
2.17	Exemples de blocs de texture inutiles	37
2.18	Densité de pixels des blocs de texture (seuil en rouge 0.14)	37
3.1	Neurone formel	42
3.2	Perceptron mono-couche	42
3.3	Perceptron multi-couches	43
3.4	Blocs des couches composant un CNN	44
3.5	Processus de la couche de convolution	46
3.6	Les couches convolutives forment une hiérarchie spatiale des objets visuels : les bords hyperlocaux se combinent en objets locaux tels que les yeux ou les oreilles, qui se combinent en concepts de haut niveau tels que «chat» . .	47

3.7	Les différentes fonctions d'activations	48
3.8	Fonctions de pooling	49
3.9	Couche <i>Flatten</i>	49
3.10	Illustration du Dropout lors de la phase d'entraînement	52
3.11	Architecture adoptée	54
3.12	Filtre de Gabor	56
3.13	Masque du filtre de Gabor de rayon 21 pixels pour $\theta = 0^\circ$, $\sigma = 7$	56
3.14	Filtre de Gabor appliqué sur une image texture	57
3.15	Filtre de Gabor dans l'architecture CNN	58
4.1	Lignes de texte manuscrites issues de la base KHATT	60
4.2	Lignes de texte manuscrites issues de la base IAM	61
4.3	Matrice de Confusion	64
4.4	AUC (aire sous la courbe ROC)	65
4.5	Images textures générées à partir de différents espaces inter-lignes	68
4.6	Application des filtres de Gabor sur les images textures	70
4.7	Recherche des paramètres optimaux	72
4.8	Les filtres de Gabor utilisés	73
4.9	Taux de bonne prédiction d'entraînement et de validation pour la base KHATT (*100%)	74
4.10	Taux de bonne prédiction d'entraînement et de validation pour la base IAM (*100%)	74
4.11	Erreur commise d'entraînement et de validation pour la base KHATT	75
4.12	Erreur commise d'entraînement et de validation pour la base IAM	75
4.13	Taux de bonne prédiction d'entraînement et de validation pour la base KHATT (*100%)	77
4.14	Taux de bonne prédiction d'entraînement et de validation pour la base IAM (*100%)	77
4.15	Erreur commise d'entraînement et de validation pour la base KHATT	78
4.16	Erreur commise d'entraînement et de validation de pour base IAM	78
4.17	Comparaison des courbes ROC pour la base KHATT	79
4.18	Comparaison des courbes ROC pour la base IAM	80
4.19	CNN vs CNN-Gabord pour la prédiction du genre	81

Liste des tableaux

1.1	Résumé des travaux de la littérature les plus récents	23
4.1	Répartition de la base de données KHATT	66
4.2	Répartition de la base de données IAM	67
4.3	Performances en terme de TBP (%) pour les différentes tailles des blocs textures - base KHATT	68
4.4	Performances en terme de TBP (%) pour les différentes tailles des blocs textures - base IAM	69
4.5	Les différentes valeurs testées des hyperparamètres pour les deux bases de données	71
4.6	Matrice de confusion pour la base KHATT (%)	76
4.7	Matrice de confusion pour la base IAM (%)	76
4.8	Matrice de confusion pour la base KHATT (%)	80
4.9	Matrice de confusion pour la base IAM (%)	80
4.10	Comparaison des performances pour la prédiction du genre à partir de l'écriture manuscrite pour les deux base de données KHATT et IAM	81

Liste des abréviations

ADN	Acide D ésoxyribonucléique
API	Application P rogramming I nterface
AUC	Area U nder the C urve
CIFAR	Canadian I nstitute F or A dvanced R esearch
CNN	Convolutional N eural N etwork
ECB	Entropie C roisée B inaire
EC	Entropie C roisée
ELM	Extreme L earning M achine
FC	Fully C onnected
GMM	Gaussian M ixture M odel
HOG	Histogram of O riented G radients
KHATT	Kfupm H andwritten A rabic T ex T
KPPV	K -Plus P roches V oisins
MLP	Multi L ayer P erceptron
oBIFs	Oriented B asic I mage F eatures
PFSA	Probabilist F inite S tate A utomata
ReLU	Rectified L inear U nit
ROC	Receiver O perating C haracteristic
SLP	Single L ayer P erceptron
SVM	Support V ector M achine
TBP	Taux de B onne P rediction

Introduction

Depuis plusieurs décennies, l'écriture manuscrite est utilisée pour identifier les personnes et les différencier les uns des autres. Ceci dit, chaque individu possède son propre style d'écriture, par conséquent, il est très difficile d'imiter parfaitement l'écriture d'une personne au point de ne pas pouvoir les différencier, rendant ainsi l'écriture manuscrite une caractéristique très personnelle et son exploitation en tant que biométrie comportementale très utile. En outre, l'avancée technologique qu'a connu le monde ces derniers temps a permis d'évoluer plusieurs domaines de recherches, nous citons, entre autres, la graphologie. La graphologie, apparue dans les années 1900, est une technique d'analyse de l'écriture manuscrite, elle consiste en l'étude de la forme des lettres et de l'allure générale de l'écriture afin de déterminer l'identité du scripteur ou d'observer son caractère et sa personnalité à travers son écriture. L'évolution de ce domaine a permis l'introduction de système biométrique se basant sur l'utilisation des différentes caractéristiques tirées de l'écriture manuscrite en tant qu'identificateur [10]. En effet, ces caractéristiques biométriques manuscrites se divisent en deux catégories :

1. Les caractéristiques biométriques dures (en anglais hard-biometrics), sont celles qui permettent d'identifier une personne d'une manière unique, comme la signature manuscrite,
2. Les caractéristiques biométriques douces (en anglais soft-biometrics), sont celles qui permettent de fournir un ensemble de caractéristique générales appartenant à une personne mais qui ne permettent pas de l'identifier d'une façon unique, tels que le genre, l'âge, la manualité et la nationalité [40].

L'analyse de l'écriture manuscrite présente de nombreuses difficultés produites par les variations interpersonnelles et intrapersonnelles lors de l'écriture : la nature cursive de l'écriture manuscrite, l'utilisation de différents types de stylos ou la présence de papier à fond bruyant [40].

Dans ce travail, nous nous proposons d'extraire une information biométrique douce, très utile et assez importante pour diverses applications : Le Genre. Le travail consiste d'abord à appliquer un pré-traitement des échantillons d'écriture manuscrite qui vise à générer une image texture par genre à partir des images brutes. Par la suite, ces textures seront exploitées par un réseau de neurones convolutifs afin de différencier automatiquement entre l'écriture masculine et l'écriture féminine. Le mémoire est divisé en quatre

chapitres structurés comme suit :

- Le chapitre 1 introduit l'écriture manuscrite comme biométrie comportementale, ainsi que le contexte générale de la prédiction du genre à partir de l'écriture manuscrite, en donnant un aperçu sur l'état de l'art.
- Le chapitre 2 est consacré au pré-traitement appliqué qui est la génération de l'image texture à partir d'échantillons d'écriture manuscrite.
- Le chapitre 3 fournit l'arrière-plan théorique des réseaux des neurones artificiels, ainsi que les réseaux de neurones convolutifs utilisés pour la prédiction du genre à partir de l'écriture manuscrite.
- Le dernier chapitre présente les tests et les résultats obtenus par l'application de l'image texture associée aux réseaux de neurones convolutifs pour la prédiction du genre à partir de l'écriture manuscrite.

Enfin, nous achevons le travail par une conclusion générale et quelques perspectives.

Chapitre 1

La prédiction du genre à partir de l'écriture manuscrite

1.1 Introduction

Depuis des années, l'écriture manuscrite a été utilisée pour identifier une personne d'une façon unique. En effet, chaque individu a son propre style d'écriture; il peut être copié mais jamais égalé. Même les jumeaux identiques qui partagent l'apparence et la génétique n'ont pas la même écriture. Ainsi, l'analyse de l'écriture manuscrite est devenue un domaine de recherche attractif pour les examinateurs médico-légaux de documents, les paléographes¹, les graphologues et les analystes légistes. Un intérêt tout particulier est porté à la détermination du genre d'un individu à partir de son écriture manuscrite, étant donné que, plusieurs études psychologiques ont montré qu'il existe bel et bien une différence entre l'écriture d'une femme et l'écriture d'un homme [19, 4]. Par conséquent, les chercheurs de la communauté de la reconnaissance de formes ont commencé à s'intéresser de plus en plus à la prédiction automatique du genre à partir de l'écriture manuscrite, où le premier travail présenté par Liwicki et al., [35] est apparu en 2007. Dans ce chapitre, nous exposons l'intérêt de prédire le genre à partir de l'écriture manuscrite, ainsi que les caractéristiques visuelles et géométriques caractérisant l'écriture de chaque genre. Par la suite, nous présentons l'architecture d'un système de prédiction du genre à partir de l'écriture manuscrite et enfin nous survolons l'état de l'art pour construire une idée sur les techniques utilisées dans ce domaine.

1.2 L'écriture manuscrite comme caractéristique biométrique comportementale

La biométrie désigne l'ensemble des procédés de reconnaissance, d'authentification et d'identification d'une personne par certaines de ses caractéristiques physiques ou comportementales qui lui sont propres. Ces caractéristiques doivent être *universelles*, pour être utilisable partout dans le monde, *uniques*, pour distinguer les personnes sans équivoque, *enregistrables* pour permettre de les stocker, et *mesurables* pour permettre de les comparer.

Comme le montre la figure 1.1, les caractéristiques biométriques peuvent se regrouper en trois catégories : biologiques, morphologiques et comportementales. Les caractéristiques biologiques concernent l'*ADN* et le sang, tandis que les caractéristiques morphologiques comportent la forme de la main, les empreintes digitales, le visage, l'iris, etc. Enfin, les caractéristiques comportementales concernent la démarche, la gestuelle, la signature manuscrite, etc.

1. Spécialiste des écritures anciennes

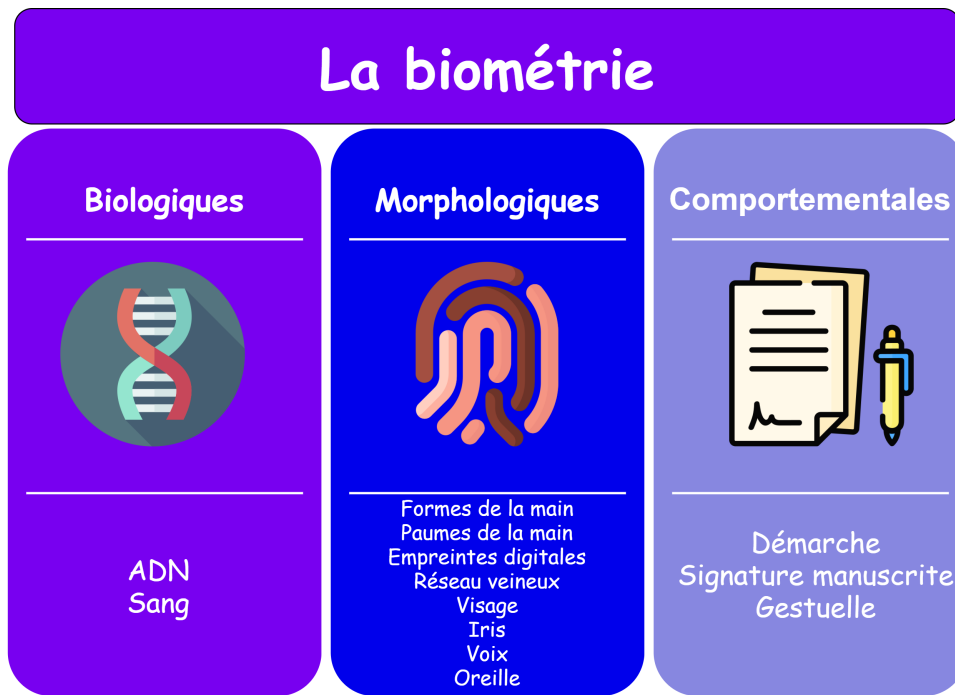


FIGURE 1.1: Classification des différents types de biométrie

La biométrie comportementale est liée à des caractéristiques comportementales dynamiques, par exemple la façon de bouger, la gestuelle, la démarche, la façon de taper sur un clavier, etc., qui s'avèrent être des caractéristiques personnelles et propres à chaque individu. L'exemple de la biométrie comportementale le plus couramment utilisé est la signature manuscrite, cependant, les progrès technologiques permettent d'identifier d'autres caractéristiques comportementales personnelles et de les utiliser pour aider à mieux identifier une personne, nous citons, entre autres, l'écriture manuscrite. En effet, il existe d'innombrables travaux qui abordent l'identification et/ou l'authentification d'un individu à partir de son écriture manuscrite.

Généralement, les caractéristiques dites soft-biométriques sont extraites à partir du corps ou du visage. Cependant l'écriture manuscrite étant une caractéristique soft-biométrique comportementale, universelle, et personnelle à chaque individu dans le sens où chaque personne a son propre style d'écriture, les spécialistes se sont penchés sur la question suivante : *Quel type d'information biométrique pourrait fournir une écriture manuscrite ?* A partir de l'écriture manuscrite, plusieurs informations peuvent être tirées. Dans [4], les auteurs ont étudié la relation entre les hormones biologiques spécifiques à chaque genre et le style d'écriture manuscrite. Les résultats obtenus ont montré que les hormones prénatales peuvent affecter l'écriture des femmes. Aussi, lors de certaines enquêtes psychologiques antérieures, les différences entre l'écriture manuscrite des hommes et celle des femmes ont été examinées [21, 22]. Aujourd'hui, plusieurs caractéristiques soft-biométriques sont prédites à partir de l'analyse de l'écriture manuscrite tels que le genre, l'âge, la nationalité, et la manualité [40, 7, 17].

1.3 Pourquoi la prédiction du genre à partir de l'écriture manuscrite ?

La prédiction du genre à partir de l'écriture manuscrite a de nombreuses applications, et facilite grandement les recherches ou enquêtes dans plusieurs domaines différents. Par exemple, la prédiction du genre à partir d'une écriture manuscrite aide dans les enquêtes criminalistiques à authentifier l'assassin ou même l'auteur d'une lettre de suicide. Cela peut paraître abstrait, mais imaginons que sur une scène de crime, un papier écrit à la main soit trouvé. L'analyse de ce bout de papier, et l'information du genre qui est tirée depuis, peut aider l'enquêteur à arriver à sa cible plus rapidement. En effet, en réduisant le groupe de suspects à "femmes" ou "hommes", cela permettra à l'enquêteur d'accélérer davantage sa procédure. Aussi, la prédiction du genre à partir de l'écriture manuscrite peut s'avérer très utile dans les documents historiques dont l'auteur est inconnu. En effet, de nombreuses références historiques restent jusqu'à nos jours inconnus. L'identification du genre de l'auteur d'un document historique présente une étape importante pour l'identification de l'écrivain.

De nos jours, les tentatives de fraudes et d'utilisation de faux papiers ne cesse d'augmenter, et les technologies pour y faire face avec. Une approche qui peut être utilisée pour confirmer l'identité d'une personne ayant soumis un document écrit à la main, de haute sensibilité, est de vérifier le genre à partir de son écriture. Bien évidemment, d'autres vérifications doivent s'aligner avec celle mentionnée, mais l'unification de toutes ces vérifications permet de confirmer si la personne prétendant être le propriétaire du document, l'est vraiment. Dans le même cadre, la vérification du genre à partir de l'écriture manuscrite peut constituer un élément clé pour vérifier le propriétaire des chèques bancaires. Par exemple, en vérifiant l'identité de la signature présente avec le genre de la personne ayant écrit le montant à virer, nous pouvons nous assurer que l'opération de virement est faite avec le consentement du propriétaire. Une autre utilisation peut aussi être imaginée pour vérifier l'authenticité des différentes procurations. Une procuration est un document par lequel nous autorisons autrui à agir à notre place. Nous comprenons toute suite la sensibilité d'un tel document. En vérifiant le genre de celui qui a écrit la procuration, et avec la vérification d'autres paramètres, nous pouvons nous assurer que le document autorise bel et bien la personne présente sur place à agir.

1.4 Différences géométriques et visuelles entre l'écriture masculine et féminine

Les différences entre la femme et l'homme sont diverses et multiples. Elles peuvent être biologiques ou physiologiques. Ceci entraîne donc une différence dans les biométries

comportementales, notamment l'écriture manuscrite.

Généralement, la différence entre l'écriture masculine et féminine peut se voir visuellement. La figure 1.2 représente une écriture féminine et masculine, dans les deux langues : Anglaise et Arabe. Ainsi, les chercheurs ont été confrontés à la question suivante : *Existe-t-il réellement une différence entre l'écriture masculine et féminine ?*

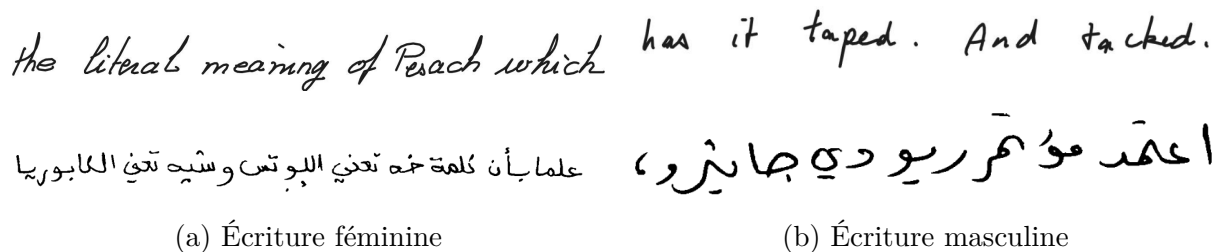


FIGURE 1.2: Échantillons d'écriture féminine et masculine en langue Anglaise et Arabe

La première étude qui a traité l'éventualité de l'existence d'une relation entre le genre et la graphologie a été faite en 1903 par *Alfred Binet* [6]. Les observations sur la façon dont le genre de l'écrivain pouvait être deviné, ont été réalisées au moyen d'enveloppes remises au sujet, dont certaines portaient des écritures masculines, tandis que d'autres indiquaient des écritures féminines. Sur 100 observations, les proportions de devinettes correctes étaient respectivement de 75% pour les experts (graphologues), 73% pour les enseignants et de 63% pour les jeunes filles de 18 ans sans instruction. *Crépieux-Jamin*, un des graphologues concernés par la précédente expérience, a tenté d'expliquer sa méthode de classification. Il a souligné que, pour le genre, il y avait différents indicateurs. Parfois, c'est la psychologie de l'écrivain qui l'informe, parfois c'est directement la forme du geste manuscrit. Dans sa démarche, il a fait appel à :

- Une écriture qui est nette, simple, précipitée, inégale, et irrégulière pour l'homme en contraste avec une écriture bien complexe, soignée, régulière, uniforme et bien organisée pour l'écriture de la femme ;
- Des détails graphiques tels que la surélévation du "r" et du "s" dans l'écriture des femmes. La simplification, qui est un signe graphologique associé à la culture, est beaucoup plus fréquente chez les hommes que chez les femmes.

Cette première étude, a confirmé, statistiquement et avec des échantillons représentatifs, qu'une relation existe bel et bien entre le genre et le style d'écriture.

D'autres travaux ont été menés sur la prédiction du genre à partir de l'écriture manuscrite. Le travail de *M. Spear* [47] s'appuie sur des données tirées des déclarations des enseignants sur ce qu'ils pensent peut être la différence entre les travaux écrits par des garçons et ceux écrits par des filles. Cependant, quelles preuves est là pour montrer que l'information du genre est aussi présente dans l'écriture manuscrite des enfants? *James Hartley* était le premier à avoir établi une étude sur des enfants dont l'âge varie de 7 à

8 ans [21]. L'étude a été faite sur 22 enfants (10 garçons et 12 filles). Il leur a tous été demandé d'écrire le même passage. Il leur a été demandé après de réécrire le passage, mais cette fois-ci, en essayant d'imiter l'écriture du sexe opposé. Il a ensuite été demandé à d'autres enfants d'essayer de prédire le genre de l'auteur de chaque paragraphe. Les résultats ont montré que les garçons prévoient correctement les écritures provenant des garçons avec un taux de 75% et de 53% pour les écritures provenant des filles. Les filles, quant à elles, prévoient correctement les écritures provenant des garçons avec une précision de 66% contre 77% pour les écritures provenant des filles. Pour les scripts imités, les résultats ont considérablement diminué. Le travail de *James Hartley* vient appuyer alors le fait qu'une relation réellement existe entre le genre et l'écriture manuscrite et ce, quelque soit l'âge, et qu'il est très difficile d'imiter la façon d'écrire d'une autre personne, appuyant ainsi le fait que l'écriture manuscrite soit une biométrie très personnelle. Tous les travaux et recherches mentionnés ci-dessus montrent et confirment l'existence réelle de l'information du genre dans l'écriture manuscrite. En effet, tous les travaux, et beaucoup d'autres qui n'ont pas été cités, convergent vers le même résultat.

1.5 Architecture d'un système de prédiction du genre à partir de l'écriture manuscrite

Comme tout problème de reconnaissance de formes, la prédiction du genre à partir de l'écriture manuscrite suit plusieurs étapes, allant du pré-traitement des données à la classification. Notre problème consiste à associer à une écriture manuscrite, portée sur un papier ordinaire, le genre de son auteur. Pour ce faire, nous devons passer par plusieurs étapes. La figure 1.3 résume les étapes suivies pour la prédiction automatique du genre à partir de l'écriture manuscrite.

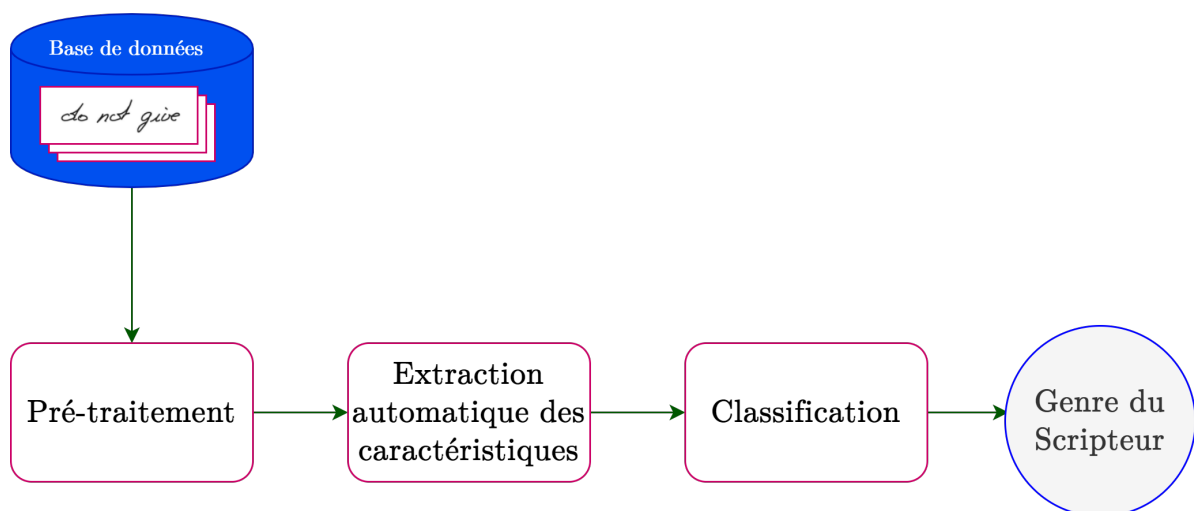


FIGURE 1.3: Architecture d'un système de prédiction du genre à partir de l'écriture manuscrite

Nous expliquons dans ce qui suit, en bref, l'intérêt de chaque étape.

- 1. Le pré-traitement de données :** L'étape de pré-traitement est une étape primordiale dans tout problème de traitement d'images. Le but est d'améliorer la lisibilité de l'image contenant le texte et de supprimer les détails qui n'ont pas de pouvoir discriminatoire dans le processus de prédiction. Effectivement, les documents sont souvent numérisés à l'aide d'un scanner. Ainsi, la qualité du document numérisé est influencée par plusieurs facteurs à savoir la performance du scanner, et la résolution. De plus, une image provenant d'un scanner est différente d'une image provenant d'un appareil à photo. Le pré-traitement des images comporte, sans s'y limiter :
 - La binarisation : Cette étape sert à isoler l'écriture du fond du document (arrière-plan). L'application de cette technique sur une image en niveaux de gris donne une image binaire où chaque pixel peut avoir une valeur 0 ou 1.
 - La segmentation : La segmentation englobe plusieurs traitements permettant d'extraire les entités du texte ciblées, tels que les lignes, les mots ou les fragments.
 - Le redimensionnement : Après la segmentation, les images possèdent systématiquement des tailles variables. Cette étape consiste à fixer une limite supérieure pour les images segmentées afin de ne pas se retrouver avec des images très grandes. Ceci assurera, entre autres, l'homogénéité de l'ensemble des images.
- 2. Extraction de caractéristiques :** L'extraction ou la génération de caractéristiques implique la réduction du nombre de ressources nécessaires pour décrire un grand ensemble de données à travers un descripteur. En effet l'extraction de caractéristiques permet, d'une part, la réduction de dimensionnalité, mais aussi, elle met en évidence les informations pertinentes contenues dans les données. Lors de l'analyse de données complexes, l'un des problèmes majeurs provient du nombre de variables impliquées. L'analyse avec un grand nombre de variables nécessite généralement une grande quantité de mémoire et de puissance de calcul. Elle peut également amener l'algorithme de classification à s'adapter aux échantillons d'apprentissage et à se généraliser mal aux nouveaux échantillons. De nombreux spécialistes du traitement d'images pensent qu'une extraction des caractéristiques correctement optimisée est la clé d'une construction efficace du modèle [15]. Plusieurs méthodes générant différents types de descripteurs ont été proposées pour la prédiction du genre. Ces descripteurs sont généralement classés en deux catégories ; statiques et pseudo-dynamiques, selon l'information qu'ils mettent en évidence. Les descripteurs statiques rehaussent la forme géométrique de l'écriture, telle que la hauteur, la largeur, les courbures, la surface du texte, les espaces intra-lettres et intra-mots, etc [45, 2]. Alors que les descripteurs pseudo-dynamiques sont basées sur les informations topologiques, de gradient ou de texture [7, 8].

3. **Classification** : La classification consiste à attribuer une étiquette de classe à un ensemble d'instances non classifiées. Dans le cas de la classification supervisée, où un ensemble de données étiquetées est fourni, l'objectif de la classification est d'analyser les données d'entrée afin de développer une description ou un modèle précis pour chaque classe, en utilisant un algorithme d'apprentissage spécifique. En fait, l'algorithme d'apprentissage se rapproche d'une fonction cible en mappant les variables d'entrée aux variables de sortie selon une règle de décision. C'est ce qui définit un classifieur. La littérature sur la prédiction du genre à partir de l'écriture manuscrite rapporte l'utilisation de plusieurs classifieurs tels que k-Plus Proches Voisins (KPPV) [14], Modèle de Mélange Gaussien (Gaussian mixture model : GMM) [34], les réseaux de neurones artificiels [15] et les Machines à Vecteur de Support (Support vector machine : SVM) [7, 17]. De nos jours, la tendance est d'utiliser l'apprentissage profond pour la prédiction du genre [26, 38].

1.6 Aperçu sur l'état de l'art

La prédiction du genre à partir de l'écriture manuscrite est un sujet de recherche qui a intéressé les scientifiques depuis le début des années 1900 [21, 22]. Aujourd'hui, avec l'avancement technologique, le développement de la puissance de calcul des ordinateurs et les différents besoins de prédire le genre à partir de l'écriture manuscrite, ce domaine de recherche a suscité l'intérêt de la communauté de la reconnaissance de formes, et plusieurs travaux ont été publiés dans ce sens. Nous citons dans ce qui suit les travaux les plus récents.

En 2016, Bouadjenek et al., [7] ont proposé la combinaison de trois systèmes basés sur une règle de combinaison floue (Min-Max), où la meilleure performance obtenue est de 82% . Par la suite, en 2017, Akbari et al., [1] ont proposé de convertir l'image contenant l'écriture manuscrite en une image texture et en construisant un automate d'état fini probabiliste (Probabilist Finite State Automata : PFSA) afin de générer les vecteurs caractéristiques. La meilleure performance obtenue est 79.5% en utilisant les réseaux de neurones artificiels . En 2018, Gattal et al., [17] ont proposé un système de prédiction du genre à partir de l'écriture manuscrite basé sur l'extraction de l'information de texture en utilisant différentes combinaison de oBIFs (oriented Basic Image Features) associé à un SVM. La meilleure performance obtenue est de 76% . Aujourd'hui, avec l'accentuation de l'utilisation des réseaux de neurones convolutifs (Convolutional Neural Networks : CNN) dans les problèmes de reconnaissance de formes, beaucoup de chercheurs se sont intéressés à utiliser cette nouvelle génération des machines d'apprentissage pour la prédiction du genre à partir de l'écriture manuscrite. La littérature rapporte quelques travaux utilisant les CNNs. En effet, Morera et al., [38] prétendent être les premiers à avoir utilisé les CNNs pour la prédiction du genre à partir de l'écriture manuscrite en utilisant les

bases de données de référence IAM et KHATT. Ils ont proposé une architecture de CNN comprenant 2 couches de convolution, 2 couches max-pooling et une couche dense. Les résultats rapportent des taux de bonne prédiction de 80.82% et 68.90% pour les bases IAM et KHATT, respectivement. Peu de temps après, Illouz et al., [26], proposent eux aussi une architecture CNN assez simple comportant 4 couches de convolution suivie chacune d’une couche max-pooling. Les tests ont été effectués sur une base de données privée en langue hébreu et anglaise ou les performances gravitent aux alentours de 75%. Par la suite en 2019, Kaljahi et al., [28] proposent d’appliquer un filtre de Gabor multidirectionnel à l’image d’entrée et d’ensuite la faire passer à un réseau de neurones convolutifs. Le tableau 1.1 récapitule ces travaux en termes de méthodes et de bases de données utilisées, mais aussi en termes de performances atteintes. En regardant ce tableau, nous remarquons que les taux de prédiction du genre à partir de l’écriture manuscrite varient entre 70% et 80% quelques soit la langue considérée. De plus, les systèmes basés CNN offrent des performances intéressantes, ce qui nous incite à explorer l’utilisation de l’apprentissage profond et plus particulièrement des CNNs pour développer notre propre système de prédiction du genre à partir de l’écriture manuscrite.

TABLE 1.1: Résumé des travaux de la littérature les plus récents

Référence	Base de données	Méthodes	Taux de Bonne Prédiction %
[7]	IAM (Anglais) KHATT (Arabe)	Combinaison floue de systèmes	82.73 82.22
[1]	QUWI (Arabe+Anglais) MSHD (Arabe+Fr)	Transformée en ondelettes discrètes + PFSA + SVM/RNA	79.30 79.90
[17]	QUWI ICDAR2013 Ar QUWI ICDAR2015 Ar QUWI ICFHR2016 Ar	OBIFs + SVM	77.07 79.50 75.00
[38]	IAM KHATT	CNN	80.72 68.90
[26]	Privée Hebreu Privée Anglais	CNN	75.00 77.63
[28]	IAM KHATT QUWI	Gabor + CNN	89.24 76.99 78.96

1.7 Conclusion

Dans ce chapitre, nous avons présenté l’écriture manuscrite en tant que caractéristique biométrique comportementale, ainsi que l’intérêt de prédire le genre à partir de l’écriture manuscrite. Nous avons également exposé l’architecture accompagnée d’un bref aperçu de l’état de l’art d’un système de prédiction du genre à partir de l’écriture manuscrite. Aujourd’hui, avec la vulgarisation de l’utilisation des réseaux de neurones convolutifs

dans tous les domaines et particulièrement, dans les derniers travaux sur la prédiction du genre à partir de l'écriture manuscrite, nous avons été incité dans ce travail, à explorer l'utilisation de ces techniques pour développer notre propre système, basé sur un pré-traitement de données et un réseau de neurones convolutifs associé à un filtre de Gabor. Le chapitre suivant sera consacré à la description de l'étape de pré-traitement qui consiste à générer l'image texture à partir d'échantillons d'écriture manuscrite.

Chapitre 2

Prétraitement des données : De l'image brute à l'image texture

2.1 Introduction

L'étape de pré-traitement regroupe l'ensemble des processus visant à améliorer l'information contenue dans une image afin d'assurer une meilleure extraction de caractéristiques et ainsi une meilleure prédiction/classification. L'objectif de ces processus est de supprimer les distorsions indésirables et/ou en améliorer certaines caractéristiques importantes de l'image afin de mettre en évidence la relation qui existe entre l'écriture manuscrite et l'information que nous voulons prédire, à savoir le genre. Dans notre travail, nous nous sommes basés sur une approche assez spéciale : l'image texture. En effet, d'après les travaux de Hanusiak et al., [20], l'image texture pourrait représenter au mieux les caractéristiques propres à chaque genre. Il est à noter qu'afin de générer l'image texture, plusieurs étapes sont nécessaires à savoir la binarisation, la segmentation, et le redimensionnement. Dans ce chapitre, nous expliquerons en détails chaque procédure afin d'arriver au résultat souhaité : l'image texture.

2.2 L'analyse de texture, quel intérêt ?

La texture est un motif répétitif de variations d'intensité locales dans une image de façon régulière. L'analyse de la texture est aujourd'hui, très répandue avec un large éventail d'avantages et d'applications liés au traitement d'images et à la vision par ordinateur. Effectivement, on retrouve l'analyse de texture dans le diagnostic médical assisté par ordinateur [18, 32], la classification des espèces forestières [49, 3], la classification en images aériennes/satellites [9, 27] et l'identification/vérification du scripteur à partir de l'écriture manuscrite [20, 5]. Un système de reconnaissance de forme est une suite d'étapes tout aussi importante les unes que les autres, parmi elles, l'étape d'extraction de caractéristiques. En effet, cette étape dans le contexte de la prédiction du genre à partir de l'écriture manuscrite n'est pas simple, étant donné la complexité de caractériser les styles d'écriture des deux genres. Par conséquent, l'étape de pré-traitement s'impose, pour préparer au mieux les données d'entrée, en renforçant la ressemblance intra-classe et en accentuant la différence inter-classe. Les problèmes rencontrés lors du pré-traitement sont liés aux difficultés de segmentation de l'écriture manuscrite en lignes, mots, ou caractères pour une caractérisation pertinente du style d'écriture. En effet, les problèmes de chevauchement, d'inclinaison et d'espace inter-mots et intra-mots, sont les problèmes de tout algorithme de segmentation. Pour contourner ces difficultés, nous avons adopté dans ce travail une stratégie de segmentation qui produira une image texture. Cette stratégie a été proposée par Hanusiak et al., [20] pour la vérification du scripteur à partir de l'écriture manuscrite. Cette texture est construite sur la base des propriétés inhérentes à l'écrivain. Autrement dit, quelle que soit la façon dont le texte a été rédigé, toutes les composantes connexes sont réorganisées dans un nouvel espace en conservant l'inclinaison d'origine mais en réduisant

les espaces entre les lignes de texte, les mots et les caractères. Ce processus crée une texture qui conserve les caractéristiques importantes du style d'écriture. La figure 2.1 illustre un exemple d'image texture obtenue à partir d'échantillons d'écriture manuscrite féminines et masculines.



FIGURE 2.1: Blocs textures

2.3 Segmentation des lignes en mots

La segmentation des lignes en mots consiste à découper les lignes afin de générer des mots unis ou un groupe de lettres qui sont proches les unes des autres. Ceci se base principalement sur la localisation et l'élimination des espaces blancs présents au sein d'une même phrase. Pour passer de la phrase aux mots séparés, plusieurs étapes sont nécessaires tel qu'indiqué dans la figure 2.2.

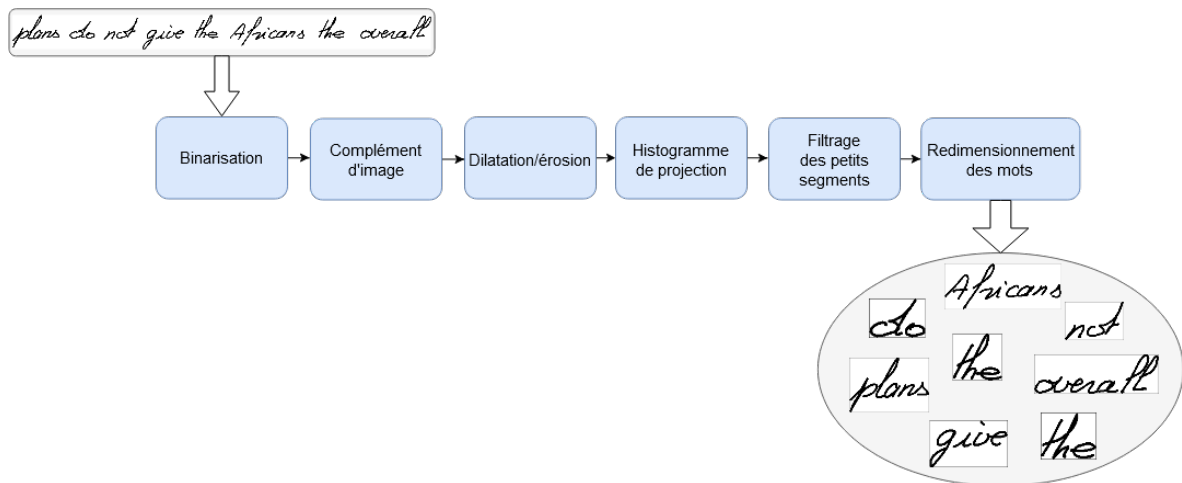


FIGURE 2.2: Processus de segmentation de la ligne en mots

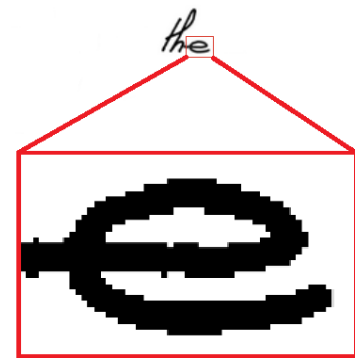
2.3.1 Binarisation des images

La binarisation vise à isoler l'écriture de l'arrière-plan du document. L'application de cette technique à une image en niveaux de gris donne une image binaire où chaque pixel peut avoir une valeur de 0 ou 1. Le processus de binarisation opère en trouvant une valeur seuil dans l'histogramme de l'image, qui sépare entre 2 entités, chacune représentant l'un des deux éléments ou un élément et son arrière-plan. Les algorithmes de binarisation sont divisés en deux catégories : binarisation globale et binarisation locale. La première utilise un seuil unique pour toute l'image, tandis que la dernière calcule des valeurs de seuil locales, pixel par pixel et région par région [42]. Dans les techniques de binarisation globale, il existe plusieurs algorithmes pour trouver la valeur du seuil optimal, mais aucun d'entre eux ne peut être qualifié de meilleur, car chaque méthode est optimale pour certaines applications. Dans notre travail, l'algorithme qui a prouvé son efficacité est l'algorithme d'Otsu [39]. Par une itération sur toutes les valeurs de seuil possibles, il mesure la dispersion des niveaux de pixels des deux côtés (élément et arrière-plan) de la valeur du seuil considéré. Ensuite, il recherche la valeur du seuil qui conduit à la somme minimale des pixels de l'élément et de l'arrière-plan [42].

Cette étape de binarisation permet d'avoir un faible espace mémoire, et donc, augmente la vitesse de traitement. La figure 2.3 illustre un exemple de binarisation d'une phrase à l'aide de la méthode Otsu.



(a) Avant binarisation



(b) Après binarisation

FIGURE 2.3: Binarisation d'un mot par la méthode Otsu

2.3.2 Complément d'image

Le complément d'une image en niveaux de gris ou en couleurs, consiste à soustraire à chaque valeur de pixel, la valeur de pixel maximale contenue dans l'image. On obtient ainsi une image de sortie, où les zones sombres deviennent plus claires et les zones claires deviennent plus sombres. Dans le cas d'une image binaire, les pixels blanc (255) deviennent noirs (0) et les pixels noirs deviennent blancs, ce qui donne une écriture en blanc sur un fond noir. La figure 2.4 montre une image binaire et son complément.



FIGURE 2.4: Complément d'image

2.3.3 Erosion et dilatation

La morphologie mathématique est un outil d'analyse et de modification des images mis au point en 1964 par l'équipe de Jean Serra à l'école des Mines de Fontaine bleu Paris [44]. A l'origine, la morphologie mathématique a été développée pour des images binaires, mais qui a depuis été étendue aux images en niveaux de gris. Le principe de base de la morphologie mathématique est de comparer l'image à analyser à un ensemble de géométries connues (voisinage) appelé élément structurant, qui représente un masque de convolution. L'élément structurant est déplacé de façon à ce que son origine (généralement au milieu de l'élément structurant) passe par toutes les positions de l'image, pour mettre en évidence certaines caractéristiques de cette dernière. En effet, à chaque position de l'élément structurant, on regarde s'il touche ou s'il est inclus dans l'ensemble initial. En fonction de la réponse, on construit un ensemble de sortie. Il existe plusieurs sortes

d'éléments structurants. Ils ne sont restreints ni en taille ni forme. La figure 2.5 montre une représentation des différents masques de convolution.

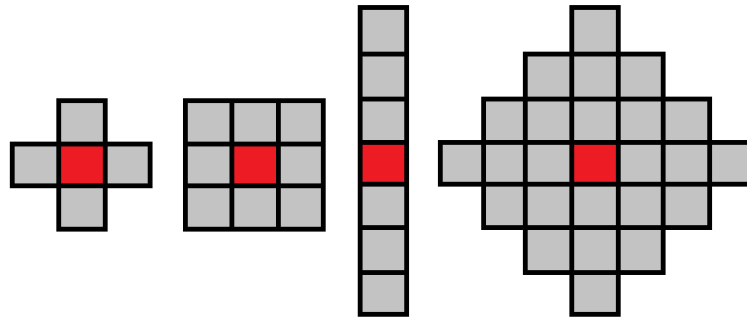


FIGURE 2.5: Éléments structurants ou masques de convolution

Les opérations de base de la morphologie mathématique sont deux opérations duales appelées respectivement érosion et dilatation. Ces deux opérations permettent d'améliorer la qualité informationnelle d'une image numérique. Leurs noms proviennent de l'effet produit sur une forme binaire par chacune de ces opérations.

a) Dilatation

La dilatation morphologique consiste à déplacer l'élément structurant sur chaque pixel de l'image, et à regarder s'il "touche" ou plus formellement "intersecte" la structure d'intérêt. C'est donc un OU logique. Soient X la structure d'intérêt qui appartient à une image I et B l'élément structurant. B est déplacé pour occuper successivement toutes les positions p dans l'image, où pour chaque position on vérifie si B a intersecté X (à au moins un pixel en commun). Mathématiquement, le dilaté de X par B est l'ensemble des pixels p pour lesquels B_p intersecte avec X . L'équation 2.1 résume le processus de dilatation.

$$\delta_B(X) = X \oplus B = \{p \in I | B_p \cap X \neq \emptyset\} \quad (2.1)$$

La dilatation possède plusieurs propriétés :

- La taille des formes (objets) augmente.
- Les trous et les concavités peuvent être bouchés (combler les intrusions).
- Les formes (objets) voisines peuvent se connecter.
- Les petits détails sont agrandis.
- Les traits interrompus sont réparés.

Dans notre travail, nous avons utilisé un élément structurant rectangulaire de taille 3 x 5 tel que représenté dans la figure 2.6 où l'élément en rouge représente le centre de l'élément structurant. L'élément structurant est choisi de cette forme car la nature

des mots est longitudinale. De ce fait, et afin d'unir davantage les lettres constituant un même mot, l'écriture manuscrite doit être étendue dans le sens longitudinal. Il est à noter que sept itérations de dilatation ont été appliquées sur l'image contenant l'écriture manuscrite. Le nombre sept est un choix empirique. En effet, ce nombre satisfait le compromis entre l'union des lettres du même mot et la séparation entre les différents mots. La figure 2.7 représente le résultat de dilatation après 7 itérations.

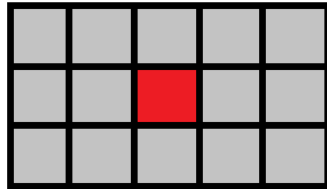


FIGURE 2.6: Éléments structurant rectangulaire de taille 3 x 5



(a) Avant Dilatation



(b) Après Dilatation

FIGURE 2.7: Dilatation morphologique

b) Érosion

L'érosion morphologique consiste à déplacer l'élément structurant repéré par son centre, sur chaque pixel de l'image et à regarder s'il est inclus complètement dans la structure d'intérêt. C'est donc un ET logique. Soient X la structure d'intérêt et B l'élément structurant. B est déplacé pour occuper successivement toutes les positions p dans l'image I , où pour chaque position p on vérifie si B est complètement inclus dans X . Mathématiquement, l'érodé de X par B est l'ensemble des pixels p pour lesquels B_p est inclus dans X . L'équation 2.2 décrit l'opération d'érosion.

$$\epsilon_B(X) = X \ominus B = \{p \in I | B_p \subset X\} \quad (2.2)$$

A son tour l'érosion possède plusieurs propriétés :

- La taille des formes (objets) réduite.
- Une forme (objet) avec des concavités ou des trous peut être séparée.
- Les petites formes (objets) et les détails disparaissent.
- Suppression des extrusions (ou aspérités).

Dans notre travail, l'image dilatée est érodée par le même élément structurant que pour la dilatation. Cette opération a pour but de réduire et de supprimer des extrusions générées par la dilatation. Trois itérations d'érosion ont été appliquées sur l'image dilatée, la figure 2.8 représente le résultat obtenu.

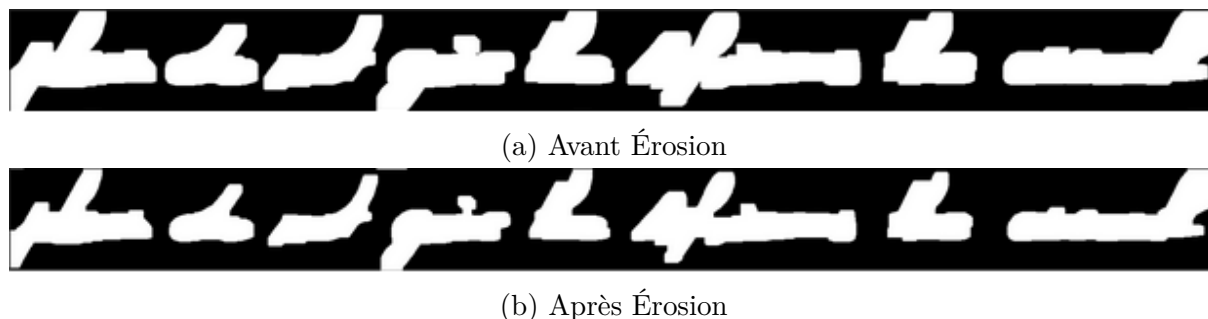


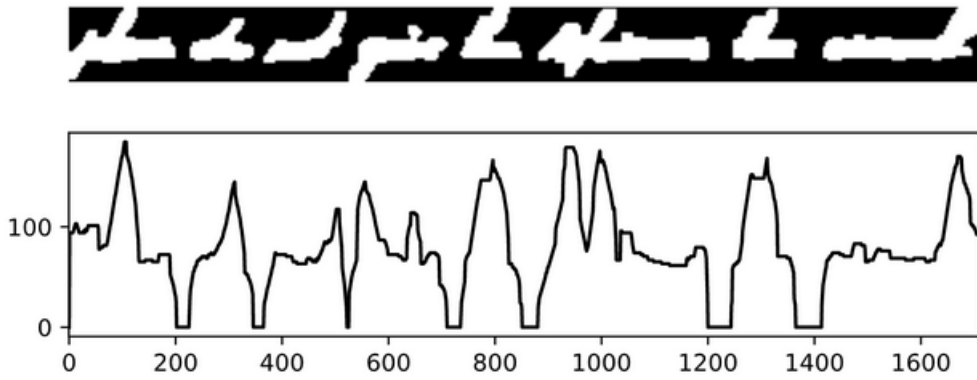
FIGURE 2.8: Érosion morphologique

2.3.4 Histogramme de projection horizontale et verticale

L'histogramme de projection d'une image dans une direction particulière fait référence à la somme cumulée des pixels dans cette direction.

Dans le contexte du traitement de texte, un histogramme de projection horizontale est nécessaire pour identifier et séparer les lignes d'un texte. Pour les images binaires, le passage par zéro de l'histogramme de projection horizontale indique un espace d'interligne et donc permet de segmenter le texte en lignes.

De même, un histogramme de projection verticale est utilisé pour effectuer la segmentation des lignes en mots, à mesure que les passages par zéro indique un espacement inter-mots. L'histogramme de projection verticale est calculé pour l'image dilatée/érodée comme indiqué dans la figure 2.9. Nous remarquons que l'histogramme contient plusieurs zéros qui correspondent à des espaces entre les mots. Nous pouvons donc segmenter l'image originale (avant la dilatation et l'érosion) aux positions de ces zéros et récupérer ainsi les mots.



(a) Histogramme de projection verticale de l'image dilatée/érodee



(b) Segmentation en mots

FIGURE 2.9: Segmentation d'une ligne en mots

2.3.5 Le filtrage des petits segments

Après que toutes les lignes soient segmentées en mots, il faut éliminer les segments aberrants. Ces derniers correspondent à des apostrophes, des ratures, des guillemets, etc. Ils se caractérisent par une petite dimension et ne contiennent pas d'information utile. La figure 2.10 présente quelques exemples de segments aberrants.



FIGURE 2.10: Exemples d'images aberrantes

2.3.6 Redimensionnement des mots

Afin de garder une homogénéité en terme de tailles des mots, une étude statistique a été effectuée sur la hauteur de chaque mot. Ceci ayant été réalisé à l'aide de l'histogramme : La fréquence de chaque hauteur est calculée. Ensuite, on fixe une hauteur seuil à partir de laquelle la fréquence des hauteurs est basse. Pour l'exemple de la figure 2.11, un seuil de 150 est choisi. On remarque qu'à partir de ce seuil, les hauteurs ont une fréquence très

basse. Les mots qui ont une hauteur supérieure à la valeur seuil fixée, sont redimensionnés à cette hauteur seuil. La figure 2.12 montre un exemple d'un mot redimensionné.

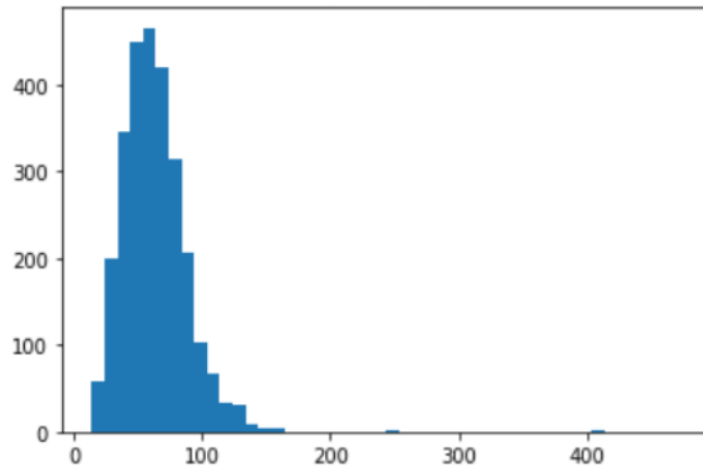


FIGURE 2.11: Histogramme des hauteurs des segments

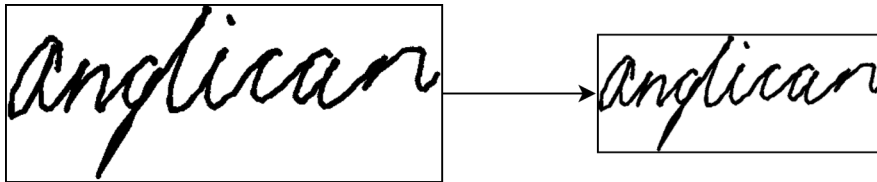


FIGURE 2.12: Exemple de redimensionnement d'un mot

2.4 Génération des images texturées

Après avoir segmenté les lignes et filtré toutes les images segmentées résultantes, l'étape de la texture intervient. Cette étape consiste l'étape clé de toute la procédure du pré-traitement. Nous expliquons dans ce qui suit les démarches suivies. Nous commençons par la définition du centre de masse, son expression mathématique et nous abordons juste après l'algorithme adopté.

2.4.1 Centre de masse

Avant d'entamer l'explication de l'algorithme adopté, nous introduisons une notion importante : Les moments. La définition des moments dans le traitement d'images est empruntée à la physique. En traitement d'image, un moment d'image est une certaine moyenne pondérée particulière des intensités des pixels de l'image. Les moments d'image sont utiles pour décrire les objets après la segmentation. Les propriétés simples de l'image qui se trouvent via les moments de l'image incluent la surface (où l'intensité totale), son centre de masse et des informations sur son orientation.

Le moment d'ordre $(i + j)$ est défini par l'équation suivante :

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y) \quad (2.3)$$

où x représente l'indice des lignes, y des colonnes, I l'image en question et $I(x, y)$ la valeur du pixel dont les coordonnées sont x et y .

Le centre de masse utilise le moment d'ordre 1. Les coordonnées du centre de masse sont données par l'équation 2.4.

$$\{\bar{x}, \bar{y}\} = \left\{ \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right\} \quad (2.4)$$

La figure 2.13 représente un exemple d'un centre de masse pour un mot.



FIGURE 2.13: Centre de masse d'un mot

2.4.2 Procédure de génération de l'image texture

Une fois les mots redimensionnés en hauteur et leurs centres de masse calculés, la génération de l'image texture suit le processus suivant :

1. Créer une image blanche (image plan) ou canevas de taille 1000x1500 pixels,
2. Placer les mots dans l'image canevas les uns après les autres, de gauche à droite. La coordonnée du centre de masse \bar{x} de chaque mot est positionnée avec le niveau x de la ligne dans l'image canevas, afin d'aligner les mots, tel que représenté dans la figure 2.14.

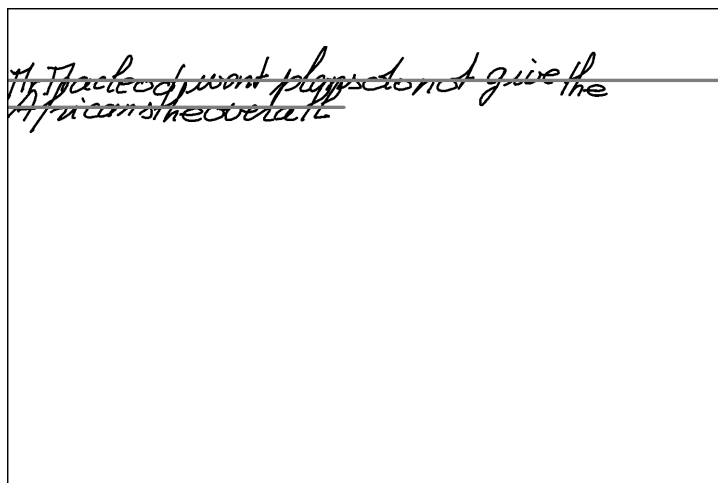


FIGURE 2.14: Remplissage de l'image canevas

- Si la taille des mots concaténés horizontalement, dépasse 1500 pixels, un saut de ligne est appliqué. L'interligne est calculé en additionnant la position de la ligne précédente à la hauteur moyenne des mots de cette dernière, multiplié par une constante, ce qui donne l'équation suivante de la position de la nouvelle ligne :

$$Position_nouvelle_x = position_précédente_x + hauteur_moyenne_des_mots * constante \quad (2.5)$$

- Une fois l'image canevas remplie en hauteur, une nouvelle image canevas est créée et remplie de la même manière.

La figure 2.15, illustre le résultat de la procédure de génération d'une image texture pour l'écriture féminine et masculine, respectivement.

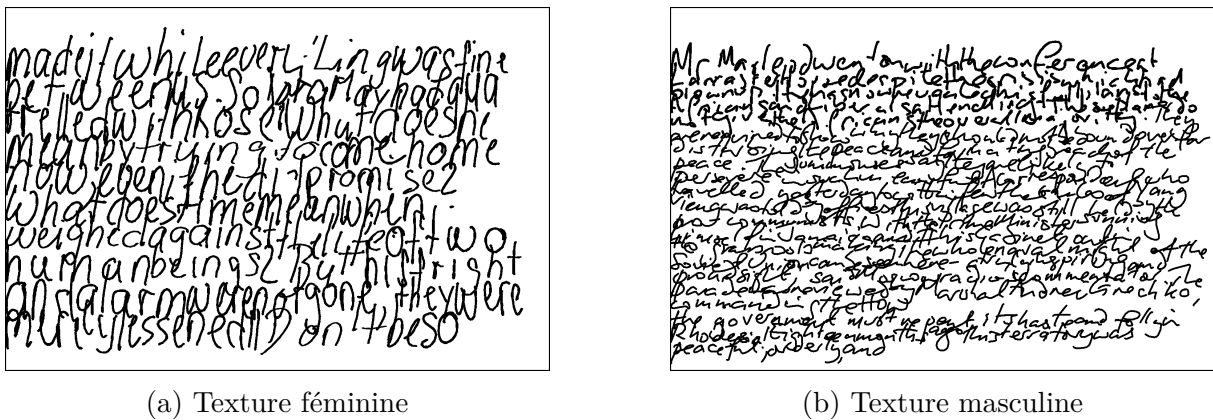


FIGURE 2.15: Images texture

2.5 Segmentation et filtrage des images textures

Les images textures générées de taille 1000x1500 pixels sont beaucoup trop grandes pour un réseaux de neurones convolutifs. Par conséquent, ces images textures par genre sont segmentées en blocs de taille inférieure (voir figure 2.16). Cette taille sera le sujet d'une étude ultérieure.

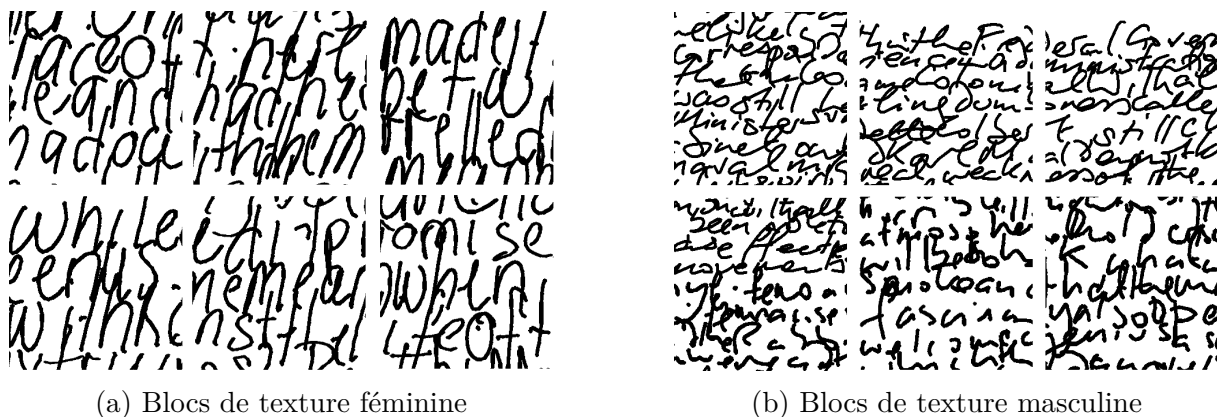


FIGURE 2.16: Blocs de texture

Après la segmentation des images textures en blocs, il y a parfois des blocs qui contiennent très peu d'information tel qu'indiqué dans la figure 2.17, il est donc nécessaire de les éliminer. Pour cela, nous avons adopté la notion de densité de pixels qui est le nombre de pixels noirs contenus dans l'image bloc sur le nombre total de pixels de l'image bloc. Cette densité de pixels nous renseigne sur la quantité d'information contenue dans le bloc. La figure 2.18 présente la densité de pixels pour 95 blocs de texture. Un seuil minimale de densité de pixels est fixé à 0.14. Tous bloc de texture ayant une densité de pixels inférieure à ce seuil est éliminé.

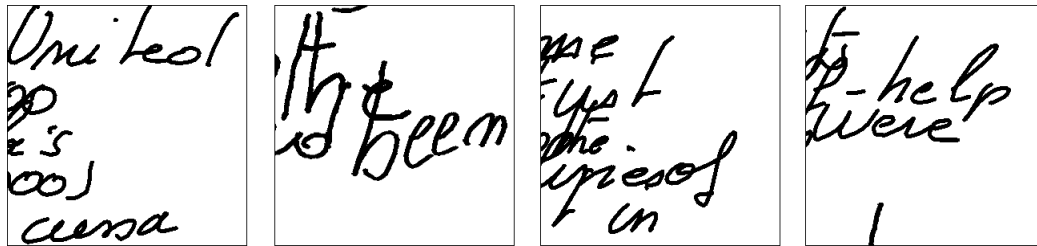


FIGURE 2.17: Exemples de blocs de texture inutilisés

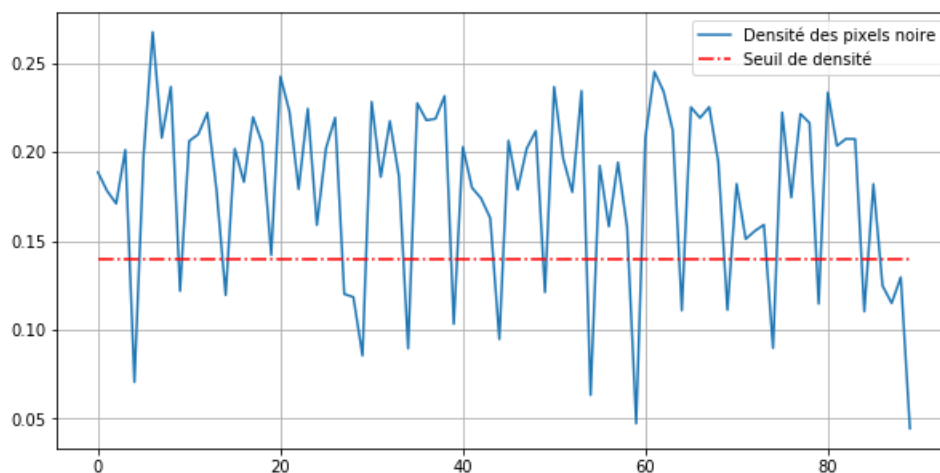


FIGURE 2.18: Densité de pixels des blocs de texture (seuil en rouge 0.14)

2.6 Conclusion

La prédiction du genre à partir de l'écriture manuscrite reste une tâche assez compliquée dans le sens où il n'est pas évident de caractériser et de différencier l'écriture masculine de l'écriture féminine et vice versa. Dans ce chapitre, nous avons présenté l'intérêt de la texture face à la prédiction du genre à partir de l'écriture manuscrite. Cette façon de générer les images textures a été introduite par Hanusiak et al., [20] et a été utilisée pour la vérification et l'identification du scripteur à partir de l'écriture manuscrite [20, 5]. Nous sommes donc passés d'échantillons d'écriture manuscrite par genre à une image texture. Les données sont donc prêtes à être exploitées par un réseau de neurones convolutifs

(CNN) qui fera l'objet du prochain chapitre, afin de différencier automatiquement entre l'écriture féminine et l'écriture masculine.

Chapitre 3

Réseaux de neurones convolutifs

3.1 Introduction

Au cours de la dernière décennie, l'intelligence artificielle a connu une croissance monumentale et s'approche de plus en plus de la capacité humaine et peut même des fois la dépasser. Ceci s'est particulièrement accentué avec l'introduction de l'apprentissage profond (deep learning). Faisant partie d'une famille plus large de méthodes d'apprentissage automatique, l'apprentissage profond est basé sur des réseaux de neurones artificiels. En effet, les progrès de la vision par ordinateur avec l'apprentissage profond ont principalement tourné autour des réseaux de neurones convolutifs (Convolutional Neural Networks : CNNs) . De nos jours, les CNNs sont considérés comme le meilleur algorithme d'intelligence artificielle pour la classification d'images. Spécifiquement, les CNNs ont été utilisés pour la prédiction du genre à partir de l'écriture manuscrite dans plusieurs travaux [38, 28]. L'utilisation des CNNs pour l'analyse de l'écriture manuscrite reste d'actualité dans plusieurs applications telles que l'identification/vérification du scripteur, la reconnaissance de mots et la vérification de la signature. En effet, les CNNs, contrairement aux autres algorithmes d'apprentissage automatique, extraient les caractéristiques présentes dans une image de façon automatique : c'est donc le point fort de ce type d'intelligence artificielle.

Dans ce chapitre, nous proposons d'utiliser les CNNs pour la prédiction du genre à partir de l'écriture manuscrite. Ce chapitre donne d'abord un bref aperçu sur les réseaux de neurones artificiels, avant de passer à une description plus détaillée des CNNs.

3.2 Historique

Les réseaux de neurones convolutifs sont inspirés du système visuel humain. En 1959, David Hubel et Torsten Wiesel [25] ont décrit des «cellules simples» et des «cellules complexes» dans le cortex visuel humain. Ils ont proposé que les deux types de cellules soient utilisés dans la reconnaissance des formes. Une «cellule simple» répond à des bords et des barres d'orientations simples, tandis qu'une «cellule complexe» répond également aux bords et aux barres mais avec des orientations particulières, cette propriété des cellules complexes est appelée «invariance spatiale». Hubel et Wiesel ont proposé en 1962 [24] que les cellules complexes arrivent à l'invariance spatiale en «sommant» la sortie de plusieurs cellules simples qui préfèrent toutes la même orientation (par exemple des barres horizontales) mais des champs récepteurs différents (par exemple en bas, au milieu ou en haut d'une image). En collectant des informations à partir de cellules simples, les cellules complexes peuvent donc répondre aux barres horizontales qui se produisent n'importe où. Ce concept - que les détecteurs simples peuvent être «sommés» pour créer des détecteurs plus complexes - se retrouve dans tout le système visuel humain et constitue également la base fondamentale des modèles de réseaux de neurones à convolution. Dans les années 1980, Dr Kuniyuki Fukushima s'est inspiré des travaux de Hubel et Wiesel

sur les cellules simples et complexes et a proposé le modèle «neocognitron» [16]. Le modèle néocognitron comprend des composants appelés «cellules S» et «cellules C». Ce ne sont pas des cellules biologiques, mais plutôt des opérations mathématiques. Les «cellules S» se trouvent dans la première couche du modèle et sont connectées aux «cellules C» qui se trouvent dans la deuxième couche du modèle. L'idée générale est de capturer le concept «simple à complexe» et de le transformer en un modèle informatique pour la reconnaissance visuelle des formes. Les premiers travaux sur les CNNs ont eu lieu dans les années 1990, inspirés du néocognitron. Yann LeCun et al., [31] a démontré qu'un modèle CNN qui agrège des fonctionnalités plus simples en fonctionnalités progressivement plus complexes peut être utilisé avec succès pour la reconnaissance de caractères manuscrits. Tout au long des années 1990 et au début des années 2000, les chercheurs ont poursuivi leurs travaux sur le modèle CNN. Autour de 2012, les CNNs ont connu une énorme augmentation de popularité (qui continue aujourd'hui) après qu'un CNN appelé AlexNet [30] ait dépassé les méthodes classiques dans la compétition de classification ImageNet. Au cours des dernières années, les CNNs ont réalisé d'excellentes performances décrivant des images naturelles (ImageNet, CIFAR-10, CIFAR-100 et VisualGenome), effectuant une reconnaissance faciale et analysant des images médicales (des radiographies pulmonaires, des photos de lésions cutanées, etc.). Dans ce qui suit, nous présentons des définitions fondamentales afin de comprendre le fonctionnement des réseaux de neurones convolutifs.

3.2.1 Notion de neurone formel

Un neurone formel est l'unité élémentaire de calcul d'un réseau de neurones. Son principe général est de retourner une information en sortie à partir de plusieurs informations en entrée. Des poids sont ensuite associés aux différentes entrées. Le neurone traitera alors la moyenne des entrées pondérées par les différents poids attribués, notons cette moyenne \bar{X} . Le rôle du neurone est donc de fournir une sortie y entre 0 et 1 à partir de \bar{X} à l'aide d'une fonction g : c'est la fonction d'activation. La sortie y aura comme formule :

$$y = g(\bar{X}) = g\left(\sum_{i=1}^n W_i X_i + W_0\right) \quad (3.1)$$

avec X_i les entrées du neurones et W_i les différentes poids attribués. La fonction g peut être linéaire ou non, mais doit fournir une sortie entre 0 et 1. La figure 3.1 illustre l'opération que fait un neurone pour deux entrées X_1 et X_2 pondérées par les poids W_1 et W_2 respectivement. Un biais W_0 est ajouté.

3.2.2 Perceptron mono-couche

Un perceptron mono-couche (Single-layer perceptron : SLP) est un type de réseau de neurones ou les neurones constituant ce réseau ne sont pas à proprement parlé en réseau

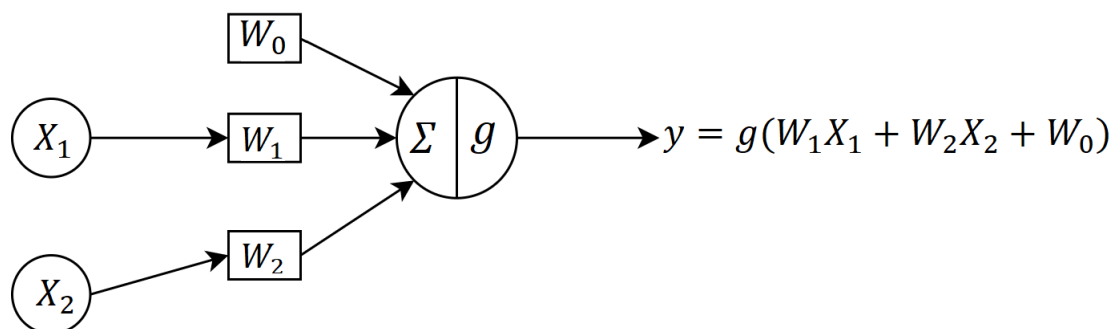


FIGURE 3.1: Neurone formel

mais ils sont considérés comme un ensemble. Ce type de réseau est capable d'apprendre à classifier des éléments dont la différence intra-classe n'est pas très compliquée. Il est donc utilisé pour des applications qui sont plus au moins simples. Le perceptron mono-couche est composé de k neurones, chacun étant connecté à toutes les variables d'entrée. Globalement, ce réseau dispose donc de n entrées et de p sorties, il peut être représenté comme sur la figure 3.2.

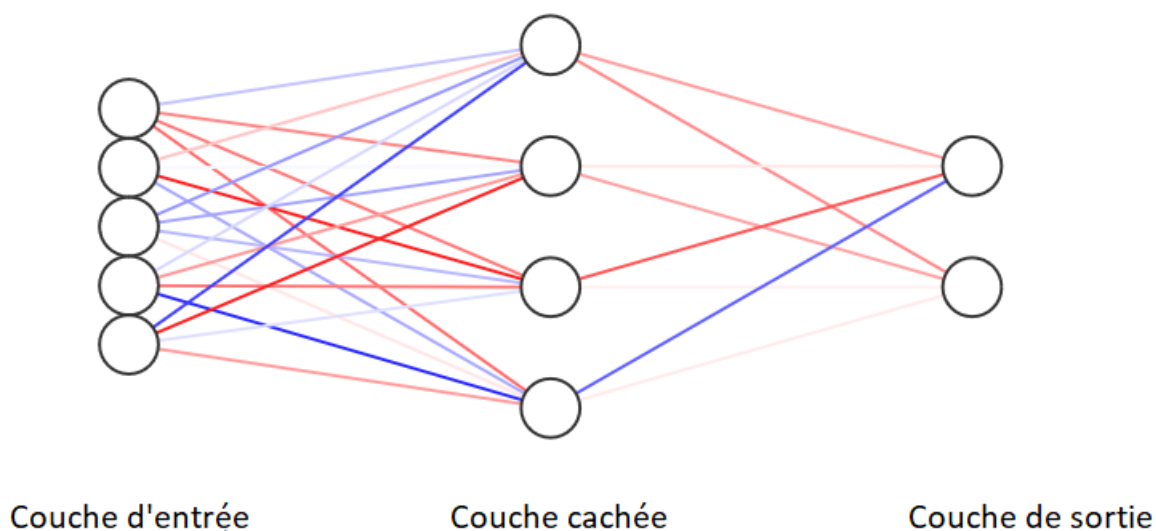


FIGURE 3.2: Perceptron mono-couche

3.2.3 Perceptron multi-couche

Il est possible de généraliser le perceptron en empilant plusieurs perceptrons mono-couches. De cette façon, les sorties d'une couche sont les entrées de la suivante. Les neurones d'une couches sont tous liés avec les neurones de la couches précédente et la couche suivante, mais aucune liaison n'existe entre les neurones de la même couche. Ce réseau de neurone est, bien entendu, plus compliqué mais s'utilise de manière similaire à un perceptron mono-couche. L'intérêt principal est d'être capable d'approcher des comportements moins linéaires et d'obtenir ainsi des erreurs plus faibles sur les données d'entraînement

(forte adéquation aux données) au prix d'une complexité plus grande. L'architecture de la figure 3.3 représente un perceptron multi-couche.

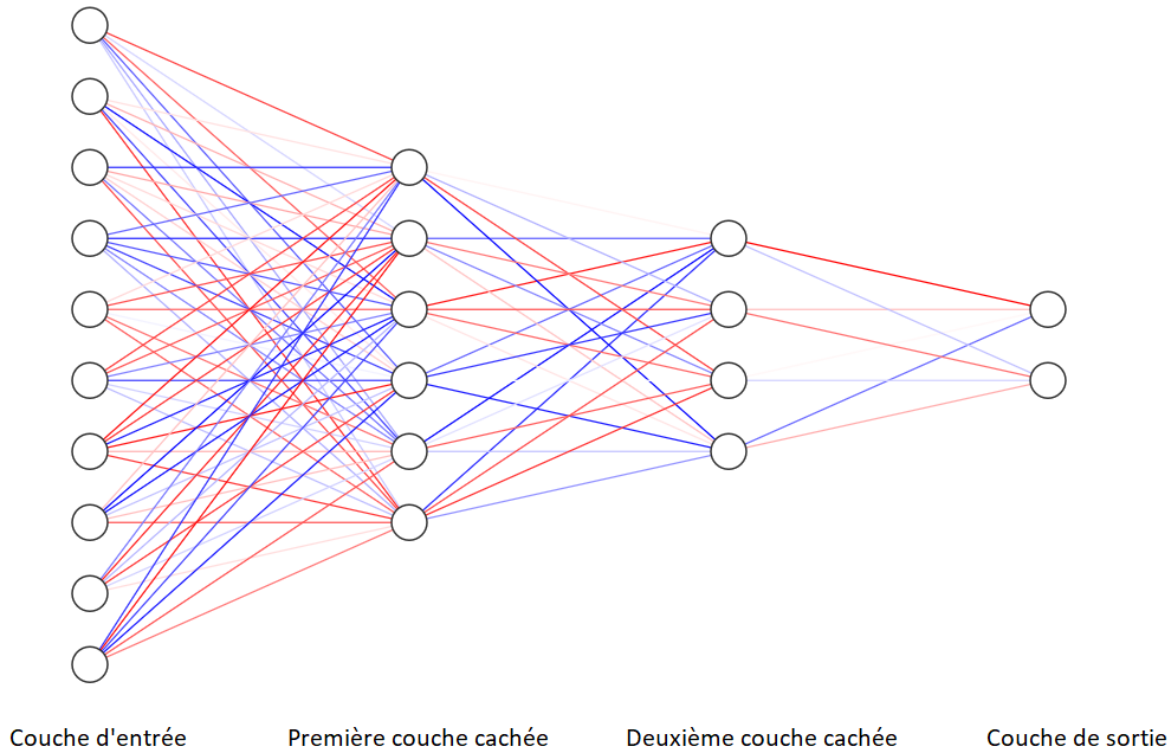


FIGURE 3.3: Perceptron multi-couches

Afin de déterminer les poids synaptiques W_i , le MLP applique l'algorithme de rétropropagation lors de l'apprentissage, qui consiste à ajuster les poids du réseau en minimisant l'erreur quadratique entre la sortie réelle et la sortie prédite (\hat{s} , s), respectivement. L'erreur quadratique est décrite par l'équation 3.2.

$$e_{\hat{s},s} = \frac{1}{2} \sum_{i=1}^m (\hat{s}_i - s_i)^2 \quad (3.2)$$

Les poids synaptiques sont ensuite mis à jour de la façon suivante :

$$W(l+1) = W(l) + \Delta W(l+1) \quad (3.3)$$

où $\Delta W(l+1)$ est le changement de poids synaptique à l'itération $l+1$, calculé de la manière suivante :

$$\Delta W(l+1) = -\tau \frac{\partial e}{\partial W} + \rho \Delta W(l) \quad (3.4)$$

Avec $\frac{\partial e}{\partial W}$ gradient de l'erreur par rapport au poids W tel que :

$$\frac{\partial e}{\partial W} = \sum_{i=1}^m (\hat{s}_i - s_i) \frac{\partial (\hat{s}_i - s_i)}{\partial W} \quad (3.5)$$

Les deux paramètres τ et ρ , sont respectivement le pas d'apprentissage et le momentum. Le pas d'apprentissage détermine à chaque itération la pas qui permet de se déplacer vers le minimum de la fonction coût (Loss Function). Le momentum est un paramètre de régularisation utilisé pour diminuer les fluctuations des changements de poids sur 2 itérations successives afin d'assurer une plus grande stabilité du processus de convergence.

3.3 Réseaux de neurones convolutifs (CNNs)

Les réseaux de neurones convolutifs sont très similaires aux réseaux de neurones artificiels : ils sont constitués de neurones qui ont des poids et des biais apprenants. Chaque neurone reçoit des entrées, effectue un produit scalaire et le suit éventuellement avec une fonction non linéaire.

Les réseaux de neurones convolutifs sont composés de deux blocs : un bloc d'extraction de caractéristiques et un bloc de classification. Le premier bloc reçoit en entrée une image, et en extrait les caractéristiques de manière hiérarchique à travers plusieurs couches de convolution et de pooling successivement. Le deuxième bloc reçoit le vecteur hiérarchique produit par le premier bloc, et le fait passer par un réseau de neurones complètement connectés (FC pour Fully Connected) qui fournira à son rôle les prédictions finales. La figure 3.4 illustre l'arrangement des couches dans un réseau de neurones convolutifs.

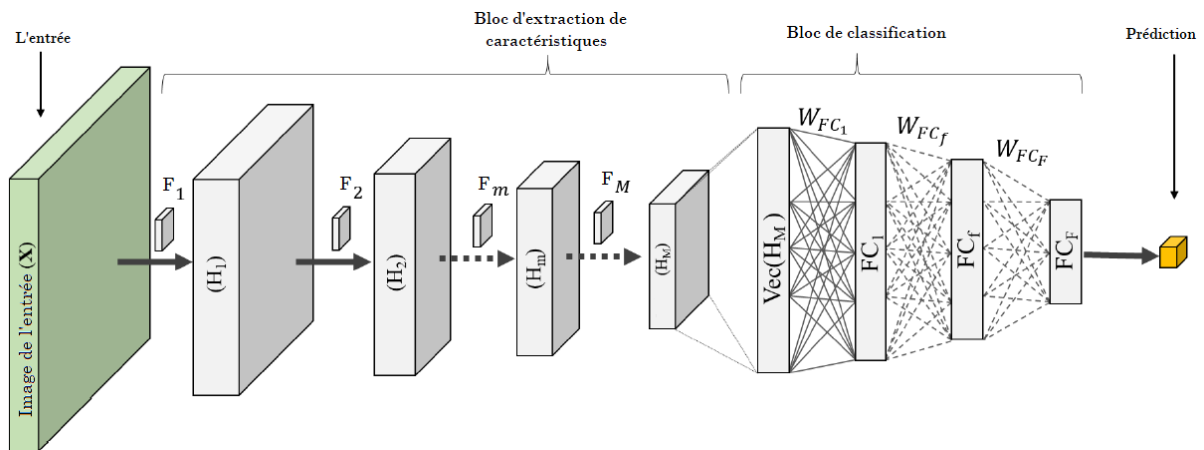


FIGURE 3.4: Blocs des couches composant un CNN

3.3.1 Bloc d'extraction de caractéristiques

Ce premier bloc fait la particularité de ce réseau puisqu'il fonctionne comme un extracteur automatique de caractéristiques. La première couche de ce bloc filtre l'image avec plusieurs noyaux de convolution et renvoie des caractéristiques qui passent par une fonction d'activation, générant ainsi une carte de caractéristiques. Les cartes de caractéristiques obtenues sont filtrées en utilisant de nouveaux noyaux de convolution plusieurs

fois, générant à chaque fois une nouvelle carte de caractéristiques. Le processus se répète autant de fois qu'il y a de couches. Enfin, les valeurs de la dernière carte caractéristiques sont concaténées et aplaties en un vecteur colonne : c'est le vecteur caractéristique. Ce vecteur définit la sortie du premier bloc et l'entrée du seconde.

Il existe plusieurs types de couches dans cette partie : les couches convolutives, les couches d'activations, les couches pooling et la couche Flatten. Nous expliquerons dans ce qui suit toutes les couches citées.

■ Couches convolutives

Une convolution est le processus consistant à faire glisser un filtre (noyau) sur l'image d'entrée tout en effectuant la somme d'une multiplication élément par élément entre les valeurs du filtre et la section correspondante de l'image d'entrée. Le réseau reçoit en entrée une image, la fait convolué avec le premier filtre F_1 et y ajoute un bias b_1 . Le même processus est répété plusieurs fois entre les couches : la sortie d'une couche i représente l'entrée de la couche $i + 1$.

Le filtre de convolution a le même nombre de couches que les canaux du volume d'entrée et le volume de sortie a la même profondeur que le nombre de filtres. Concrètement, la couche de convolution reçoit en entrée une image de taille (W_1, H_1, D_1) . Sur cette entrée, K convolutions sont appliquées, K étant le nombre de filtres constituant la couche de convolution. Chaque convolution est appliquée avec un filtre de taille (w_1, h_1, D_1) , généralement, $w_1 = h_1 = k$ où k est appelé taille du filtre. En plus du nombre de filtres, la couche de convolution présente deux autres hyperparamètres :

- Le stride S : il indique le pas de déplacement du noyau après chaque convolution et ;
- Le padding P : Il indique combien de 0 sont ajoutés autour de l'image. Le padding simple garantit que la sortie de la convolution a la même dimension que l'image d'entrée.

Enfin, chaque convolution avec un filtre produit une carte caractéristique en sortie, nous aurons donc, en sortie, un ensemble de carte caractéristique de taille (W_2, H_2, D_2) où : $W_2 = \lfloor \frac{W_1 - k + 2P}{S} \rfloor + 1$; $H_2 = \lfloor \frac{H_1 - k + 2P}{S} \rfloor + 1$ et $D_2 = K$. La figure 3.5 résume le processus de la couche de convolution.

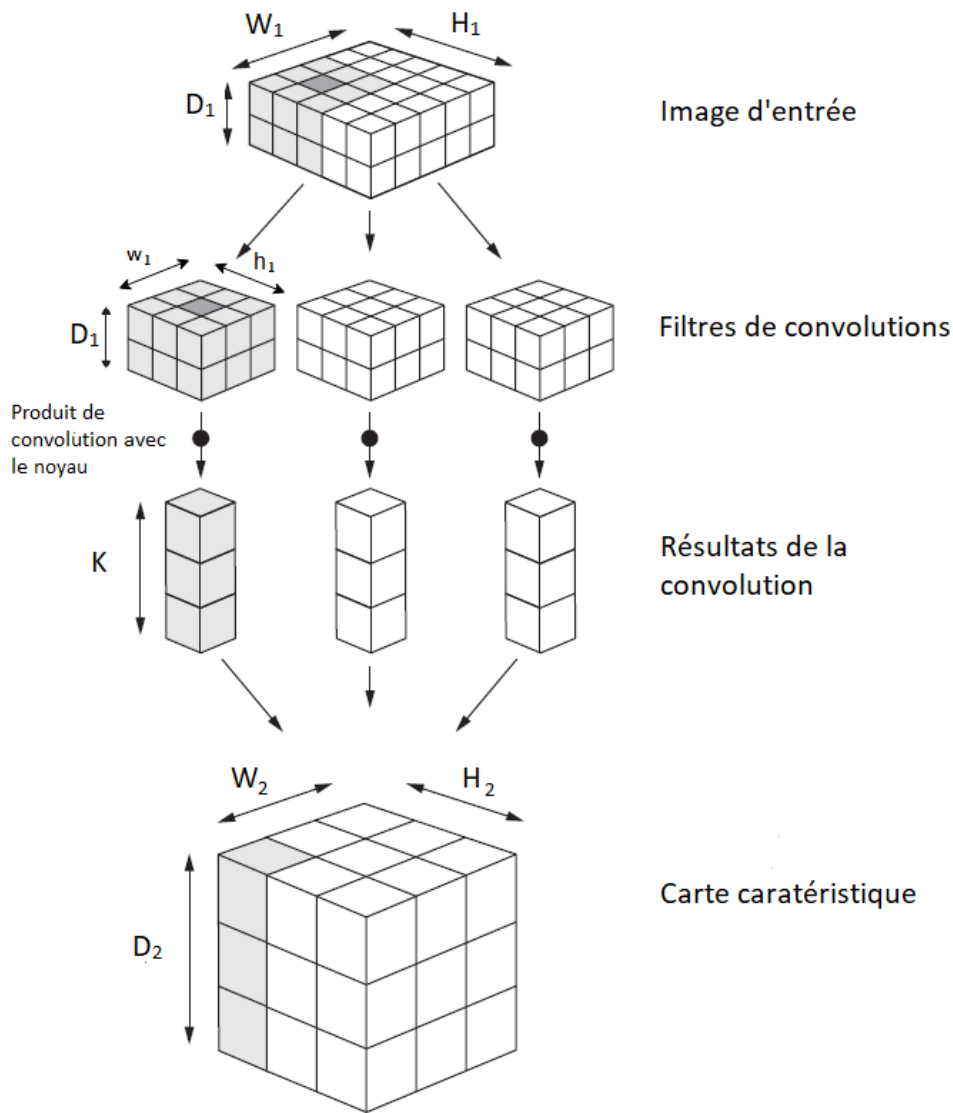


FIGURE 3.5: Processus de la couche de convolution

Les couches de convolution apprennent les motifs locaux dans une image, cette caractéristique clé donne aux réseaux de neurones convolutifs deux propriétés intéressantes :

- Les motifs qu'ils apprennent sont invariants par rapport à la translation ;
- Ils peuvent apprendre les hiérarchies spatiales des objets. Une première couche de convolution va apprendre de petits motifs locaux tels que les bords, alors qu'une deuxième couche de convolution va apprendre des motifs plus grands constitués des caractéristiques des premières couches, etc. L'exemple de la figure 3.6 indique ce qu'un CNN apprend au fil des couches. Ceci lui permet d'apprendre efficacement des concepts visuels de plus en plus complexes et abstraits.

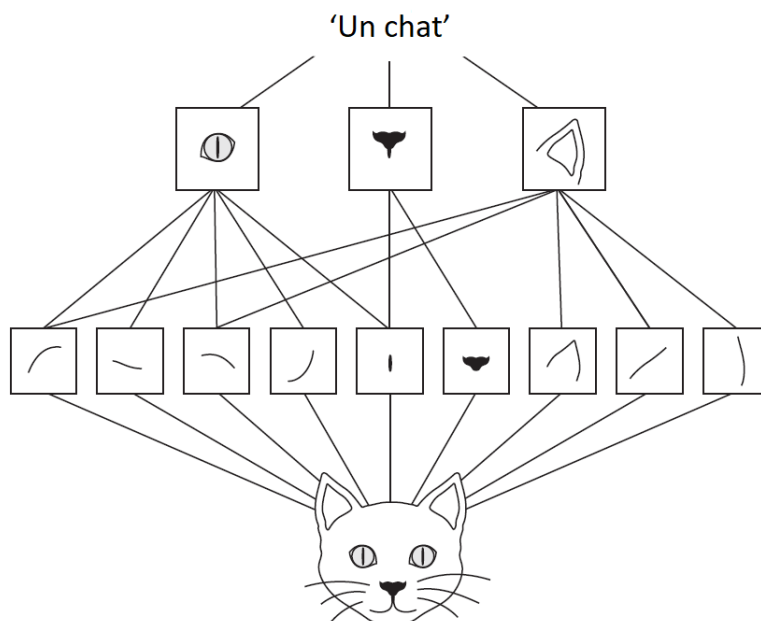


FIGURE 3.6: Les couches convolutives forment une hiérarchie spatiale des objets visuels : les bords hyperlocaux se combinent en objets locaux tels que les yeux ou les oreilles, qui se combinent en concepts de haut niveau tels que «chat»

■ Couches d'activation

Jusqu'à présent, seules des transformations linéaires ont été appliquées. Afin de capturer des caractéristiques plus complexes, des transformations non linéaires sont nécessaires. Ces transformations sont appelées fonctions d'activation $g(\cdot)$. Ces fonctions sont appliquées élément par élément sur la sortie de chaque convolution, comme l'indique l'équation 3.6

$$a = g(\mathbf{F}_m \mathbf{H}_{m-1} + \mathbf{b}_m) \quad (3.6)$$

où $\mathbf{F}_m \mathbf{H}_{m-1} + \mathbf{b}_m$ représente la sortie de la couche de convolution. Il existe plusieurs couches d'activations, nous citons entre autres :

- La couche de correction ReLU (Rectified Linear Units) : Elle désigne la fonction non-linéaire définie par : $ReLU(x) = \max(0, x)$. La couche de correction ReLU remplace donc toutes les valeurs négatives reçues en entrées par des zéros ;
- La couche de correction Leaky ReLU : C'est une version améliorée de ReLU, elle vient résoudre le problème du *Vanishing Gradient problem*¹. Cette couche non-linéaire est définie par : $LeakyReLU(x) = \max(\epsilon x, x)$ avec $\epsilon \ll 1$;
- La couche *Sigmoïde* : Cette fonction, utilisée à la fin du réseau, réduit les valeurs en entrée en une probabilité entre 0 et 1. Elle permet donc de donner

1. Un effet qui se produit dans les réseaux de neurones profonds : Les gradients de la sortie du réseau par rapport aux paramètres des premières couches deviennent extrêmement faibles

une probabilité d'appartenance à une classe précise. Sigmoides s'utilise dans les problèmes de classification binaire. Elle est décrite par l'équation suivante : $Sigmoid(x) = \frac{1}{1+e^{-x}}$.

- La couche *Softmax* : Cette fonction est utilisée dans les problèmes de classifications multiclassés. Softmax attribue des probabilités décimales à chaque classe d'un problème à plusieurs classes. La somme de ces probabilités décimales doit être égale à 1. Elle est décrite par l'équation suivante : $G(e_j) = \frac{\exp(e_j)}{\sum_i \exp(e_i)}$ où e_j est l'élément considéré j du vecteur d'entrée.

La Figure 3.7 illustre la représentation graphique des ces fonctions.

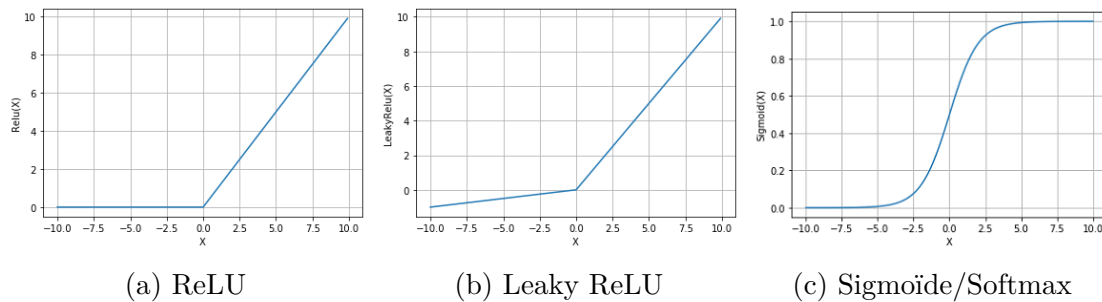


FIGURE 3.7: Les différentes fonctions d'activations

■ Couche Pooling

La couche Pooling assure une opération de sous-échantillonnage de l'image en fonction des dimensions spatiales, largeur et hauteur uniquement. L'image d'entrée est divisée en une série de carrés qui ne se chevauchent pas. Ce type de couche est fréquemment inséré entre deux couches convolutives successives d'une architecture CNN, pour réduire le nombre de paramètres et la puissance de calcul requise, tout en préservant les caractéristiques les plus importantes. En pratique, l'image est divisée en petites cellules carrées adjacentes, espacées les unes des autres d'un pas, afin de ne pas perdre trop d'informations. Le même nombre d'images de sortie que les images d'entrée est obtenu, mais chaque image a un nombre de pixels inférieur [43]. Il existe deux principaux types de pooling : le max-pooling et le average-pooling. Le max-pooling renvoie la valeur maximale de la partie de l'image recouverte par le noyau. D'autre part, le average-pooling renvoie la moyenne de toutes les valeurs de la partie de l'image recouverte par le noyau. Dans la pratique, le max-pooling est le plus utilisé et fonctionne bien mieux que le average-pooling. La figure 3.8 représente un exemple d'opérations de max-pooling et d'average-pooling.

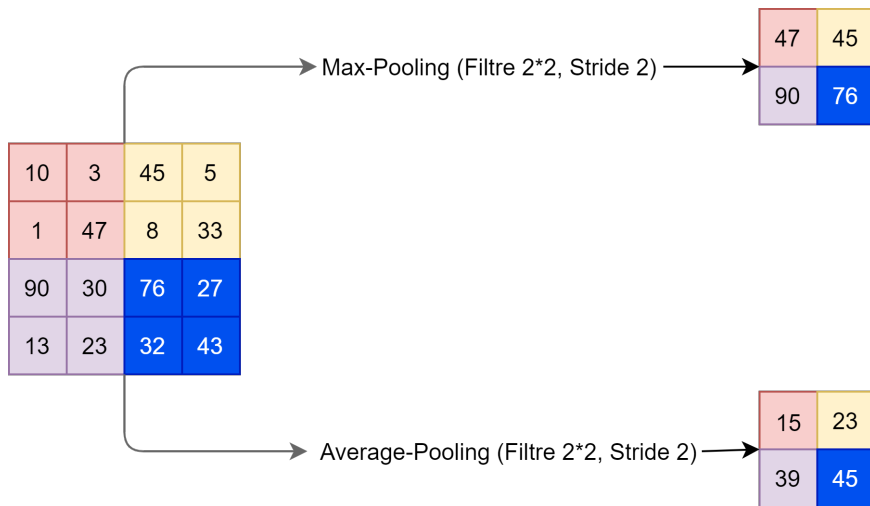


FIGURE 3.8: Fonctions de pooling

■ Le flattening (aplatissement)

La dernière couche du bloc d'extraction de caractéristiques est la couche d'aplatissement ou Flattening. Elle consiste à concaténer les lignes des matrices issues de la dernière couche dans un vecteur unidimensionnel que l'on peut connecter aux couches du bloc qui suit, comme indiqué sur la figure 3.9.

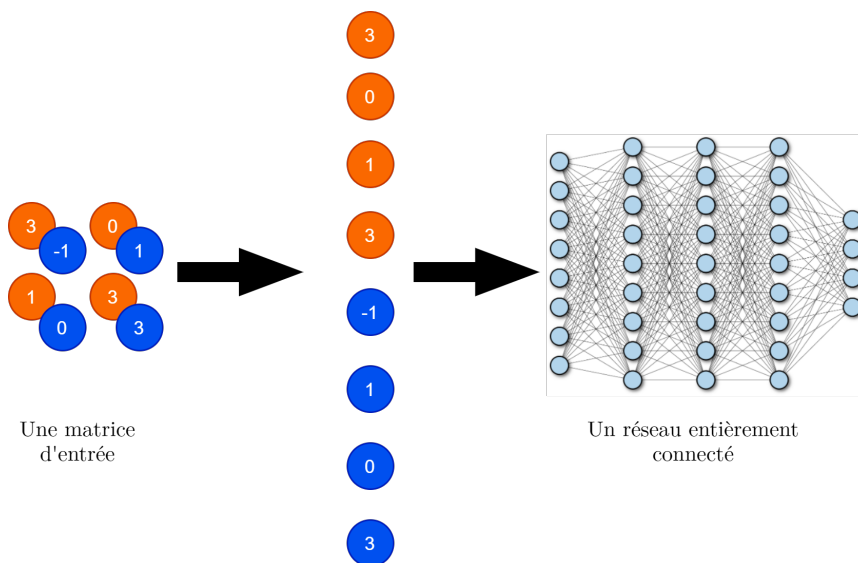


FIGURE 3.9: Couche *Flatten*

3.3.2 Bloc de classification

La classification est effectuée via des couches entièrement connectées (: Fully Connected; FC), similaires au perceptron multicouches. L'objectif des couches FC est d'utiliser le vecteur d'aplatissement d'entrée pour apprendre les meilleurs paramètres afin d'attribuer une classe à une image. En pratique, plusieurs couches sont placées les unes après les autres pour renforcer l'apprentissage des caractéristiques et améliorer la prise de dé-

cision. Un "dropout" peut être partiellement appliqué sur ces couches afin de réduire le sur-apprentissage et réduire le temps d'apprentissage en réduisant le nombre de paramètres à définir. La dernière couche FC renvoie la sortie du CNN, qui correspond à un vecteur de taille égale au nombre de classes. Les neurones dans cette couche utilisent la fonction d'activation Softmax pour fournir les probabilités avec lesquelles une image d'entrée appartient aux différentes classes [41].

3.3.3 L'entraînement

Le processus de l'entraînement consiste en la recherche des paramètres appropriés $\Theta = \{W, b\}$ où W représente les poids synaptiques et b le biais, pour trouver la plus proche fonction qui fait la correspondance entre la sortie désirée et la sortie prédite. A partir de là, il est nécessaire d'utiliser une fonction afin d'évaluer l'erreur commise : c'est la fonction coût $\mathcal{L}(\Theta)$.

Dans le cas d'une classification multiclassées, l'entropie croisée (EC) entre la probabilité prédite \hat{p} de chaque classe et la vraie probabilité p est utilisée, son expression est décrite par l'équation suivante :

$$EC(\Theta) = - \sum_{k=1}^K (\log \hat{p}(k)) \cdot p(k) \quad (3.7)$$

Pour un problème de classification binaire, l'entropie croisée binaire (ECB) est utilisé, son expression est décrite telle que :

$$ECB(\Theta) = -p \log \hat{p} - (1 - p) \log(1 - \hat{p}) \quad (3.8)$$

- Optimisation

La fonction coût $\mathcal{L}(\Theta)$ définie en dessus doit être minimisée. Pendant l'entraînement, cette fonction est minimisée par rapport aux paramètres $\Theta = \{W, b\}$. L'algorithme de la descente du gradient (gradient descent) est la technique la plus appliquée pour effectuer une optimisation sur des réseaux de neurones profonds. Cette technique permet de trouver le minimum de n'importe quelle fonction convexe en convergeant progressivement vers celui-ci. Ceci est possible en utilisant la rétro-propagation. La rétro-propagation consiste à rétro-propager l'erreur commise par un neurone à ses synapses² et aux neurones qui y sont reliés, et ce en utilisant la rétro-propagation du gradient de l'erreur, qui consiste à corriger les erreurs selon l'importance des poids qui ont justement participé à la réalisation de ces erreurs. Les poids synaptiques qui contribuent à engendrer une erreur importante

2. Points de connexion avec les autres neurones

se verront modifiés de manière plus significative que les poids qui ont engendré une erreur moins importante. Ensuite, les paramètres seront mis à jour de la manière suivante : Θ à chaque itération comme suit :

$$\Theta \leftarrow \Theta - \alpha \cdot \nabla_{\Theta} \mathcal{L}(\Theta) \quad (3.9)$$

avec α de le pas d'apprentissage, ∇_{Θ} le gradient de la fonction coût par rapport aux paramètres Θ , et $\mathcal{L}(\Theta)$ est la fonction coût. Le processus d'optimisation se répète autant de fois qu'il existe d'itérations.

3.4 Régularisation

Les réseaux de neurones profonds ont généralement des dizaines de milliers de paramètres, parfois même des millions. Avec autant de paramètres, le réseau dispose d'une liberté incroyable et peut s'adapter à une grande variété d'ensembles de données complexes. Par ailleurs, cette grande flexibilité signifie qu'elle est sujette au sur-apprentissage de l'ensemble d'entraînement. Dans cette section, nous présenterons certaines techniques de régularisation les plus populaires pour les réseaux de neurones afin d'éviter le sur-apprentissage.

■ Arrêt précoce (Early stopping)

En apprentissage automatique, l'arrêt précoce ou le early stopping est une forme de régularisation utilisée pour éviter le sur-apprentissage lors de l'entraînement des réseaux de neurones. Cette technique repose sur la notion de nombre d'itérations appelées "epochs". Cependant, fixer ce nombre peut avoir des conséquences sur les performances du modèle : Si le nombre n'est pas suffisamment grand, le modèle apprendra mal la relation existante entre l'entrée et la sortie tandis que si le nombre est assez grand, le modèle risquera de tomber dans le sur-apprentissage. Le early stopping vient donc palier ce problème. Il est appelé une fois qu'une itération se termine, il vérifie si la métrique pour laquelle le modèle est configurée, le taux de bonne prédiction par exemple, s'est améliorée par rapport à la meilleure valeur trouvée jusqu'à présent. Si elle ne s'est pas améliorée, il augmente le nombre de «temps non améliorés depuis la meilleure valeur» de un. Si elle s'est effectivement améliorée, il réinitialise ce décompte. Ainsi, le early stopping permettra donc d'arrêter l'entraînement d'un modèle après un certain nombre d'itérations fixé au préalable, appelé *patience*, évitant ainsi le sur-apprentissage.

■ Dropout

La technique de régularisation la plus populaire pour les réseaux de neurones convolutifs est le *Dropout* [48]. A chaque itération durant la phase d'entraînement, chaque neurone a une probabilité p d'être temporairement désactivé ("dropped out" en anglais), ce qui signifie qu'il sera entièrement ignoré lors de cette itération d'entraînement, mais il peut être actif à l'itération suivante. L'hyper-paramètre p est appelé dropout rate, il prend une valeur entre 0 et 1. Après que l'entraînement se termine, les neurones seront tous réactivés [11]. La deux schémas de la figure 3.10 montrent l'effet de l'application du *Dropout* sur un réseau de neurones standard

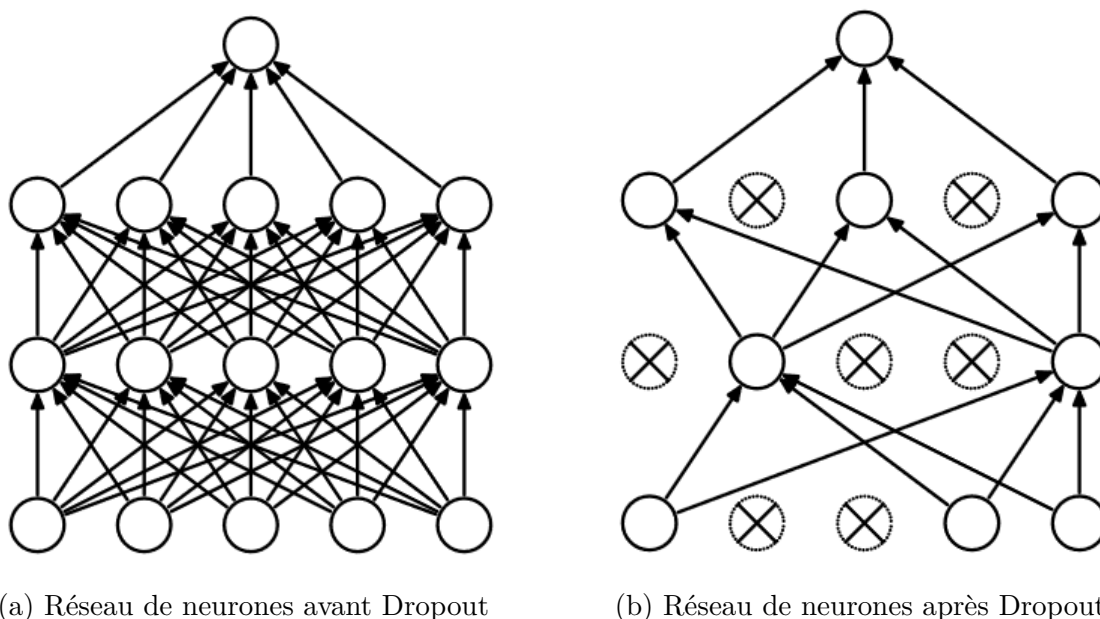


FIGURE 3.10: Illustration du Dropout lors de la phase d'entraînement

Le *Dropout* a de nombreux avantages. D'abord, il empêche efficacement le sur-apprentissage en réduisant la corrélation entre les neurones. Aussi, il permet d'avoir un réseau neuronal unique à chaque itération d'entraînement. En effet, chaque neurone peut être présent ou absent dans les différentes itérations, il en résulte un total de 2^N réseaux de neurones possible, N étant le nombre de neurones pouvant être désactivés. Le réseau neuronal résultant peut donc être considéré comme un ensemble de moyenne de tous ces réseaux de neurones.

3.5 Les hyperparamètres

Il est nécessaire de savoir évaluer n'importe quel algorithme d'apprentissage sur son jeu de données. Ceci se fait grâce au réglage des différents hyperparamètres du modèle proposé. Il est donc important de bien les choisir et les adapter à notre problème.

Un hyperparamètre est un paramètre dont la valeur est utilisée pour contrôler le processus

d'apprentissage. En revanche, les valeurs d'autres paramètres (généralement les poids des nœuds) sont dérivées via l'apprentissage. Donc, les hyperparamètres ne peuvent pas être déduits lors de l'apprentissage, ils doivent être ajustés. Les hyperparamètres nécessaires pour notre modèles sont les suivants :

- **Le pas d'apprentissage (Learning rate)** : C'est un hyperparamètre qui contrôle la taille du pas à chaque itération en fonction de l'erreur estimée à chaque fois que les poids du modèle sont mis à jour. Le choix du pas d'apprentissage est difficile car une valeur trop petite peut entraîner un long processus d'entraînement qui pourrait rester bloqué, tandis qu'une valeur trop grande peut entraîner l'apprentissage d'un ensemble de poids divergeant.
- **Taille et nombre de filtres** : Ces deux hyperparamètres concernent les filtres des couches de convolution. Il est important de bien les ajuster afin de d'extraire aux mieux les caractéristiques relatives au problème.
- **Taille du lot (Batch Size)** : La taille du lot définit le nombre d'échantillons qui seront propagés à travers le réseau à chaque itération.
- **Nombre de neurones** : Pour la partie FC, le nombre de neurones qui constituent chaque couche. Il est nécessaire d'adapter ce nombre aux caractéristiques extraites par la partie convolutive. Un petit nombre de neurones peut entraîner une mauvaise classification tandis qu'un grand nombre de neurones peut conduire le modèle à un sur-apprentissage.

3.6 Architecture adoptée

Pour la prédiction du genre à partir de l'écriture manuscrite nous avons adopté une architecture CNN plutôt simple qui comprend 6 couches de convolution, chacune suivie par une couche Average Pooling. La figure 3.11 illustre cette architecture. La partie FC est constituée quant à elle de 2 couches Dense. Le réseau se termine par un neurone qui donne en sortie une probabilité sur le genre de l'auteur. L'architecture présentée a été largement inspirée de l'architecture AlexNet [30] en y apportant quelques modifications. La raison pour laquelle nous n'avons pas opté pour le Transfer Learning est que le modèle a été entraîné sur une base de données qui est de nature très différentes de celle dont nous disposons. Cependant, l'architecture AlexNet reste relativement la mieux adaptée à notre utilisation car elle a déjà été utilisée pour la texture [46].

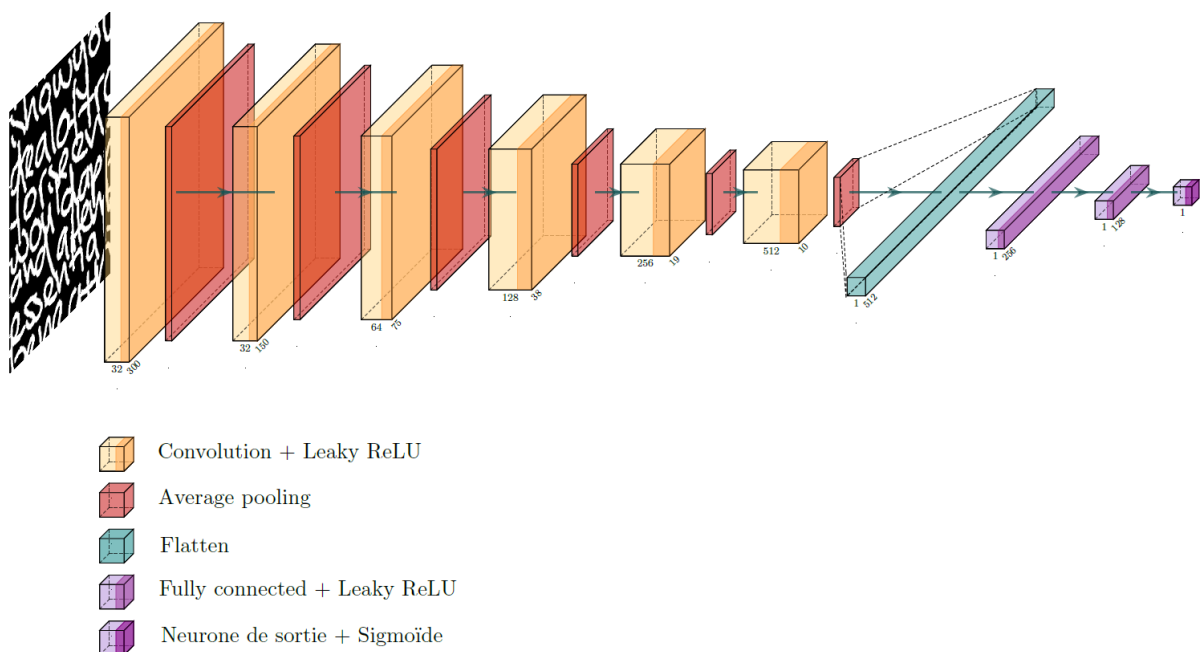


FIGURE 3.11: Architecture adoptée

L'architecture adoptée a été configurée selon les paramètres suivants :

■ La fonction coût

Comme notre problème est un problème de classification binaire qui consiste à associer l'écriture manuscrite au genre de son auteur, nous utilisons alors l'Entropie croisée binaire comme fonction coût.

■ Optimisateur

L'optimisateur le plus utilisé et le plus performant est l'optimisateur *Adam* [29]. Ce dernier est un algorithme d'optimisation du pas d'apprentissage adaptatif, spécialement conçu pour l'entraînement des réseaux de neurones profonds.

■ Couches d'activations

La couche d'activation *ReLU* est l'une des couches d'activations les plus utilisées dans les réseaux de neurones convolutifs. Nous avons utilisé dans ce travail une version améliorée de *ReLU*, il s'agit de la couche d'activation *Leaky ReLU* [50].

■ Régularisation

L'architecture proposée est assez profonde, il en résulte un grand nombre de paramètres à entraîner. Par conséquent, le problème du sur-apprentissage peut facilement avoir lieu. Afin d'éviter ce problème, nous avons utilisé le *Dropout* et le *Early Stopping*.

3.7 Extreme Learning Machine

En apprentissage automatique, le terme extreme learning machine (ELM) fait référence à un type de réseau de neurones «Inventé» en 2006 par G. Huang [23]. Sa spécificité est de n'avoir qu'une seule couche de nœuds cachés, où les poids des entrées de connexion de ces nœuds sont initialisés à des valeurs précises et jamais mis à jour. En théorie, ces modèles peuvent produire une bonne performance de généralisation et avoir un processus d'apprentissage beaucoup plus rapide que les réseaux de neurones ordinaires. ELM peut être appliqué avec les réseaux de neurones convolutifs. En effet, une couche de convolution peut être initialisée par des valeurs d'un autre filtre et ces dernières ne seront jamais mis à jour. Nous proposons dans ce travail, d'initialiser la première couche de convolution par un filtre de Gabor.

3.7.1 Filtre de Gabor

Les filtres de Gabor constituent une classe particulière des filtres linéaires ; ce sont des filtres orientés. Ces filtres ont une réponse impulsionnelle de la forme :

$$h(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right) \quad (3.10)$$

où :

$(x', y') = (x \cos \phi + y \sin \phi, -x \sin \phi + y \cos \phi)$, c'est-à-dire les coordonnées (x, y) tournées d'un angle ϕ .

La réponse impulsionnelle $h(x, y)$ est donc une fonction complexe sinusoïdale modulée par une gaussienne bidimensionnelle de rapport d'axes γ , de facteur de dilatation σ et où ϕ est l'orientation de l'axe x' par rapport à l'axe x . La représentation tridimensionnelle de la fonction de Gabor est donnée dans la figure 3.12.

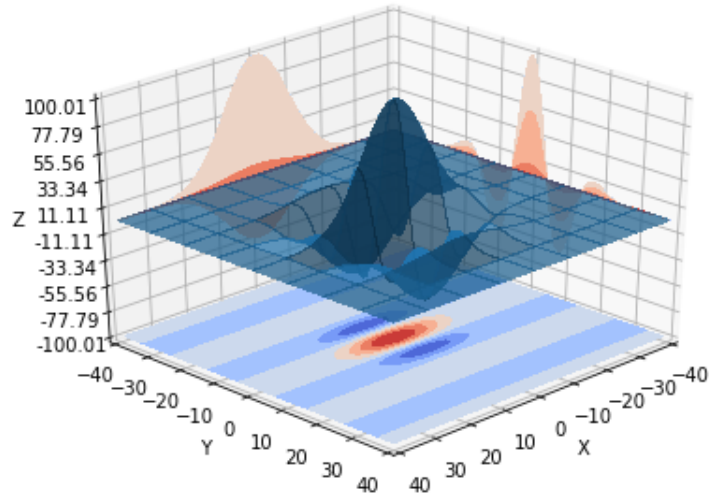


FIGURE 3.12: Filtre de Gabor

En appliquant cette fonction à un masque de convolution, on définit un filtre de convolution que nous appelons filtre de Gabor (Figure 3.13).

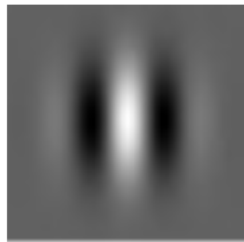


FIGURE 3.13: Masque du filtre de Gabor de rayon 21 pixels pour $\theta = 0^\circ$, $\sigma = 7$

Présentés ainsi, les filtres de Gabor jouissent de quatre degrés de liberté qui permettent d'extraire une partie de l'information spectrale de l'image. Il s'agit du rapport d'axes γ , du facteur de dilatation σ des gaussiennes dans les deux plans, de la longueur d'onde λ du sinusöide, et de la translation imposée par ψ . Ils permettent de mettre en évidence des textures ainsi que des zones homogènes d'une image. Grâce à la forme gaussienne des filtres, les enveloppes des images filtrées apportent une information spectrale locale en chaque pixel. De plus, elles renseignent sur le contenu énergétique de l'image dans la direction du filtre choisi [13]. Sur la figure 3.14, il est montré un exemple du filtre de Gabor appliqué sur une image texture.

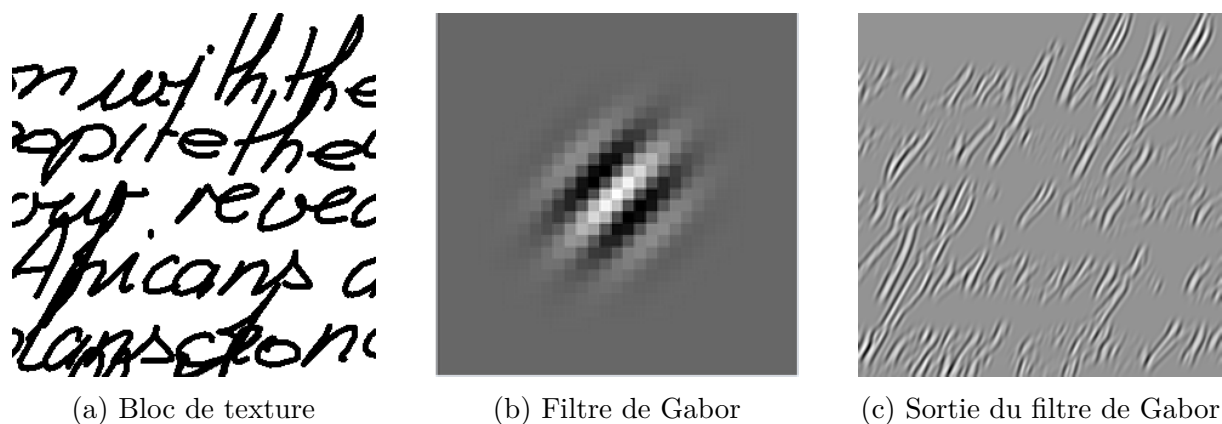


FIGURE 3.14: Filtre de Gabor appliqué sur une image texture

3.8 Implémentation du Filtre de Gabor dans l'architecture CNN

Les filtres de Gabor sont une classe particulière des filtres linéaires, ce sont des filtres orientés. Ils permettent de mettre en évidence des textures ainsi que des zones homogènes d'une image. Nous comprenons tout de suite que l'utilisation d'un tel filtre peut être intéressante compte tenu de la nature de nos données et les caractéristiques relatives au genre qui y sont présentes. La combinaison des filtres de Gabor avec les réseaux de neurones convolutifs pour l'analyse de l'écriture manuscrite est une méthode qui a été abordée pour la première fois en 2019 par *Maryam Asadzadeh Kaljahi et al.*, [28]. Cependant, ces derniers ont utilisé les filtres de Gabor en dehors de l'architecture des réseaux de neurones convolutifs. En effet, d'après l'article publié, 8 filtres de Gabor avec 8 orientations différentes ont été utilisés. Ces derniers sont ensuite convolués avec l'image requête et les 8 images résultantes sont fusionnées manuellement suivant un certain algorithme. A partir de là, nous avons pensé à introduire la génération des filtres de Gabor au sein de l'architecture CNN adopté. C'est particulièrement possible car les filtres de Gabor sont des filtres *linéaires*. Ceci a principalement été motivé par le fait que la fusion des filtres proposée dans l'article peut ne pas être adaptée à notre problème. Par conséquent, nous avons estimé qu'il serait mieux que la fusion des filtres soit apprise automatiquement par le réseaux. Ceci a été fait en incluant une couche convolutive, non entraînable, au début de l'architecture proposée dans la section précédente. Cette couche convolutive a été initialisée par les valeurs du filtre de Gabor. Les détails des paramètres utilisés pour le filtre de Gabor seront expliqués dans le protocole expérimental.

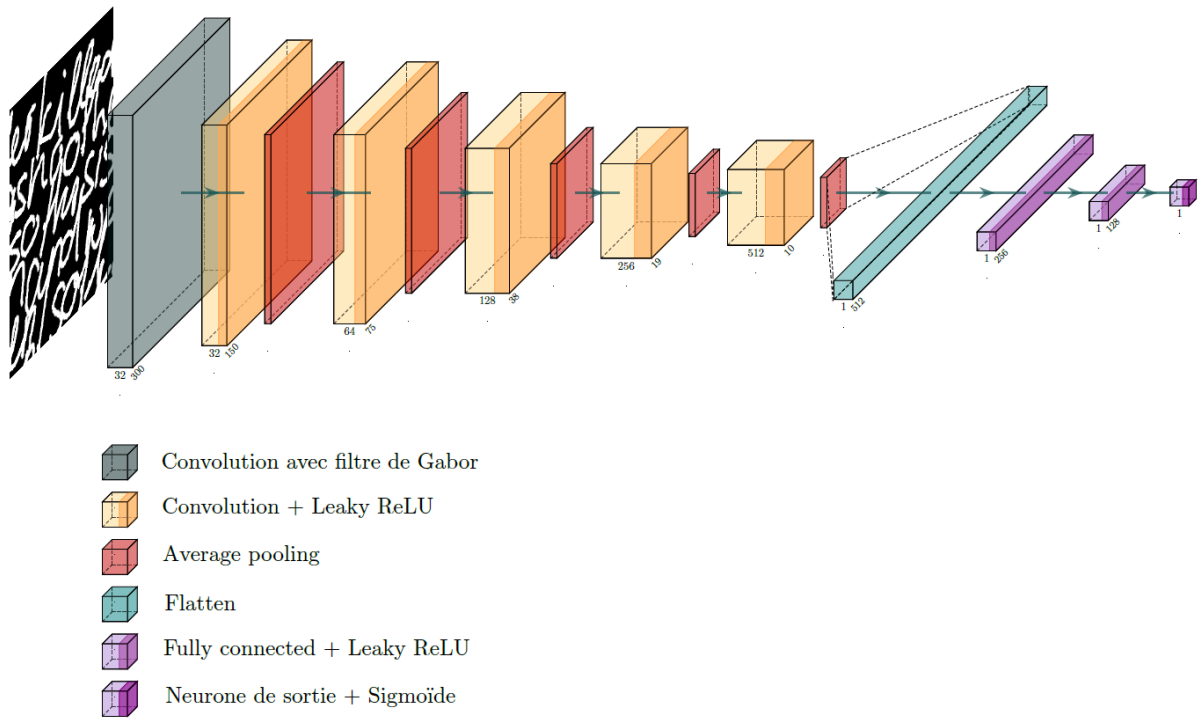


FIGURE 3.15: Filtre de Gabor dans l'architecture CNN

3.9 Conclusion

Il a été sujet dans ce chapitre de présenter les réseaux de neurones convolutifs. Nous avons commencé par définir les différentes parties formant un réseaux de neurones convolutifs. Nous avons ensuite expliqué comment s'entraînent les réseaux de neurones ainsi que les différentes méthode de régularisation qui existent pour faire face au sur-apprentissage. Nous avons aussi introduit la notion d'Extreme Learning Machine utilisant le filtre de Gabor pour la prédiction du genre à partir de l'écriture manuscrite. Les tests effectués et les résultats obtenus seront présentés et discutés dans le chapitre suivant.

Chapitre 4

Tests et résultats expérimentaux

4.1 Introduction

Ce chapitre est dédié aux différents tests effectués et aux résultats expérimentaux obtenus à partir des architectures établies pour la prédiction du genre à partir de l'écriture manuscrite en utilisant les images textures générées à partir des échantillons d'écritures manuscrites. Ces architectures se basent sur l'utilisation des réseaux de neurones convolutifs. Plus précisément, nous commençons par évaluer en utilisant les CNNs, l'influence de la taille des blocs textures ainsi que l'espace inter-lignes sur les performances du modèle. Ensuite, nous montrerons l'intérêt derrière l'utilisation des filtres de Gabor avec les images textures. Dans un second lieu, nous évaluerons sur la meilleure configuration (taille, espace inter-ligne) pour les blocs textures les résultats obtenus en utilisant les CNNs et les CNNs fusionnés avec les filtres de Gabor. Enfin, nous comparons les résultats obtenus avec l'état de l'art.

4.2 Description des bases de données

Pour ce travail, nous avons utilisé deux bases de données standards : KHATT et IAM. La première base contient des échantillons d'écriture manuscrite en Arabe tandis que la deuxième contient des échantillons d'écriture manuscrite en Anglais. Ces deux bases de données sont présentées dans ce qui suit.

4.2.1 La base KHATT

En 2014, Mahmoud et al., [36] ont publié un nouvel ensemble de données en Arabe, à savoir KHATT (KFUPM Handwritten Arabic Text) conçu pour servir la recherche dans différentes applications liées à la reconnaissance de l'écriture manuscrite Arabe. La base de données est composée de 1000 formulaires qui couvrent toutes les formes de caractères Arabes, remplis par 1000 scripteurs de 18 pays différents. Chaque écrivain a écrit six paragraphes, dont deux sous forme libre sur des sujets choisis par l'auteur lui-même. Les textes ont été segmentés en paragraphes puis en lignes de texte manuscrites. Cette base de données a été collectée en tenant compte du genre, de la qualification, la manualité, de la tranche d'âge et de la nationalité. Dans notre travail, nous avons utilisé les lignes de textes issues de 643 scripteurs : 347 scripteurs hommes et 296 scripteurs femmes. La figure 4.1 illustre quelques exemples de lignes de textes utilisées pour les deux genres.

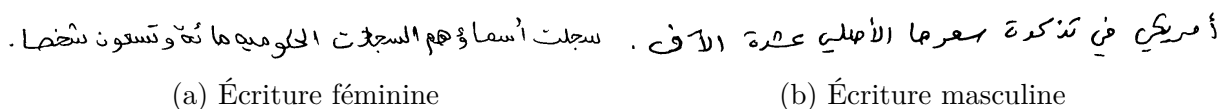


FIGURE 4.1: Lignes de texte manuscrites issues de la base KHATT

4.2.2 La base IAM

IAM (IAM-OnDB)¹ [33] est une base de données dite en ligne (On-line), qui comporte des écritures manuscrites en langue Anglaise. Une version hors ligne «off-line» est obtenue en projetant les coordonnées en ligne sur un plan bidimensionnel. Cette base de données a été développée par un groupe de chercheurs dans le domaine de l'intelligence artificielle, à l'Université de Bern en Suisse. Elle a été publiée pour la première fois en 1999 [37]. Elle a été collectée avec la contribution de 657 scripteurs de différent âge, genre, manualité et niveau éducatif. Cette base de données peut être utilisée pour différentes applications liées à la reconnaissance de l'écriture manuscrite, telles que l'identification/vérification du scripteur, la reconnaissance de mots, la prédiction des caractéristiques soft-biométriques, etc. Cependant, les informations liées au genre ne sont disponibles que pour 217 scripteurs. Chacun a participé avec huit textes, sans aucune contrainte imposée. Chaque scripteur a écrit en moyenne sept lignes, donnant un nombre totale de lignes de texte égale à 13017. Les deux genres sont inégalement présents avec 143 hommes contre 74 femmes. Dans notre travail, nous avons utilisé les lignes de textes déjà segmentées et disponibles gratuitement, issues de 166 scripteurs : 92 scripteurs hommes et 74 scripteurs femmes. La figure 4.2 présente quelques exemples de ligne de texte de la base IAM pour les deux genres.

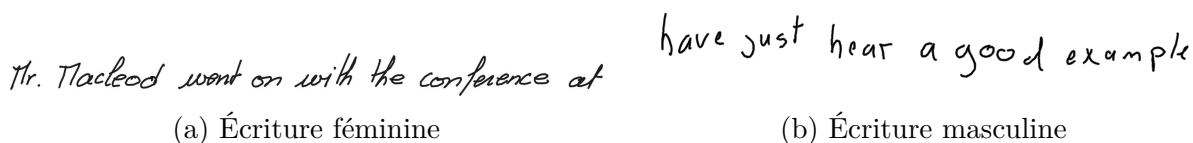


FIGURE 4.2: Lignes de texte manuscrites issues de la base IAM

4.3 Logiciels, bibliothèques et matériel utilisés

Dans cette section, nous présentons un bref aperçu du langage de programmation et des outils logiciels/bibliothèques utilisés dans notre travail, à savoir, Python², OpenCV³, TensorFlow⁴, Keras⁵, Google Colab⁶, ainsi que le matériel utilisé.

1. <http://www.fki.inf.unibe.ch/databases/iam-on-line-handwriting-database>.

2. python.org/.

3. opencv.org/.

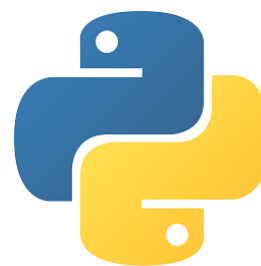
4. tensorflow.org/.

5. <https://keras.io/>.

6. <https://colab.research.google.com/notebooks/intro.ipynb>.

4.3.1 Python

Python est un langage puissant permettant le développement de projets complexes et ambitieux grâce aux nombreuses bibliothèques qui lui sont associées, notamment dans le domaine du Deep learning où il est très sollicité pour la stabilité de ses bibliothèques et leurs robustesses pour implémenter les réseaux de neurones. Dans ce travail, nous avons utilisé Python version 3.7.



4.3.2 OpenCV

OpenCV (Open Source Computer Vision) est une bibliothèque graphique open source proposant un ensemble d'algorithmes de vision par ordinateur et d'apprentissage automatique pour le traitement des images. Elle est adaptée à plusieurs langages de programmation tels que C++ et Python. Dans ce travail, nous avons utilisé la version 4.0.1.



4.3.3 TensorFlow

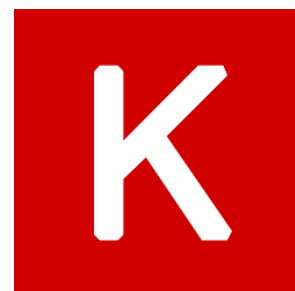
Développé par les chercheurs de Google, TensorFlow est un outil open source d'apprentissage automatique (Machine Learning) et d'analyses statistiques et prédictives. Il convient particulièrement bien aux applications à traitement parallèle à très grande échelle : les réseaux neuronaux en sont un exemple courant. La version utilisée dans ce travail est la 2.2.0.

L'un des outils puissants qu'offre TensorFlow est TensorBoard. TensorBoard fournit les solutions de visualisation et les outils nécessaires aux tests de l'apprentissage automatique. Il permet d'héberger, de suivre et de partager facilement les test et les résultats.



4.3.4 Keras

Keras est une API de réseaux de neurones de haut niveau. Elle s'appuie sur TensorFlow qui est bien optimisé pour les opérations de bas niveau. Elle a été développée avec pour objectif de permettre des expérimentations rapides. Dans notre travail, la version utilisée est la 2.3.1.



4.3.5 Google Colab

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook. Il est destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles d'apprentissage automatique directement dans le Cloud, sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur.



4.3.6 Matériel utilisé

Pour la partie pré-traitement des données, nous avons utilisés nos PC personnels. Le pré-traitement a été effectué sur deux PC : Dell Xps et Dell Latitude. Ils sont dotés du matériel suivant :

- Pour le Dell XPS : un processeur intel i7-6700HQ 2.6GHz, une RAM de 16GB DDR4, un Disk 512GB SSD et un système d'exploitation Windows 10 Pro.
- Pour le Dell Latitude : un processeur intel i5-6300U 2.4GHz, une RAM de 16GB DDR4, un Disk 256GB SSD et un système d'exploitation Windows 10 Pro.

Après avoir pré-traité les données, nous sommes passés à Google Colab Pro pour la carte graphique puissante qu'il propose. Ce dernier est doté d'un processeur Intel Xeon CPU 2.30GHz, une RAM de 25GB, un disque de 110GB, un système d'exploitation Linux-4.19.112+-x86_64-with-Ubuntu-18.04-bionic, et un processeur graphique (GPU) Tesla P100-PCIE.

4.4 Critères d'évaluation

Afin d'évaluer au mieux les performances des différentes architectures que nous avons implémenté, il est essentiel d'avoir recours à plusieurs critères d'évaluation définis dans ce qui suit.

4.4.1 Le taux de bonne prédiction

La première métrique utilisée pour l'évaluation des performances de notre système est le Taux de Bonne Prédiction (TBP). Le TBP ou *accuracy* représente le nombre de prédictions correctes sur le nombre total de prédictions. Soient \hat{P} , P et N la classe prédite, la classe réelle et le nombre total de prédiction, respectivement ; alors le TBP est décrit par l'équation suivante :

$$\text{TBP} = \frac{1}{N} \sum_{i=1}^N f(\hat{P}_i, P_i) \quad (4.1)$$

où f est définie par :

$$f(a, b) = \begin{cases} 1 & \text{si } a = b \\ 0 & \text{si } a \neq b \end{cases}$$

4.4.2 Matrice de confusion

La Matrice de confusion indique le nombre de prédictions correctes et incorrectes pour chaque classe. Chaque ligne du tableau correspond à une classe prédite, et chaque colonne correspond à une classe réelle, comme l'indique la matrice de la figure 4.3. Cette matrice permet d'analyser l'erreur commise par le classifieur lorsqu'il effectue des prédictions.

		Classes réelles	
		Positive Classe A	Négative Classe B
Classes prédites	Positive Classe A	Vrais Positifs	Faux Positifs
	Négative Classe B	Faux Négatifs	Vrais Négatifs

FIGURE 4.3: Matrice de Confusion

4.4.3 Courbe ROC

Une courbe ROC (Receiver Operating Characteristic) est un graphique représentant les performances d'un modèle de classification pour tous les seuils de classification. Cette courbe trace le taux de vrais positifs en fonction du taux de faux positifs pour différents seuils de classification : Le taux de vrais positifs (TVP) est l'équivalent du rappel. Il est donc défini comme suit :

$$TVP = \frac{VP}{VP + FN}$$

Le taux de faux positifs (TFP) est défini comme suit :

$$TFP = \frac{FP}{FP + VN}$$

où VP est l'abréviation de Vrais positifs, FN l'abréviation de Faux Négatifs, FP Faux Positifs et enfin, VN Vrais Négatifs.

Diminuer la valeur du seuil de classification permet de classer plus d'éléments comme positifs, ce qui augmente le nombre de faux positifs et de vrais positifs.

4.4.4 AUC : aire sous la courbe ROC

AUC (Area Under Curve) signifie aire sous la courbe ROC. Cette valeur mesure l'intégralité de l'aire à deux dimensions située sous l'ensemble de la courbe ROC (par calculs d'intégrales) de (0,0) à (1,1).

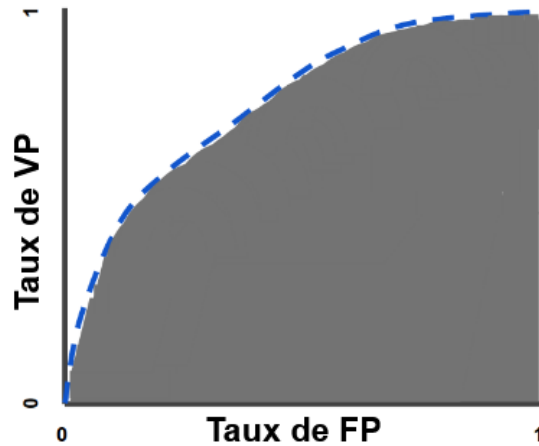


FIGURE 4.4: AUC (aire sous la courbe ROC)

L'AUC fournit une mesure agrégée des performances pour tous les seuils de classification possibles. On peut interpréter l'AUC comme une mesure de la probabilité pour que le modèle classe un exemple positif aléatoire au-dessus d'un exemple négatif aléatoire. Les valeurs d'AUC sont comprises entre 0 et 1. Un modèle dont 100% des prédictions sont erronées a un AUC de 0. Si toutes ses prédictions sont correctes, son AUC est de 1. L'AUC présente les avantages suivants [12] :

- Elle est invariante d'échelle. Elle mesure la qualité du classement des prédictions, plutôt que leurs valeurs absolues.
- Elle est indépendante des seuils de classification. Elle mesure la qualité des précisions du modèle quel que soit le seuil de classification sélectionné.

4.5 Protocole expérimental

La conception d'un système de prédiction du genre à partir de l'écriture manuscrite nécessite le choix d'une procédure afin d'obtenir des résultats optimaux. Plus précisément, une étude sur l'influence de la taille ainsi que l'espace inter-ligne de l'image texture sera abordée. Aussi, nous expliquerons la répartition des données et la génération des images texture ainsi que le réglage des hyperparamètres du CNN.

4.5.1 Répartition des données

L'objectif fondamental de l'apprentissage automatique est de généraliser au-delà des données utilisées pour entraîner le modèle. Nous voulons évaluer le modèle pour estimer la qualité de sa généralisation pour des données sur lesquelles il n'a pas été entraîné. De façon générale, pour la plupart des applications de reconnaissance de formes, trois ensembles de données sont nécessaires :

1. Ensemble de données d'entraînement : C'est les échantillons de données utilisés pour ajuster le modèle, c'est-à-dire l'ensemble de données utilisé pour former le modèle (adaptation des poids et des biais). Le modèle observe et apprend de ces données et optimise ses paramètres. Dans notre travail, 80% de la totalité des données sont utilisés comme ensemble d'entraînement.
2. Ensemble de données de validation : Ce sont les données utilisées pour que le modèle évalue les paramètres appris à travers l'ensemble d'entraînement. Cet ensemble est sollicité à chaque nouvelle mise à jour des paramètres. Dans notre travail, 10% de la totalité des données sont utilisés comme ensemble de validation.
3. Ensemble de données test : Après avoir établi un modèle fiable, il est généralisé sur de nouvelles données appelées données test, pour tester sa performance et sa fiabilité. Dans notre travail, les 10% restant de la totalité des données sont utilisés comme ensemble de test.

Noter que les images textures sont générées par scripteur. En d'autres termes, tous les échantillons d'écriture d'un même scripteur sont utilisés pour former les images textures et les blocs textures par la suite. Pour les deux bases de données utilisées dans ce travail à savoir KHATT et IAM, nous constituons les 3 ensembles de données, entraînement, validation et test de façon à ce que les blocs textures d'un même scripteur ne soient présents que dans l'un des ensembles. Ceci est essentiel pour que le modèle apprenne au mieux la relation existante entre les blocs textures et le genre de l'auteur au lieu d'apprendre le style d'écriture de l'auteur, ce qui risque de mener sur une identification du scripteur. Nous résumons, dans les deux tableaux 4.1 et 4.2 la répartition des données adoptée pour les deux bases : KHATT et IAM.

TABLE 4.1: Répartition de la base de données KHATT

	Hommes			Femmes		
	Nombre de blocs : 3210			Nombre de blocs : 3210		
	Nombre de scripteurs : 347			Nombre de scripteurs : 296		
	E	V	T	E	V	T
Nombre de blocs	2495	360	355	2495	360	355
Nombre de scripteurs	238	52	57	185	57	54

E : Ensemble de données Entraînement. V : Ensemble de données Validation. T : Ensemble de données Test.

TABLE 4.2: Répartition de la base de données IAM

	Hommes			Femmes		
	Nombre de blocs : 3,070			Nombre de blocs : 3,070		
	Nombre de scripteurs : 92			Nombre de scripteurs : 74		
	E	V	T	E	V	T
Nombre de blocs	2,460	305	305	2,460	305	305
Nombre de scripteurs	77	7	8	61	7	6

E : Ensemble de données Entraînement. V : Ensemble de données Validation. T : Ensemble de données Test.

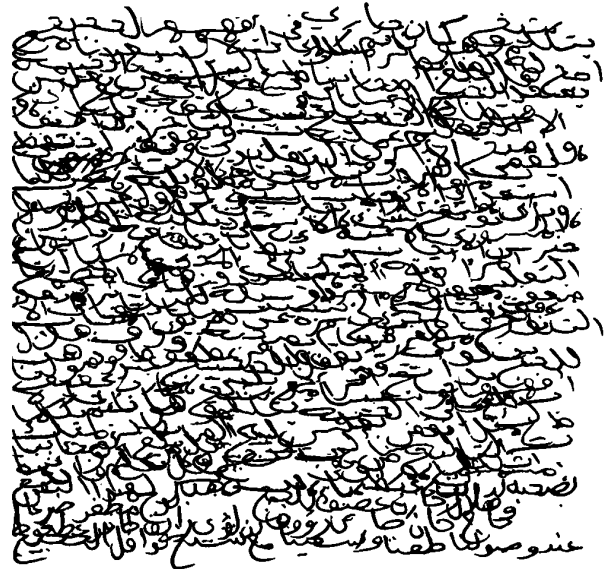
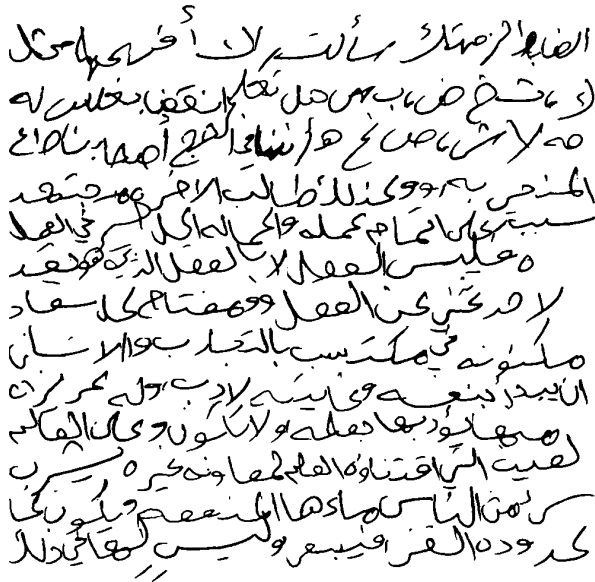
4.5.2 Influence de la taille du bloc texture et l'espace inter-lignes sur les performances du modèles

La taille du bloc texture est un paramètre très important qui a un effet direct sur les performances du modèle. Un compromis entre l'information présente dans un seul bloc et la taille du bloc doit être trouvé. Il est important donc de tenir compte de ces deux hypothèses abordés dans le livre de François Chollet [11], développeur de Keras :

- Les sorties (le genre) doivent pouvoir être prédites compte tenu des entrées (les blocs texture) et,
- Les données disponibles doivent être suffisamment informatives pour apprendre la relation entre les entrées et les sorties.

De ce fait, dans un bloc texture, il devrait y avoir l'information nécessaire relative au genre qui peut permettre au CNN de bien l'exploiter afin de trouver une relation entre l'écriture manuscrite et le genre de son auteur. D'autre part, il est important de prendre en considération une limite supérieure de la taille du bloc de texture. Un grand bloc texture exige plus de performances et de ressources pour le CNN. Aussi, en choisissant une grande taille, nous réduirons considérablement le nombre de blocs textures présents dans chaque base de données, ce qui n'est pas en faveur du réseau profond proposé (risque de sur-apprentissage).

D'autre part, l'espace inter-ligne est un paramètre clé dans le processus de génération des images textures, ceci doit être choisi de manière rigoureuse afin de ne pas perdre la notion de la texture en choisissant un espace inter-lignes très grand, et de ne pas aussi perdre la structure géométrique de l'écriture en choisissant un espace inter-ligne très petit. La figure 4.5 montre deux images textures. L'une générée avec un espacement inter-ligne de 0.50 et l'autre avec un espace inter-ligne de 1.25.



(a) Image texture avec un espace inter-lignes de 1.25

(b) Image texture avec un espace inter-lignes de 0.50

FIGURE 4.5: Images textures générées à partir de différents espaces inter-lignes

Nous voyons que pour la valeur 0.50, l'écriture est beaucoup condensée et la structure s'est détériorée tandis que pour la valeur de 1.25, on voit que l'aspect de la texture a totalement disparu. Une valeur entre ces deux doit être trouvée.

Afin de trouver les paramètres optimaux qui donnent les meilleures performances, nous avons procédé à un Grid Search pour les deux bases de données, comprenant les variantes suivantes :

- Cinq différentes tailles ont été testées : 80x80, 100x100, 150x150, 300x300, 400x400 et,
- Cinq différents espaces inter-lignes ont été testés : 0.50, 0.60, 0.75, 1.00, 1.25.

Les différentes test ont été effectués sur l'architecture CNN présentée dans le chapitre 3. Les résultats pour les deux bases de données sont montrées dans les tableaux 4.3 et 4.4.

TABLE 4.3: Performances en terme de TBP (%) pour les différentes tailles des blocs textures - base KHATT

		Espace inter-ligne				
		0.50	0.60	0.75	1.00	1.25
Taille du Bloc	80x80	65.33	66.28	63.20	63.02	61.09
	100x100	71.38	72.00	71.93	70.03	68.30
	150x150	72.06	73.13	70.06	71.27	69.40
	300x300	75.18	80.87	76.60	74.166	73.09
	400x400	74.68	76.42	74.78	71.30	70.01

TABLE 4.4: Performances en terme de TBP (%) pour les différentes tailles des blocs textures - base IAM

		Espace inter-ligne				
		0.50	0.60	0.75	1.00	1.25
Taille du Bloc	80x80	61.34	62.02	60.30	59.29	58.89
	100x100	62.71	67.30	61.42	62.03	60.12
	150x150	64.06	74.75	63.01	64.54	63.31
	300x300	66.01	79.38	69.90	68.6	68.29
	400x400	65.53	74.29	67.20	69.48	69.03

D'après les résultats obtenus, nous pouvons conclure que, pour les deux bases de données :

- Une taille trop petite d'un bloc texture ne permet pas d'inclure les caractéristiques nécessaires relatives au genre,
- Une taille trop grande d'un bloc texture détériore les performances du modèle,
- Une taille de 300x300 est la taille optimale trouvée pour un bloc de texture pour les deux bases de données,
- Un espace inter-ligne très petit déforme effectivement la structure de l'écriture manuscrite et a un effet négatif sur les performances du modèle,
- Un espace inter-ligne très grand fait perdre la notion de l'image texture et par conséquent détériore les performances du modèle,
- Un espace inter-ligne optimale de 0.60 a été trouvé pour les deux bases de données.

4.5.3 Application des filtres de Gabor sur les images textures

Comme ça été expliqué dans le chapitre 3, les filtres de Gabor permettent de mettre en évidence des textures ainsi que des zones homogènes d'une image. Concrètement, pour les images textures que nous avons générées, 3 caractéristiques sont mis en valeur par les filtres de Gabor : L'orientation, l'épaisseur et l'espacement.

Les filtres de Gabor permettent d'isoler les contours d'une image d'orientation parallèle à ϕ répondant à une certaine épaisseur, qui dépend de λ . Ceci justifie le fait que nous avons généré plusieurs filtres de Gabor. Ainsi, l'association d'un ensemble de filtres de Gabor permet de couvrir plus largement l'espace fréquentiel et de détecter un plus grand nombre d'orientations, et donc d'extraire tous les contours de l'image. Dans la figure ci-dessous, nous montrons l'application des filtres de Gabor sur des images textures afin d'explicitier l'influence des différents paramètres.

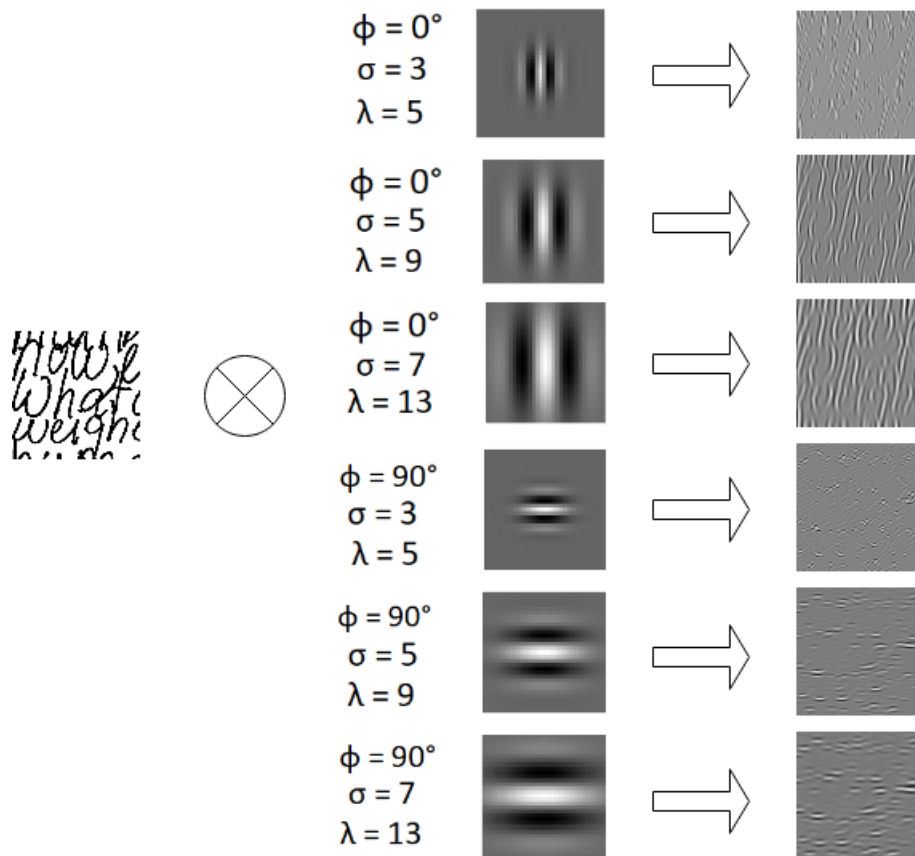


FIGURE 4.6: Application des filtres de Gabor sur les images textures

Nous voyons dans la figure ci-dessus l'influence des filtres de Gabor pour l'image texture d'exemple. ϕ a une influence sur l'orientation de l'écriture. On voit que pour $\phi = 0^\circ$, l'image de sortie est bien activée (traits blancs), ce qui veut dire que l'image texture d'entrée comporte une écriture manuscrite qui comporte beaucoup de traits verticaux. λ a une influence directe sur l'épaisseur de l'écriture manuscrite. On voit que des valeurs de λ petit, l'image résultante est presque grise, ce qui veut dire que l'image texture en question présente une écriture épaisse. Enfin, σ influence sur la netteté des contours.

4.5.4 Le réglage des hyperparamètres

Le modèle que nous avons proposé est profond, par conséquent, il contient plusieurs hyperparamètres. Les hyperparamètres qui doivent être fixés sont diverses : Taille des filtres, nombre de filtres, le Stride, la valeur du Dropout, le pas d'apprentissage (learning rate), le nombre de neurones dans la couche FC, et la taille du lot (Batch Size) : une infinité de combinaison existe pour tous ces paramètres! De ce fait, nous nous sommes basés sur l'utilisation d'une grille de recherche ou Grid Search en utilisant *TensorBoard*. Les paramètres qui ont été testés sont le pas d'apprentissage, le nombre de neurones dans la couche FC, la taille du lot (Batch Size) et la valeur du Dropout. Les valeurs des paramètres qui ont été testées sont présentés dans le tableaux 4.5.

TABLE 4.5: Les différentes valeurs testées des hyperparamètres pour les deux bases de données

	Pas d'apprentissage	Nombre de neurones	Taille du Lot	Valeur Dropout
KHATT	[0.0001 - 0.0004]	[128 - 256]	[32 - 64]	[0.2 - 0.4]
IAM	[0.00005 - 0.00007]	[128 - 256]	[32 - 64]	[0.2 - 0.4]

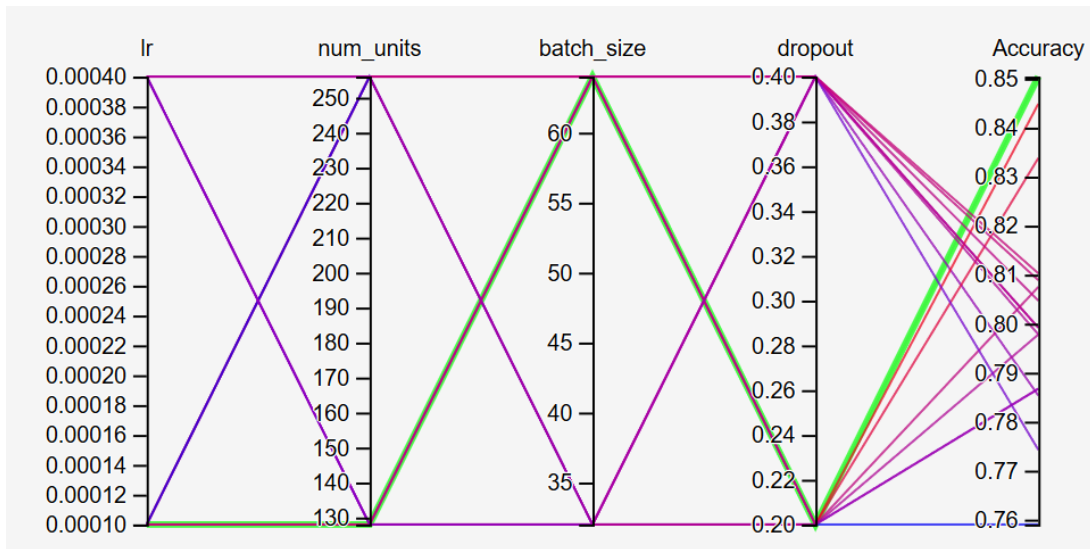
Pour la taille et le nombre de filtres dans chaque couche de convolution, nous sommes arrivés aux valeurs empiriques suivantes :

- Pour la base KHATT : La taille des filtres est fixée à 3x3 pour tous les filtres. Le nombre de filtre est respectivement, suivant l'ordre des couches du plus haut au plus bas : 32, 64, 128, 256, 256, 512. Tous les filtres ont été initialisés par he-normal⁷. Aussi, un *padding* pareil (same) a été utilisé.
- Pour la base IAM : La taille des filtres est fixée à 11x11 et 5x5 pour la première et la deuxième couche respectivement. Pour les couches qui suivent, la taille des filtres est fixée à 3x3. Le nombre de filtre est respectivement, suivant l'ordre des couches du plus haut au plus bas : 32, 32, 64, 128, 256, 512. Tous les filtres ont été initialisés par he-normal. Aussi, un *padding* pareil (same) a été utilisé.

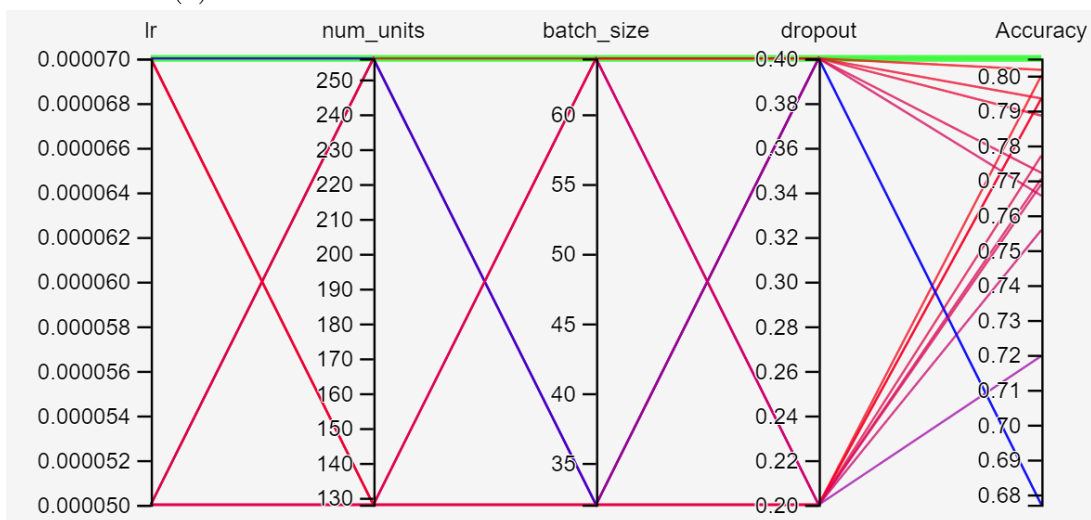
Les résultats des différents tests, pour les deux bases, sont présentées respectivement dans la figure 4.7. Celle-ci représente un résumé de tous les test effectués, avec comme critère d'évaluation le taux de bonne prédiction (TBP) sur les données validation (validation accuracy). Pour retrouver les paramètres optimaux, la ligne surlignée en vert représente donc les paramètres optimaux pour le meilleur TBP.

- Pour la base KHATT : Le nombre de neurones est 256 neurones pour la première couche FC et de 128 neurones pour la deuxième couche FC, le dropout est de 20%, utiliser après la couche d'aplatissement et chaque couches FC, le pas d'entraînement est de 0.0001 et la taille du lot est de 64.
- Pour la base IAM : Le nombre de neurones est 512 neurones pour la première couche FC et 256 neurones pour la deuxième couche FC, le dropout est de 40%, utiliser après la couche d'aplatissement et chaque couches FC, le pas d'entraînement est de 0.00007 et la taille du lot est de 64.

7. he-normal initialise les poids selon une distribution normal de telle façon à avoir une moyenne 0 et une certaine variance Var .



(a) Résumé des différents test effectués sur la base KHATT

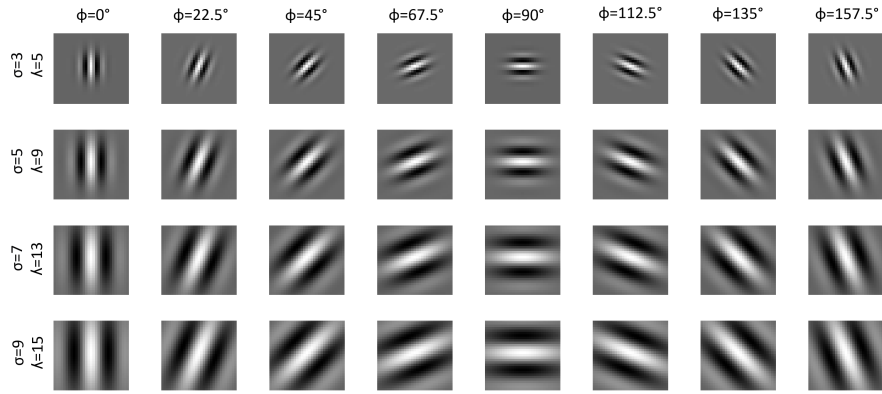


(b) Résumé des différents test effectués sur la base IAM

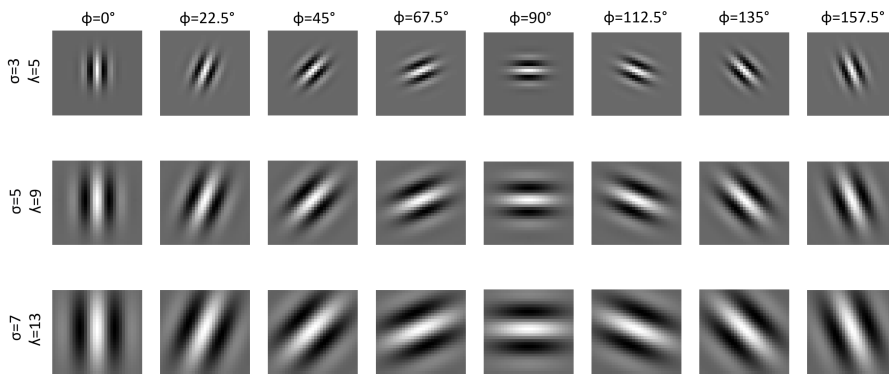
FIGURE 4.7: Recherche des paramètres optimaux

Pour l'implémentation du filtre de Gabor avec le CNN, nous avons gardé les mêmes paramètres qui ont donné le meilleur résultats avec le CNN seul. Les paramètres du filtre de Gabor empiriques trouvées pour les deux bases sont les suivantes :

- Pour la base KHATT : 32 filtres de Gabor ont été utilisés avec 8 orientations distribuées entre 0° et 157.5° et avec 4 valeurs pour λ : 5, 9, 13 et 15. La valeur de γ est fixée à 0.8, la valeur de ϕ à 0 et σ prends les valeurs 3, 5, 7 et 9. Les filtres générés sont présentés dans la figure 4.8a.
- Pour la base IAM : 24 filtres de Gabor ont été utilisés avec 8 orientations de 0° à 157.5° . Les valeurs de λ sont : 5, 9, et 13. Les valeurs de σ sont : 3, 5 et 7 respectivement. La valeur de γ est fixée à 0.8, et enfin, la valeur de ϕ à 0. Les filtres générés sont présentés dans la figure 4.8b.



(a) KHATT



(b) IAM

FIGURE 4.8: Les filtres de Gabor utilisés

4.6 Résultats obtenus

Cette section est dédiée aux différents résultats obtenus avec les deux architectures proposées pour la prédiction du genre à partir de l'écriture manuscrite pour les deux bases de données KHATT et IAM. Nous montrerons à travers un exemple l'effet des filtres de Gabor sur les images texture. Une comparaison et une discussion sera tout aussi abordée. Enfin, nous comparons les résultats obtenus avec l'état de l'art.

4.6.1 CNN sans filtre de Gabor

Pour la base de données KHATT, l'architecture CNN proposée engendre 2,322,433 paramètres. Les résultats présentés ont été obtenus après 80 itérations d'apprentissage. Pour la base KHATT, le *Early Stopping* a été utilisé avec une patience de 15 itérations et avec une patience de 20 itérations pour la base IAM. Pour la base de donnée IAM, l'architecture CNN proposée engendre 1,991,457 paramètres, les résultats présentés ont

été obtenus après 100 itérations d'apprentissage.

Les figures 4.9 et 4.10 représentent l'évolution du taux de bonne prédiction en fonction des itérations pour les bases KHATT et IAM, respectivement, pour les données d'entraînement et de validation.

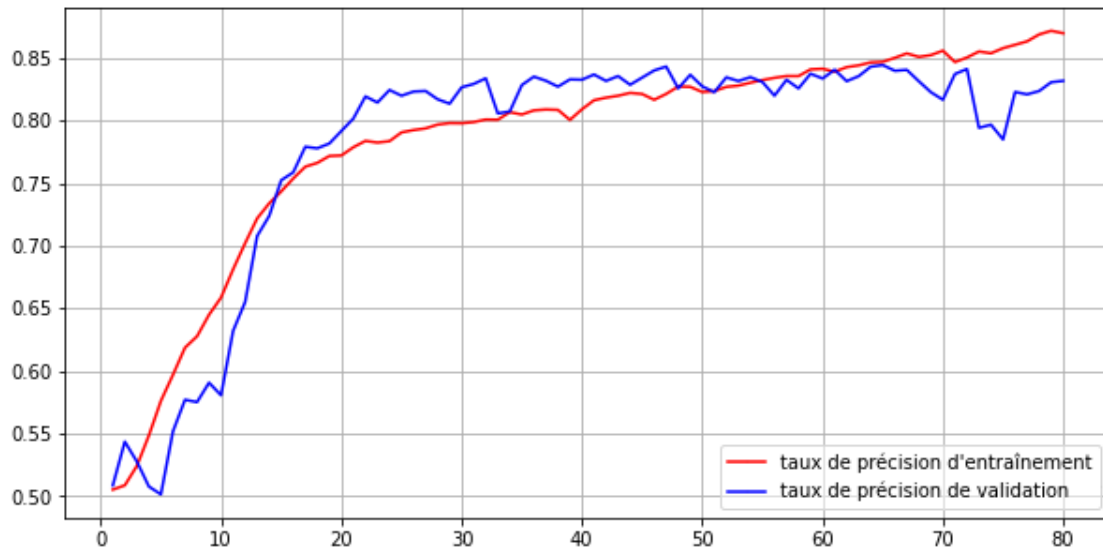


FIGURE 4.9: Taux de bonne prédiction d'entraînement et de validation pour la base KHATT (*100%)

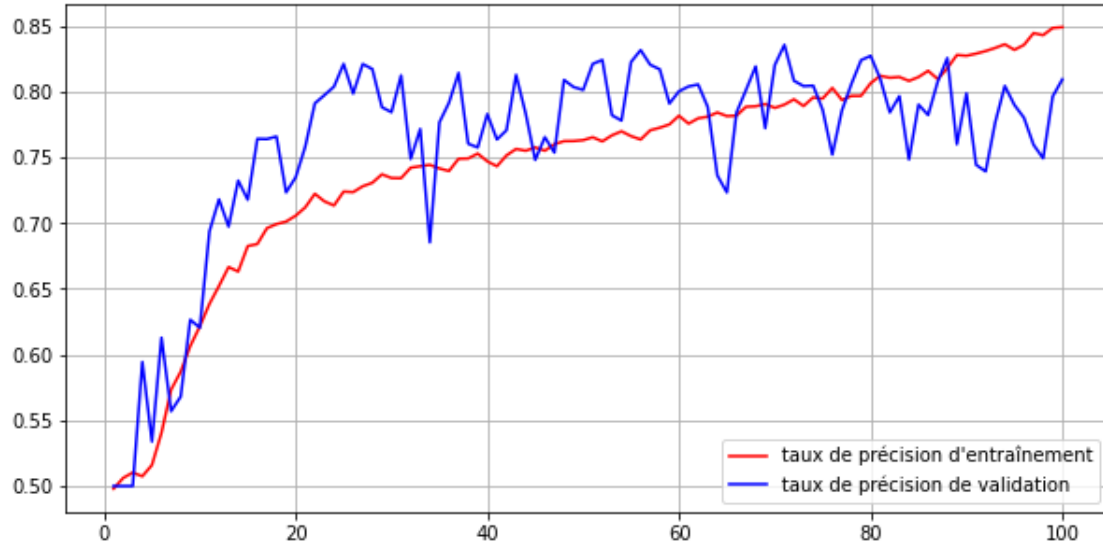


FIGURE 4.10: Taux de bonne prédiction d'entraînement et de validation pour la base IAM (*100%)

Les figures 4.11 et 4.12 représentent l'erreur commise en fonction du nombre d'itérations pour la base KHATT et IAM, respectivement, des données d'entraînement et de validation. Il est à mentionner que pour la base KHATT, le modèle pris est celui qui a été arrêté à la 65^{ème} itérations d'apprentissage, juste avant que le modèle tombe dans le sur-apprentissage, de même pour la base IAM, le modèle pris est celui qui a été arrêté à la

80^{ème}. Le TBP final atteint pour la base KHATT en utilisant l'architecture CNN proposée est de 84.66%, tandis que le taux final atteint pour la base IAM est de 81.26% pour les données validation. Nous observons que le TBP appliqué aux données d'entraînement et de validation augmente au fil des itérations ce qui démontre que le réseau apprend et généralise correctement sur les données de validation. Le modèle est ainsi testé a travers les données test ou des taux de bonne prédiction de 86,17% et 81,80% sont respectivement atteints pour les bases KHATT et IAM.

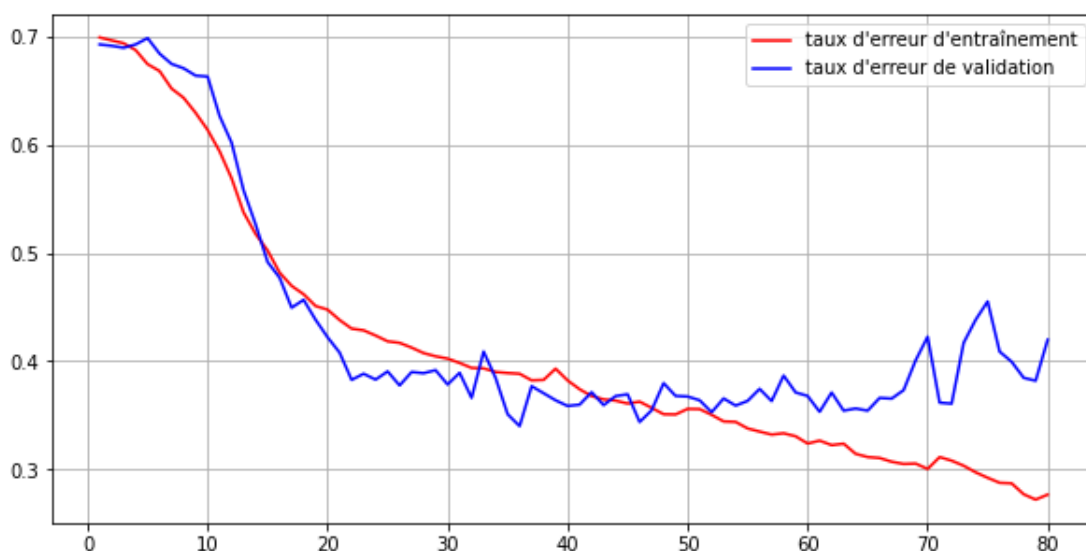


FIGURE 4.11: Erreur commise d'entraînement et de validation pour la base KHATT

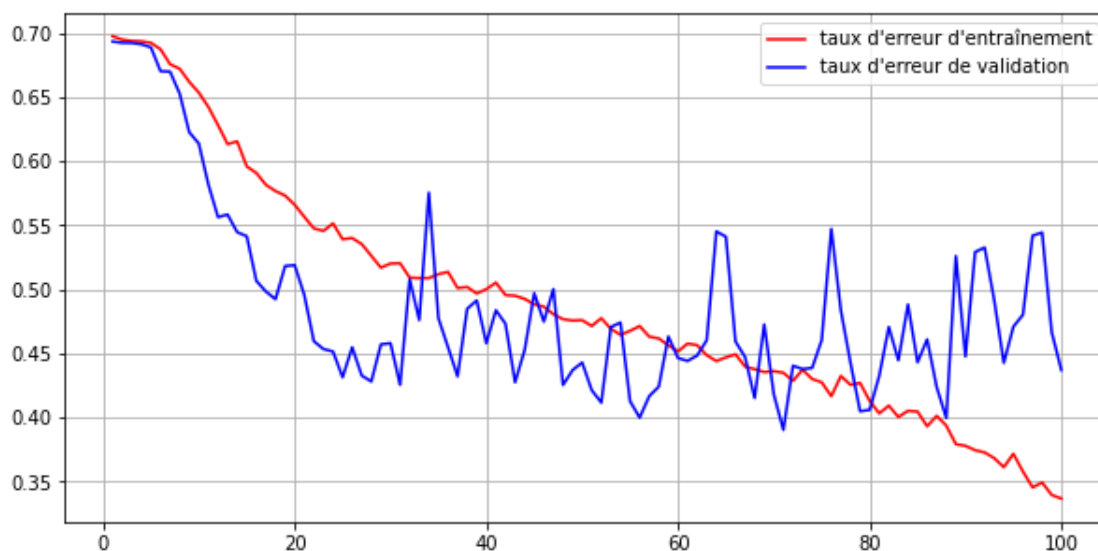


FIGURE 4.12: Erreur commise d'entraînement et de validation pour la base IAM

L'erreur finale atteinte pour la base KHATT en utilisant l'architecture CNN proposée est de 0.362 tandis que l'erreur finale atteinte pour la base IAM est de 0.405. Nous observons que l'erreur commise pour les données d'entraînement et de validation diminue au fil des itérations. Pour la base de donnée KHATT, l'erreur commise pour

les données validations augmente à partir de la 65^{ème} itérations, raison pour laquelle l'entraînement a été arrêté à la 80^{ème} itérations. Pour la base de donnée IAM, l'erreur commise pour les données validations augmente à partir de la 80^{ème} itérations, raison pour laquelle l'entraînement a été arrêté à la 100^{ème} itérations.

Afin de tester les performances du modèle, une évaluation doit être faite sur les données inconnues (données test). L'erreur atteinte pour les données test de la base KHATT est de 0.339 alors que l'erreur atteinte pour les données test de la base IAM est de 0.401

Notre réseau atteint un AUC de 0.936 pour la base KHATT et un AUC de 0.903 pour la base IAM, ce qui indique la qualité du classement des prédictions. Afin de mieux apprécier et interpréter les résultats atteints par le modèle, nous proposons de calculer la matrice de confusion. La matrice de confusion pour les deux bases KHATT et IAM est présentée dans les tableaux 4.6 et 4.7, respectivement.

TABLE 4.6: Matrice de confusion pour la base KHATT (%)

		Classes réelles	
		Hommes	Femmes
Classes prédites	Hommes	87.01	14.71
	Femmes	12.99	85.29

TABLE 4.7: Matrice de confusion pour la base IAM (%)

		Classes réelles	
		Hommes	Femmes
Classes prédites	Hommes	77.38	13.77
	Femmes	22.62	86.23

La matrice de confusion nous donne donc un aperçu sur le taux de de bonne prédiction par classe. En effet, nous observons, pour la base KHATT, que la classe Hommes a été prédite correctement avec un taux de 87.01% tandis que la classe Femmes a été prédite correctement avec un taux de 85.29%. Ceci peut être dû au fait que l'écriture féminine est plus compliquée à prédire que l'écriture masculine. Cependant, pour la base IAM, la classe Hommes a été prédite correctement avec un taux de 77.38% tandis que la classe Femmes a été prédite correctement avec un taux de 86.23%, ce qui émet l'hypothèse que pour l'écriture en langue anglaise, l'écriture féminine est plus facilement prédite que l'écriture masculine.

4.6.2 CNN avec filtre de Gabor

Après avoir testé l'architecture du CNN proposée pour la prédiction du genre a partir d'image texture de l'écriture manuscrite, qui a montré des résultats prometteurs, nous passons a présents aux tests effectués avec une première couche convolutive des filtre de

Gabor présentés antérieurement.

Comme précédemment, on présente d'abord les résultats de KHATT (TBP, erreur, ROC) ensuite IAM.

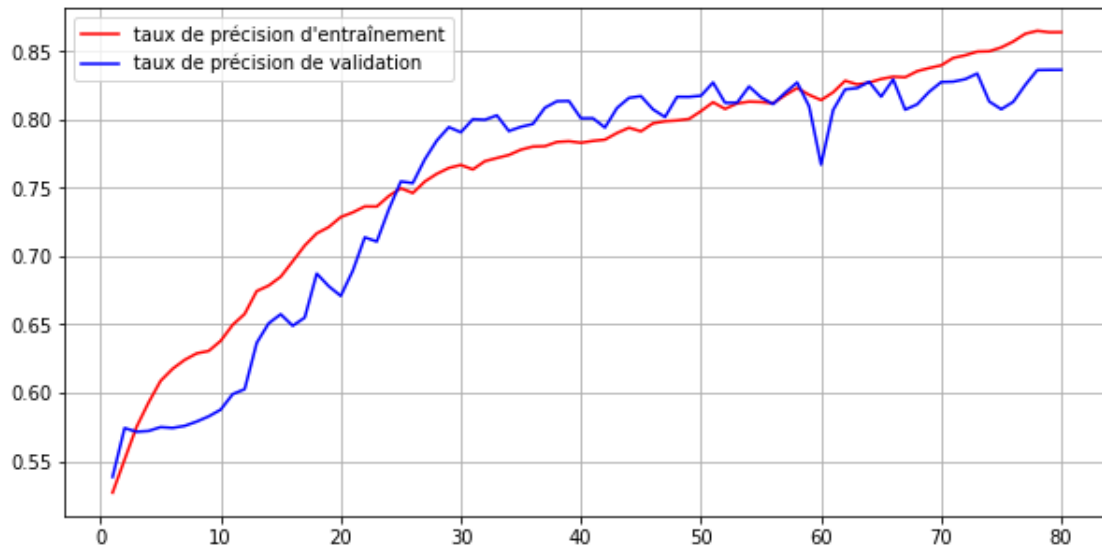


FIGURE 4.13: Taux de bonne prédiction d'entraînement et de validation pour la base KHATT (*100%)

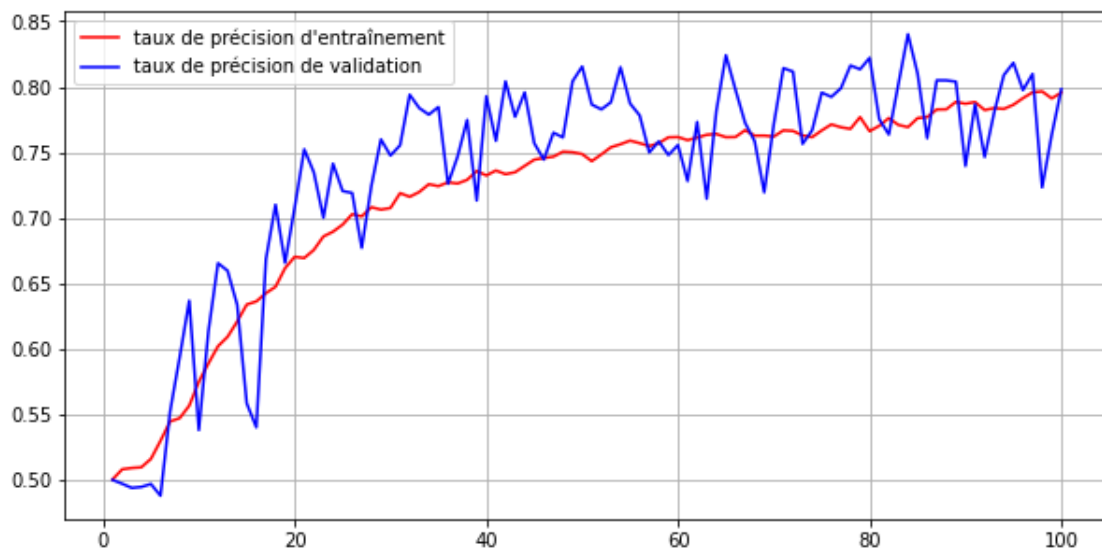


FIGURE 4.14: Taux de bonne prédiction d'entraînement et de validation pour la base IAM (*100%)

Le taux de bonne prédiction final atteint pour les données validation est de 83.66% pour la base KHATT et de 80.10% pour la base IAM. Nous observons que les graphes du taux de bonne prédiction appliqué aux données d'entraînement et validation se suivent et augmentent pour les deux bases de données. Nous remarquons et après avoir comparé les graphes de l'architecture CNN simple et la présente architecture que le graphe présente une meilleure allure avec moins d'oscillations.

Afin de s'assurer des performances du modèle, une évaluation doit être faite sur les données

test. Le TBP atteint pour les données test de la base KHATT est de 84.87%. Pour la base IAM, le TBP atteint pour les données test est de 82.13% : une valeur supérieure à celle trouvée avec l'architecture CNN précédente.

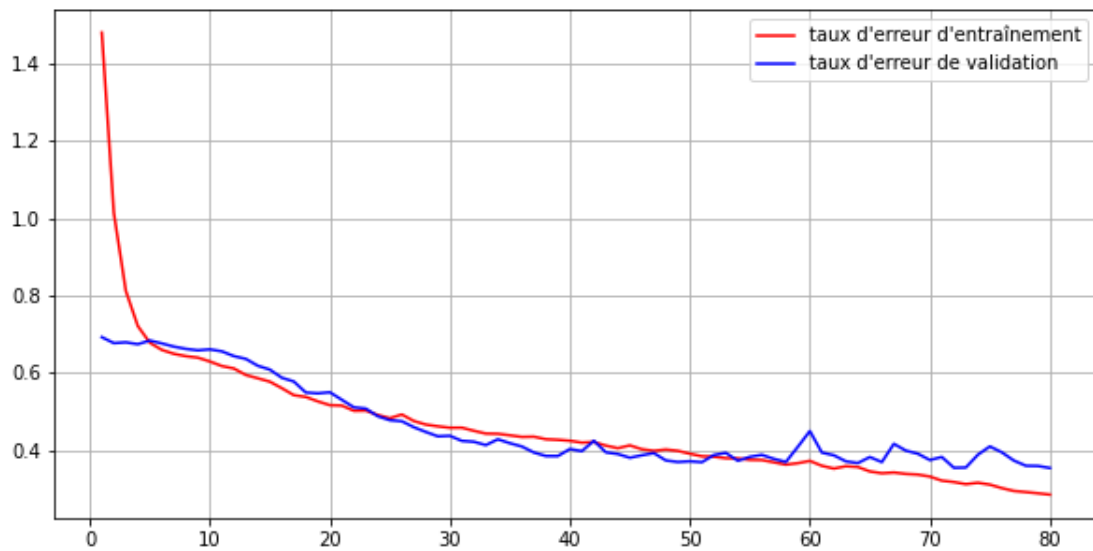


FIGURE 4.15: Erreur commise d'entraînement et de validation pour la base KHATT

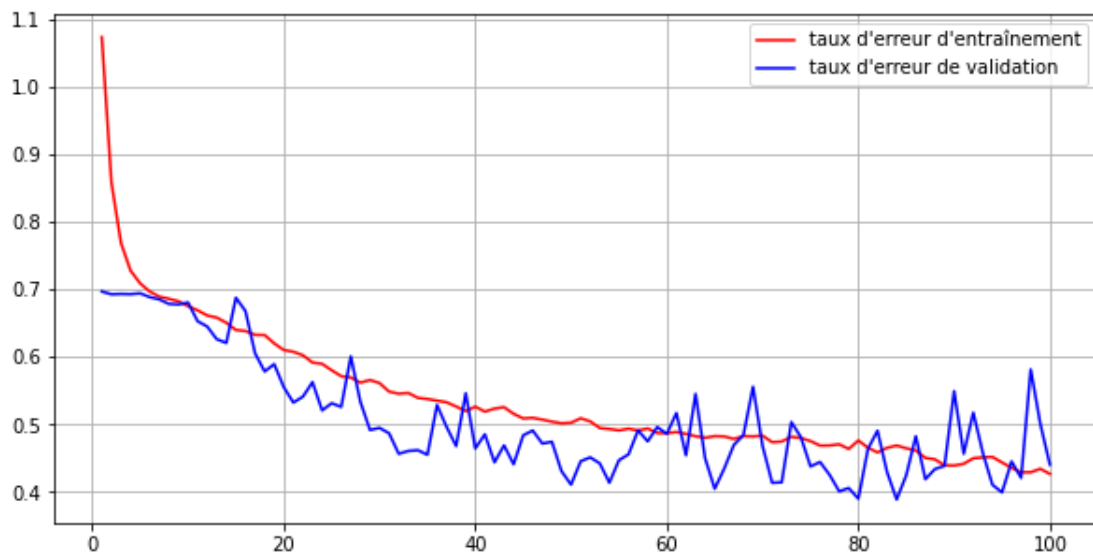


FIGURE 4.16: Erreur commise d'entraînement et de validation de pour base IAM

L'erreur finale atteinte pour les données validation de la base KHATT est de 0.350 tandis que pour la base IAM, l'erreur finale atteinte est de 0.448. Nous observons que les graphes de l'erreur commise pour les données d'entraînement et validation se suivent et diminuent, synonyme d'une bonne généralisation. Nous déduisons tout de même que ce modèle présente une meilleure allure avec moins d'oscillations.

Afin de s'assurer des performances du modèle, une évaluation doit être faite sur les données test qu'il n'a jamais vu. L'erreur atteinte pour les données test de la base KHATT est

de 0.387. Pour la base IAM, l'erreur atteinte est de 0.399 : une valeur inférieure à celle trouvée avec l'architecture CNN précédente.

La présente architecture atteint un AUC de 0.922 pour la base KHATT ce qui est tout de même un bon résultats et meilleur que les deux modèles précédents. Pour la base IAM, l'architecture a atteint un AUC de 0.907, une valeur tout aussi meilleure que celle atteinte par les deux architectures précédentes.

Nous traçons sur la figure 4.17 les trois courbes ROC des trois architectures proposées. Nous y voyons pour la base KHATT que les trois graphs sont pratiquement les mêmes, avec une légère avance sur le AUC pour l'architecture du CNN simple. Pour la base IAM, nous y voyons que l'architecture du CNN avec le filtre du Gabor a donné le meilleur résultat avec une plus grande surface sous la courbe "ROC" (AUC).

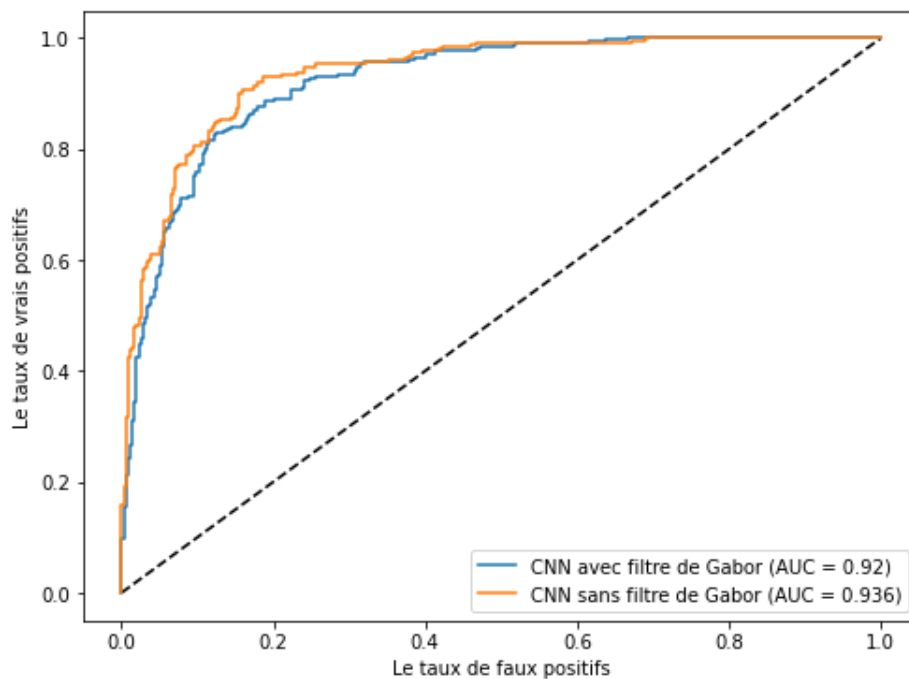


FIGURE 4.17: Comparaison des courbes ROC pour la base KHATT

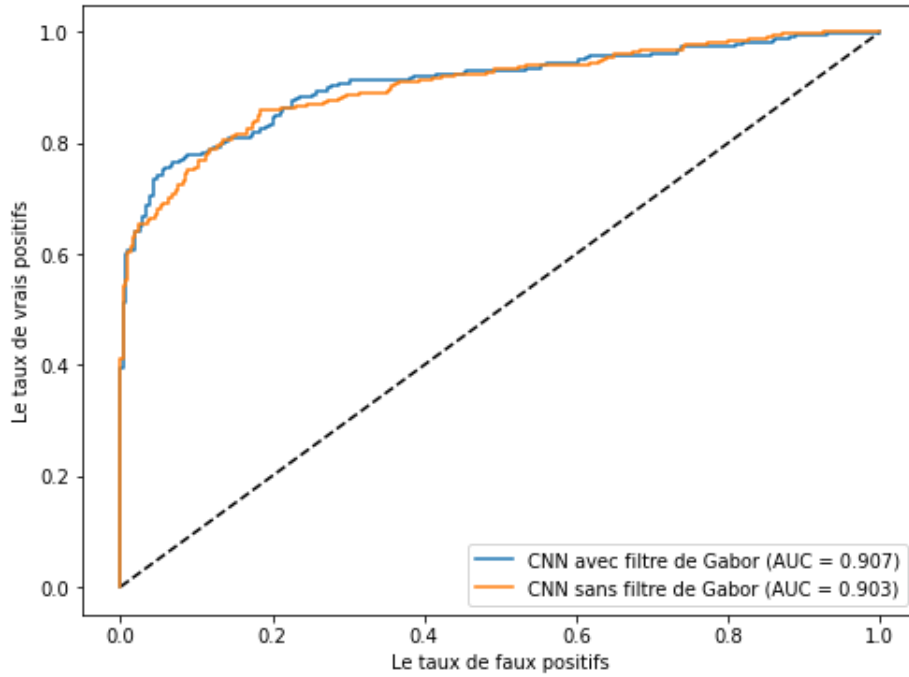


FIGURE 4.18: Comparaison des courbes ROC pour la base IAM

Nous donnons, tout comme précédemment, la matrice de confusion obtenue avec le présent modèle pour les deux bases de données dans les tableaux 4.8 et 4.9.

TABLE 4.8: Matrice de confusion pour la base KHATT (%)

		Classes réelles	
		Hommes	Femmes
Classes prédites	Hommes	87.85	18.24
	Femmes	12.15	81.76

TABLE 4.9: Matrice de confusion pour la base IAM (%)

		Classes réelles	
		Hommes	Femmes
Classes prédites	Hommes	77.38	13.11
	Femmes	22.62	86.89

Pour la base KHATT, la classe Hommes présente maintenant un TBP de 87.85% tandis que la classe Femmes présente un TBP de 81.76%. Nous déduisons alors que la classe Hommes a été plus réactive aux filtres de Gabor que la classe Femmes. Pour la base IAM, nous remarquons que les résultats pour la classe Hommes sont égales au résultats trouvé avec l'architecture CNN précédente, cependant, les résultats se sont améliorés pour la classe Femmes. Nous déduisons que pour la base IAM, la classe Femmes a été plus réactive aux filtres de Gabor. Nous résumons dans la figure 4.19 les résultats obtenus en terme de taux de bonnes prédictions pour les deux architectures : CNN et CNN avec Gabor.

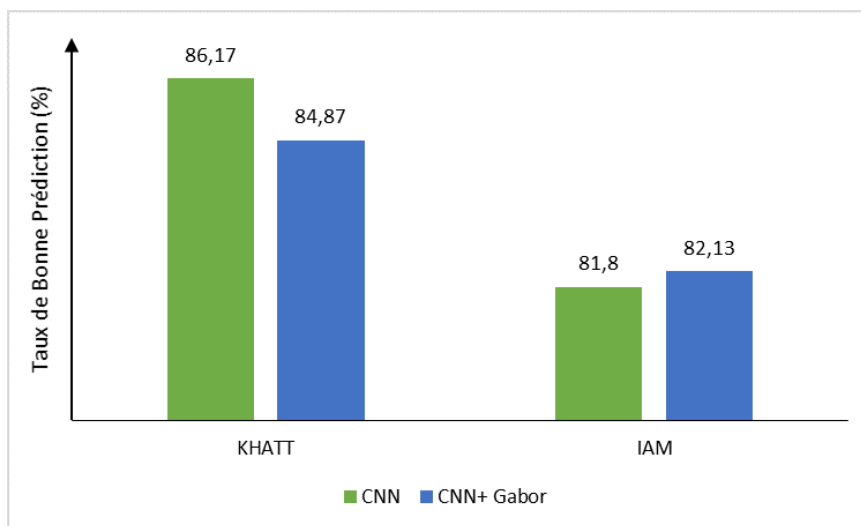


FIGURE 4.19: CNN vs CNN-Gabord pour la prédiction du genre

4.6.3 Comparaison avec les résultats de l'état de l'art

Dans cette sous-section, nous comparons les résultats de l'état de l'art trouvé pour la prédiction du genre à partir de l'écriture manuscrite pour les deux bases de données KHATT et IAM. Le tableau 4.10 résume les résultats obtenus.

TABLE 4.10: Comparaison des performances pour la prédiction du genre à partir de l'écriture manuscrite pour les deux base de données KHATT et IAM

	Bouadjenek et al., [7]	Morera et al., [38]	Kaljahi et al., [28]	Notre approche (CNN)	Notre approche (CNN+Gabor)
KHATT	82.22%	68.90%	77%	86.17%	84.87%
IAM	82.73%	80.72%	89.24%	81.80%	82.13%

Nous observons dans le tableau ci-dessus que les résultats que nous avons obtenus pour la base KHATT sont les plus élevés parmi les travaux cités. Les résultats obtenus avec l'architecture du CNN avec les filtres de Gabor ont dépassé les résultats publiés dans l'article de Kaljahi et al., [28], où ils ont utilisé des filtres de Gabor mais avec un algorithme conçu spécialement pour leur combinaison. La méthode que nous avons proposée semble être plus généraliste étant donné que la fusion des filtres générés se fait d'une manière automatique et s'apprend au fil des entraînements. Ceci assurera entre autres une application plus large pour d'autres bases de données. Pour la base IAM, nous avons obtenu tout de même de bons résultats mais pas les meilleurs, ceci peut être dû au fait que la base de données dont nous disposons ne contient pas beaucoup de données. Les résultats des deux bases de données, sont susceptibles à être améliorés d'avantage en explorant la combinaison entre le CNN, les filtres de Gabor et éventuellement d'autres descripteurs utilisés pour l'écriture manuscrite. Il est tout de même important de mentionner que l'étape de pré-traitement et qui consiste à générer des blocs textures a joué un grand

rôle pour arriver à ces résultats dans le sens où les caractéristiques relatives au genre ont nettement été mieux exposées.

4.7 Conclusion

A travers ce chapitre, il a été montré que la taille des images textures ainsi que l'espace inter-lignes influencent grandement sur les performances du modèle. Il a aussi été sujet d'expliquer l'influence des différents paramètres du filtre de Gabor et de montrer leurs effets sur les images textures. De plus, les résultats obtenus ont montré l'efficacité d'utiliser les images textures avec les CNNs en obtenant des résultats très satisfaisants. Les filtres de Gabor ont tout aussi montré leurs efficacité avec la base IAM en les combinant avec les CNNs, un système qui reste ouvert à plusieurs améliorations.

Conclusion

L'objectif du travail présenté dans ce mémoire est de prédire le genre à partir de l'écriture manuscrite. Cette application s'avère être très intéressante et utilisable dans plusieurs domaines tels que la criminologie, la recherche d'informations dans des documents historiques, etc. Notre approche se base sur deux éléments clé. Le premier élément clé se trouve dans la partie de pré-traitement des données où nous avons opté pour une méthode assez spéciale qui consiste en la génération des images textures en suivant un algorithme bien rigoureux. Le deuxième élément clé concerne l'utilisation des réseaux de neurones convolutifs ainsi que leur combinaison avec les filtres de Gabor, où ils ont montré leur efficacité avec la texture de l'écriture manuscrite. Nous résumons les principales conclusions dans ce qui suit :

- La génération de l'image texture, avec l'algorithme décrit, présente mieux les caractéristiques relatives au genre que l'image brute ;
- La constitution de la base de données d'entraînement est une étape très importante dans tout problème d'apprentissage automatique. Il est nécessaire que la base de données d'entraînement soit équilibrée et présente assez de diversité pour que le modèle apprenne au mieux les caractéristiques relatives au genre ;
- Il faut assurer un maximum d'indépendance entre les trois bases de données : Entraînement, validation et test afin d'éviter que les résultats soient biaisés. Ceci peut être fait en séparant les scripteurs présents dans chaque sous-base de données ;
- Deux éléments clés sont à la base de l'utilisation de l'image texture et qui influencent grandement les performances du modèles, à savoir, la taille de l'image texture et l'espace inter-ligne. D'après l'étude présentée, nous concluons que la taille optimale est de 300x300 pixels et l'espace inter-ligne optimal est de 0.60 ;
- L'utilisation des filtres de Gabor avec les images textures mettent en valeur des caractéristiques propres au genre tel que l'inclinaison, l'épaisseur et l'espacement ;
- La combinaison des filtres de Gabor avec les réseaux de neurones convolutifs a montré des résultats prometteurs et a largement diminué les oscillations ;
- Malgré le fait qu'on ait essayé d'améliorer les résultats en apportant des modifications à l'architecture CNN proposée tels que les filtres de Gabor, il a été trouvé que si les données sont distribués d'une manière correcte, le CNN simple présente les

meilleurs résultats et est une approche assez suffisante pour répondre aux besoins de notre problème.

Enfin, les résultats obtenus dans ce travail sont très prometteurs et peuvent être assujettis à plusieurs améliorations. Pour de prochains travaux, il serait intéressant d'entraîner les paramètres du filtre de Gabor tels que λ , σ , ψ , etc. Enfin, combiner le vecteur caractéristique généré par le *Flatten* de l'architecture citée avec d'autres descripteurs utilisés pour l'écriture manuscrite peut donner de meilleurs résultats.

Bibliographie

- [1] Y. AKBARI et al. « Wavelet-based gender detection on off-line handwritten documents using probabilistic finite state automata ». In : *Image and Vision Computing* 59 (2017).
- [2] S. AL-MAADEED et A. HASSAINE. « Automatic prediction of age, gender, and nationality in offline handwriting ». In : *Eurasip Journal on Image and Video Processing* 2014.10 (2014).
- [3] Panagiotis BARMPOUTISA et al. « Wood species recognition through multidimensional texture analysis ». In : *Science Direct* (2017).
- [4] J. R. BEECH et I. C. MACKINTOSH. « Do differences in sex hormones affect handwriting style? evidence from digit ratio and sex role identity as determinants of the sex of handwriting? » In : *Personality and Individual Differences* 39 (2005), p. 459-468.
- [5] D. BERTOLINI et al. « Texture-based descriptors for writer identification and verification ». In : *Expert Systems with Applications* (2012).
- [6] Alfred BINET. « Can intelligence be measured scientifically by graphology? » In : (1903).
- [7] Nesrine BOUADJENEK, Hassiba NEMMOUR et Youcef CHIBANI. « Robust soft-biometrics prediction from off-line handwriting analysis ». In : *Science Direct* (2015).
- [8] Nesrine BOUADJENEK, Hassiba NEMMOUR et Youcef CHIBANI. « Writer's gender classification using HOG and LBP features ». In : *Lecture Notes in Electrical Engineering* 411 (2017).
- [9] Samia BOUTELDJA et Assia KOURGLI. « Multiscale texture features for the retrieval of high resolution satellite images ». In : *IEEE* (2015).
- [10] J. CHAPRAN. « Biometric writer identification : Feature analysis and classification ». In : *International Journal of Pattern Recognition and Artificial Intelligence* 20.4 (2006), p. 483-503.
- [11] François CHOLLET. In : *Deep Learning with Python*. <https://www.manning.com/books/deep-learning-with-python> (visité le 20/04/2020). Shelter Island, NY 11964 : Manning, 2017. Chap. Deep learning for computer vision, p. 119-177.

- [12] *Classification : ROC et AUC*. (visité le 05/07/2020). URL : <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=fr>.
- [13] Marc Van DROOGENBROECK. *Filtres de Gabor*. (visité le 10/06/2020). URL : http://www.telecom.ulg.ac.be/teaching/notes/totali/elen016/node46_mn.html.
- [14] M. ERBILEK, M. FAIRHURST et C. LI. « Exploring gender prediction from digital handwriting ». In : *24th Signal Processing and Communication Application Conference* (May 2016).
- [15] S. FEFFER. *It's all about the features*. (visité le 25/02/2020). URL : <https://reality.ai/it-is-all-about-the-features>.
- [16] Kunihiko FUKUSHIMA. « Neocognitron : A hierarchical neural network capable of visual pattern recognition ». In : *Science Direct* (1988).
- [17] A. GATTAL et al. « Gender classification from offline multi-script handwriting images using oriented Basic Image Features (oBIFs) ». In : *Science Direct* (2018).
- [18] C. GEADYA et al. « Bridging the gap between micro- and macro-scales in medical imaging with textural analysis – A biological basis for CT radiomics classifiers? » In : *Science Direct* (2020).
- [19] M. GENNA et A. ACCARDO. « Gender and age influence in handwriting performance in children and adolescents ». In : *5th European Conference of the International Federation for Medical and Biological Engineering (IFMBE)* (2012).
- [20] R. K. HANUSIAK et al. « Writer verification using texture-based features ». In : *International Journal on Document Analysis and Recognition (IJ DAR)* 2012.15 (2011), p. 213-226.
- [21] J. HARTLEY. « Sex Differences in Handwriting : a comment on Spear ». In : *British Educational Research Journal* 17.2 (1991).
- [22] William N. HAYES. « Identifying sex from handwriting ». In : *Perceptual and Motor Skills* 83 (1996).
- [23] Guang-Bin HUANG, Qin-Yu ZHU et Chee-Kheong SIEW. « Extreme learning machine : Theory and applications ». In : *Science Direct* (2006).
- [24] D. H. HUBEL et T. N. WIESEL. « Receptive fields, binocular interaction and functional architecture in the cat's visual cortex ». In : *Signal Processing* (1962).
- [25] D. H. HUBEL et T. N. WIESEL. « Receptive fields of single neurones in the cat's striate cortex ». In : *The journal of Physiology* (1959).
- [26] Evyatar ILLOUZ, Eli DAVID et Nathan S. NETANYAHU. « Handwriting-Based Gender Classification Using End-to-End Deep Neural Networks ». In : *International Conference on Artificial Neural Networks* (2018).

- [27] Javed IQBAL et Mohsen ALI. « Weakly-supervised domain adaptation for built-up region segmentation in aerial and satellite imagery ». In : *Science Direct* (2020).
- [28] Maryam Asadzadeh KALJABI, P. V. Vidya VARSHINI et Palaiahnakote SHIVAKUMARA. « Word-Wise Handwriting Based Gender Identification Using Multi-Gabor Response Fusion ». In : *Communications in Computer and Information Science* (2019).
- [29] Diederik P. KINGMA et Jimmy BA. « Adam : A Method for Stochastic Optimization ». In : *The 3rd International Conference for Learning Representations* (2017).
- [30] Alex KRIZHEVSKY, Ilya SUTSKEVER et Geoffrey E. HINTON. « Imagenet classification with deep convolutional neural networks ». In : *NIPS Proceedings* (2012).
- [31] Y. LECUN et al. « Gradient-based learning applied to document recognition ». In : *IEEE* 86.11 (1998).
- [32] Wenbo LIA et al. « A medical image retrieval method based on texture block coding tree ». In : *Science Direct* (2017).
- [33] M. LIWICKI et H. BUNKE. « IAM-OnDB - an On-Line English Sentence Database Acquired from Handwritten Text on a Whiteboard ». In : *Document Analysis and Recognition 2* (2005).
- [34] M. LIWICKI, A. SCHLAPBACH et H. BUNKE. « Automatic gender detection using on-line and off-line information ». In : *Pattern Analysis and Applications (PAA)* 14.1 (2011).
- [35] M. LIWICKI et al. « Automatic detection of gender and handedness from on-line Hand-writing ». In : *Proceedings of the 13th Conf. Intl Graphonomics Society* (2007).
- [36] S. MAHMOUD et Mohammad ALSHAYEB. « KHATT : An open Arabic offline handwritten text database ». In : *Pattern Recognition* 47.3 (2014).
- [37] U. MARTI et H. BUNKE. « A full English sentence database for off-line handwriting recognition ». In : *In Proc. of the 5th Int. Conf. on Document Analysis and Recognition* (1999).
- [38] Ángel MORERA et al. « Gender and Handedness Prediction from Offline Handwriting Using Convolutional Neural Networks ». In : *Hindawi* (2018).
- [39] N. OTSU. « A Threshold Selection Method from Gray-Level Histograms ». In : *IEEE Transactions on Systems, Man, and Cybernetics* (1979).
- [40] R. PLAMONDON et S. N. SRIHARI. « On-line and off-line handwriting recognition : A comprehensive survey ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.1 (2000), p. 63-84.
- [41] R. PRABHU. *Understanding of convolutional neural network CNN deep learning*. (visité le 04/08/2020). URL : <https://medium.com/@RaghavPrabhu/understanding-of-convolutionalneural-network-cnn-deep-learning-99760835f148>.

- [42] PUNEET et Naresh GARG. « Binarization Techniques used for Grey Scale Images ». In : *International Journal of Computer Applications* (2013).
- [43] Sumit SAHA. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. (visité le 07/04/2020). URL : <https://towardsdatascience.com/a-comprehensiveguide-toconvolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
- [44] J. SERRA. « Image analysis and mathematical morphology ». In : *Academic Press, inc., London* (1982).
- [45] I. SIDDIQI et al. « Automatic analysis of handwriting for gender classification ». In : *Pattern Analysis and Applications (PAA)* 18.4 (2015).
- [46] Fabio SPANHOL et al. « Breast Cancer Histopathological Image Classification using Convolutional Neural Networks ». In : *International Joint Conference on Neural Networks (IJCNN)* (2016).
- [47] M. G. SPEAR. « Differences between the written work of boys and girls ». In : *British Educational Research Journal* 15 (1989), p. 271-277.
- [48] Nitish SRIVASTAVA, Geoffrey HINTON et Alex KRIZHEVSKY. « Dropout : A Simple Way to Prevent Neural Networks from Overfitting ». In : *Journal of Machine Learning* (2014).
- [49] Giovanni TARASCHI et Joao B. FLORIND. « Computing fractal descriptors of texture images using sliding boxes : An application to the identification of Brazilian plant species ». In : *Science Direct* (2019).
- [50] Bing XU, Naiyan WANG et Tianqi CHEN. « Empirical Evaluation of Rectified Activations in Convolutional Network ». In : *Journal of Machine Learning* (2015).