UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE
APPLIQUÉES

PAR
ABDALLAH BENKADJA

HYBRIDE CNN-SVM :
VERS UNE MEILLEURE CLASSIFICATION

AVRIL 2023

Table des matières

Table des figures **iii**

Liste des tableaux **iv**

Résumé **i**

Abstract **ii**

Remerciement **iii**

1 Chapitre 1 Introduction **1**
1.1 Mise en contexte 1
1.2 Objectifs du mémoire 5

2 Chapitre 2 Support Vector Machine (SVM) **6**
2.1 La marge SVM 9
2.2 Soft Margin SVM 11
2.3 Dualité de Lagrange 13
2.4 Les noyaux 15

3 Chapitre 3 Réseaux de neurones à convolution (CNN) **17**
3.1 Les éléments constitutifs de CNN 17
3.1.1 Couche de convolution 18
3.1.2 Couche de sous-échantillonnage (Pooling layer) 20
3.1.3 La fonction d'activation 21
3.1.4 Normalisation de lot (Batch normalization) 23
3.1.5 Dropout 24
3.1.6 Fully connected layer 25
3.2 Les architectures CNN 25

4 Chapitre 4 Hybride CNN-SVM **28**

5 Chapitre 5 Reconnsaissance du manuscrit **31**
5.1 Les caractères latins 31
5.2 Les caractères arabes 33
5.2.1 Les lettres arabes 33
5.2.2 Les chiffres arabes orientaux 35

6 Chapitre 6 Méthodologie **38**
6.1 Datasets 38
6.1.1 EMNIST 39
6.1.2 AHCD 40
6.1.3 MADBase 42
6.2 Approche proposée 45

6.2.1 SVM 45
6.2.2 CNN 46
6.2.3 Hybride CNN-SVM 48
6.3 Conclusion 50

7 Chapitre 7 Expérimentations et discussion 51
7.1 Caractères latins : EMNIST 52
7.2 Caractères arabes 55
7.2.1 Lettres arabes (AHCD) 55
7.2.2 Chiffres arabes (MADBase) 57
7.3 Synthèse 58

8 Chapitre 8 Conclusion et perspectives 62

Références 64

ANNEXE A : Hybride CNN-SVM : vers une meilleure classification des données manuscrites 69

ANNEXE B : Statistical profiling of Hybride CNN-SVM effectiveness 78

Table des figures

1 Projection sur l'hyperplan séparatif 7
2 L'hyperplan séparatif dans $\mathbb{R}^3$ 7
3 Infinité de solutions possibles pour l'hyperplan séparatif. 9
4 La marge dans SVM. 10
5 Soft Margin SVM. 12
6 L'opération de la convolution[9]. 20
7 L'opération du sous-échantillonnage[9]. 21
8 Les fonctions d'activation. 23
9 Dropout dans un réseau de neurones artificiels[42]. 24
10 L'avantage empirique du Dropout[42]. 25
11 Couche *Fully connected*. 26
12 Les différentes architectures développées dans la littérature[30]. 27
13 L'architecture Trainable Feature Extractor (TFE)[31]. 30
14 Les datasets constitutifs de EMNIST[16]. 41
15 Échantillon des images de EMNIST Balanced. 42
16 Échantillon des images de EMNIST Letters. 43
18 Le système de numération indo-arabe en Moyen-Orient[10]. 43
17 Échantillon des images de AHCD. 44
19 Échantillon des images de MADBase. 45
20 Le modèle Hybride CNN-SVM. 48

Liste des tableaux

1 Résultats de la littérature pour les différents datasets de EMNIST 32
2 Les datasets utilisés. 39
3 Architectures CNN utilisées 47
4 Les taux de reconnaissance en (%) par SVM et CNN sur les bases de test. . 52
5 Les taux de reconnaissance en (%) par Hybride CNN-SVM sur les bases de test. 53
6 Les seuils observés relativement aux hypothèses étudiées. 54
7 Tableau comparatif des résultats pour EMNIST 59
8 Tableau comparatif des résultats pour les caractères arabes 59

Résumé

L'objectif de ce mémoire est de présenter une nouvelle approche de classification et de l'évaluer empiriquement dans le contexte de la reconnaissance automatique du manuscrit. Notre approche consiste en un modèle hybride appelé "Hybride CNN-SVM", qui combine un réseau de neurones à convolution (CNN) avec un classifieur de Support Vector Machine (SVM). L'architecture du modèle implique le remplacement de la dernière couche du CNN par le SVM. Cela permet une flexibilité considérable dans le choix des paramètres pour le CNN et le SVM.

À travers nos expérimentations, nous avons comparé les performances de notre approche hybride avec celles obtenues en utilisant individuellement le CNN et le SVM. Cela nous a permis de fournir une analyse comparative approfondie des résultats obtenus par notre approche hybride par rapport à ces deux méthodes utilisées séparément Le modèle proposé est abordé selon deux perspectives : sa capacité à reconnaitre les caractères manuscrits et celle à maintenir sa performance en termes de taux de reconnaissance par rapport à CNN et à SVM.

Nos expérimentations ont été menées sur des jeux de données de bien connus et couramment utilisés dans le domaine de la reconnaissance automatique des caractères manuscrits. Nos sources de données sont des images manuscrites composées de chiffres et de lettres pour les deux types d'écritures : latin et arabe. L'extension de l'application de notre approche sur des datasets aussi variés a permis de confirmer empiriquement que l'approche Hybride CNN-SVM est une méthode prometteuse pour la classification en général et la reconnaissance de l'écriture manuscrite en particulier.

Mots clés : Classification, CNN, SVM, Hybride CNN-SVM.

Abstract

The objective of this thesis is to present a new classification approach and empirically evaluate it in the context of automatic handwriting recognition. Our approach consists of a hybrid model called "Hybrid CNN-SVM", which combines a convolutional neural network (CNN) with a support vector machine (SVM) classifier. The architecture of the model involves replacing the last layer of the CNN with the SVM, allowing for considerable flexibility in choosing parameters for both the CNN and SVM.

Through our experiments, we compared the performance of our hybrid approach with that of using CNN and SVM individually. This allowed us to provide a comprehensive comparative analysis of the results obtained by our hybrid approach compared to these two methods used separately. The proposed model is evaluated from two perspectives : its ability to recognize handwritten characters and its ability to maintain performance in terms of recognition rates compared to CNN and SVM.

Our experiments were conducted on well-known and commonly used datasets in the field of automatic handwriting recognition. Our data sources consist of handwritten images composed of digits and letters for both Latin and Arabic scripts. The extension of our approach to such diverse datasets has empirically confirmed that the Hybrid CNN-SVM approach is a promising method for classification in general, and handwriting recognition in particular.

Keywords : Classification, CNN, SVM, Hybride CNN-SVM.

Remerciements

Je tiens à exprimer ma profonde gratitude envers mes deux professeurs, Ismail Biskri et Nadia Ghazzali, pour leur inestimable contribution à mon travail de recherche. Leurs conseils éclairés, leur patience et leur encouragement constant m'ont inspiré à donner le meilleur de moi-même et à persévérer malgré les défis rencontrés. Je suis également reconnaissant pour leur dévouement à l'excellence académique et leur passion pour l'enseignement. Leurs commentaires constructifs ont grandement enrichi mon travail et m'ont permis de développer mes compétences en recherche. Je suis honoré d'avoir eu l'opportunité de travailler avec eux et je suis reconnaissant pour leur mentorat précieux. Merci du fond du cœur à Ismail Biskri et Nadia Ghazzali pour leur soutien et leur contribution inestimable à mon mémoire.

Je tiens également à exprimer ma profonde gratitude envers ma famille pour leur soutien inconditionnel tout au long de mon parcours académique. Leur amour, leur encouragement et leur confiance en moi ont été une source inestimable de motivation et de soutien. Je suis reconnaissant pour leur présence constante et leur encouragement à poursuivre mes objectifs académiques. Leur soutien indéfectible a été un pilier dans ma réussite et je leur suis profondément reconnaissant pour tout leur amour et leur soutien. Merci du fond du cœur à ma famille pour leur rôle essentiel dans mon parcours académique.

J'exprime ma gratitude envers tous ceux qui ont contribué de près ou de loin à la réalisation de mon mémoire.

Chapitre 1 Introduction

1.1 Mise en contexte

Au cours des dernières décennies, l'apprentissage automatique (Machine Learning) a suscité de grandes préoccupations auprès de la communauté des chercheurs ; la liste des domaines touchés s'étend rapidement. L'essor connu de ses applications fut possible grâce d'une part au progrès qu'a connu la puissance calculatoire des machines ces dernières décennies et d'autre part au développement qu'a connu la digitalisation de l'information et l'apparition de bases de données importantes en volume. L'apprentissage automatique est au cœur du développement des outils visant à exploiter les données afin d'extraire des paramètres permettant d'identifier des modèles de prédiction et de classification.

Le domaine de l'apprentissage automatique a été largement pratiqué dans le domaine de la reconnaissance des formes et la vision par ordinateur ; or, de nombreuses applications ont tiré avantage de son développement : la reconnaissance faciale, la reconnaissance d'images, la reconnaissance d'empreintes digitales, la reconnaissance de caractères manuscrits, etc. Les techniques basées sur l'apprentissage automatique ont suscité un intérêt majeur dans la communauté de recherche pour résoudre une variété de problèmes d'apprentissage supervisé et non supervisé.

Parmi les approches les plus utilisées dans le domaine de l'apprentissage automatique, nous avons les réseaux de neurones à convolution (Convolutional Neural Network) CNN [33] et les Support Vector Machine (SVM) [17]. Les réseaux CNN sont une catégorie de réseaux de neurones capables d'extraire automatiquement les caractéristiques pertinentes des données d'entrée via l'opération de la convolution. SVM est une technique d'apprentissage supervisé pour des fins de classification, largement utilisée dans la recherche d'information, la vision par ordinateur, la biologie, etc. Ces approches d'apprentissage automatique ont démontré leurs fiabilités et leurs performances dans un large domaine d'applications.

Malgré leur efficacité inhérente démontrée dans le domaine de la reconnaissance des formes, plus particulièrement, la reconnaissance des chiffres et des lettres manuscrits, on constate, plus récemment, l'émergence d'une approche hybride combinant CNN et SVM dans laquelle la dernière couche de CNN est remplacée par le classifieur SVM. Les travaux de recherche ont montré que cette approche hybride s'est avérée efficace dans divers domaines, notamment la classification d'images IRM et la détection de tumeurs cérébrales, [29], la classification des chiffres manuscrits [1, 4, 37]. Ahlawat et Choudhary [1] et Niu et Suen [37] ont appliqué une approche hybride combinant CNN et SVM dans le domaine de la reconnaissance automatique des chiffres manuscrits de MNIST [1]. Ils rapportent que l'approche utilisée donne des taux de reconnaissance respectifs de 98,88 et 99,81, soit une augmentation respectivement de 1,03 % (98,88 % - 97,85 %) par rapport à SVM seul et de 0,4 % (99,81 % - 99,41 %) par rapport à CNN seul. Les résultats des travaux de recherche de Ahlawat et Choudhary [1] et Niu et Suen [37] suggèrent que l'approche Hybride CNN-SVM peut améliorer le taux de reconnaissance des chiffres manuscrits dans le contexte du jeu de données MNIST. Cependant, il n'est pas clair si cette approche offre un avantage significatif par rapport à l'utilisation individuelle des modèles CNN et SVM lorsqu'elle est appliquée à des ensembles de données plus hétérogènes. La plupart des études publiées sur l'approche Hybride CNN-SVM se limitent à des expérimentations sur des ensembles de données spécifiques utilisés en amont. Par conséquent, la synthèse des résultats de la littérature pour tirer des conclusions définitives sur l'efficacité et l'avantage de cette approche par rapport à CNN et SVM s'avère difficile en raison de l'hétérogénéité des ensembles de données utilisés dans la littérature ainsi que des nombreux paramètres des approches sous-jacentes CNN et SVM.

Les avantages révélés par certains travaux quant à l'application de l'approche hybride, constitue le point d'encrage de notre étude exploratrice de son éventuel apport empirique comparativement à CNN et SVM dans le domaine du manuscrit. Donc, nous visons à évaluer les performances de l'approche Hybrid CNN-SVN et sa capacité à maintenir sa performance en termes de taux de reconnaissance par rapport à CNN et à SVM à travers une application dans le domaine de la reconnaissance automatique des caractères manuscrits.

1. http ://yann.lecun.com/exdb/mnist/

La reconnaissance du manuscrit est un domaine de pointe où l'intelligence artificielle trouve une application significative. Elle revêt un intérêt majeur dans le domaine du traitement d'image et de la reconnaissance des formes, et est utilisée dans de nombreuses applications telles que la vérification de chèques, la bureautique, les affaires, la lecture d'adresses postales, et bien d'autres. Une amélioration constante du taux de reconnaissance est toujours recherchée dans la recherche en apprentissage automatique ainsi que dans les applications concrètes de l'industrie.

Le domaine de la reconnaissance du manuscrit a été largement exploré par les chercheurs depuis plusieurs décennies, avec l'utilisation de divers algorithmes tels que SVM, Multi-Layer Perceptron (MLP), Hidden Markov Model (HMM), Deep Networks (DNN), Recurrent Neural Networks (RNN), CNN, etc. Il reste un sujet de recherche actif, à la fois dans son application et dans le développement de modèles de classification et d'apprentissage automatique. Parallèlement, la reconnaissance du manuscrit s'avère être également un excellent banc d'essai pour les algorithmes d'apprentissage, de nombreux travaux utilisant cette tâche comme un test de première ligne pour évaluer les performances des approches proposées. Cela souligne l'importance de ce domaine en tant que défi et référence pour évaluer l'efficacité des algorithmes d'apprentissage automatique dans un contexte concret. [12, 17, 28, 32, 35]. L'application de l'apprentissage automatique dans le domaine de la reconnaissance du manuscrit offre ainsi un double avantage. Tout d'abord, elle permet d'évaluer les performances de notre approche hybride, en testant son efficacité sur cette tâche spécifique. En utilisant la reconnaissance du manuscrit comme cas d'étude, nous pouvons mesurer les résultats de notre approche et évaluer sa pertinence dans ce contexte. En outre, l'application de l'apprentissage automatique dans la reconnaissance du manuscrit permet également d'identifier les perspectives d'amélioration potentielles des taux de reconnaissance des caractères manuscrits étudiés. Ainsi, l'application de l'apprentissage automatique dans la reconnaissance du manuscrit offre une opportunité d'évaluation de notre approche et d'identification des domaines d'amélioration potentiels, ce qui contribue à l'avancement de la recherche dans ce domaine et à l'amélioration continue des performances des systèmes de reconnaissance de caractères manuscrits.

Le fonctionnement d'un système d'apprentissage automatique nécessite des bases de données importantes en volume. L'importance de bons datasets référentiels est critique ; ils fournissent un moyen rapide, quantitatif et équitable d'analyser différentes approches et techniques d'apprentissage automatique, permettant ainsi aux chercheurs d'avoir rapidement un aperçu des performances de leurs approches proposées. Or, la disponibilité des datasets référentiels dans le domaine du manuscrit a contribué au développement qu'a connu la recherche dans le domaine. Par exemple, la base MNIST est devenue une source de données standard dans l'étude de la reconnaissance automatique des chiffres manuscrits. Elle a été exploitée depuis son apparition par un grand nombre de travaux. Cependant, un seul ensemble de données ne peut couvrir qu'un objectif spécifique. L'existence d'une suite variée de datasets permet une approche plus holistique dans l'évaluation des performances d'un algorithme ou d'un système. Dans l'optique d'atteindre nos objectifs de recherche, on mène nos expérimentations sur des datasets variés et importants en volume.

Dans le domaine du manuscrit, de nombreuses méthodes ont été proposées et testées sur les chiffres manuscrits latins de MNIST et rapportent des taux de reconnaissance très élevés. Cependant, le progrès de la recherche est moins prononcé en ce qui concerne la reconnaissance automatique du manuscrit autres que latin. Pour les besoins de notre recherche, on étendra l'application de notre approche impliquant CNN, SVM, et leur combinaison à des datasets des images manuscrites composées de chiffres et de lettres pour les deux types d'écritures : latin et arabe. Cela nous procurera un avantage supplémentaire quant à l'évaluation de notre approche sur des données manuscrites autres que latines afin de pouvoir accumuler plus d'éléments probants quant à la pertinence de notre approche. Jusqu'à récemment, et à notre connaissance, aucun chercheur n'a encore appliqué les approches CNN, SVM ainsi que l'approche combinant CNN et SVM à l'écriture manuscrite au moyen des datsets aussi importants et aussi variés que l'on utilise dans le travail en cours.

Les conclusions tirées dans le cadre de cette étude ont été publiées dans un article de conférence, qui est annexé à ce mémoire (Annexe A). Nous sommes également fiers d'annoncer que cet article a été sélectionné pour une publication plus étendue dans le volume de post-proceedings intitulé *New Frontiers in Textual Data Analysis (M. Misuraca & G. Giordano,*

editors) et publié par Springer Nature dans la série *Springer Nature* de la série de *Studies in Classification, Data Analysis, and Knowledge Organization.* [2]

1.2 Objectifs du mémoire

L'objectif de ce mémoire est d'explorer une nouvelle approche combinant les techniques de CNN et SVM afin d'améliorer le taux de reconnaissance de l'écriture manuscrite. Pour cela, nous avons adopté une approche en deux étapes pour étudier l'approche Hybride CNN-SVM.

Dans la première étape, nous avons examiné la reproductibilité et la fiabilité des résultats de l'approche hybride par rapport aux approches individuelles CNN et SVM. Nous avons également évalué l'amélioration de la précision de la reconnaissance automatique des caractères manuscrits en prenant en compte un large éventail de paramètres pour les deux techniques.

Dans la deuxième étape, nous avons comparé les performances de notre approche avec les résultats récents dans le domaine pour chaque dataset étudié. Nous avons utilisé des jeux de données variés en termes de nature des données, nombre de classes et taille d'échantillon, comprenant des chiffres et des lettres pour deux types d'écritures : latin et arabe.

Après cette introduction, le mémoire est divisé en six chapitres avant de conclure. Le chapitre 2 sera consacré à l'approche SVM, suivi par le chapitre 3 sur CNN, puis le chapitre 4 sur l'Hybride CNN-SVM. Le chapitre 5 abordera la reconnaissance de l'écriture manuscrite, suivi par le chapitre 6 sur la méthodologie. Enfin, le chapitre 7 traitera les expérimentations et les discussions.

2. https ://www.springer.com/series/1564

Chapitre 2 Support Vector Machine (SVM)

L'approche SVM est issue originalement des travaux de Cortes et Vapnik [17]. Elle consiste à résoudre un problème de classification binaire. La formulation mathématique consiste à identifier la fonction de décision : $f : \mathbb{R}^p \to \{+1, -1\}$.

La valeur d'entrée consiste en un vecteur de données x_i avec p dimensions. La valeur de sortie est binaire, soit $+1$ pour une appartenance à une classe spécifique, soit -1 pour une non-appartenance, en raison de la nature d'une classification binaire. La signification de ces classes peut varier en fonction du problème étudié, par exemple : {True, False}, {tumeur maligne, tumeur bénigne}, etc.

Puisque la SVM est une solution à un problème de classification binaire, on peut considérer le dataset d'entrainement comme étant un ensemble de vecteurs $x_i \in \mathbb{R}^p$ conjointement avec leurs valeurs de classe $y_i \in \{+1, -1\}$. Ainsi, le dataset d'entrainement correspond à l'ensemble $\{(x_1, y_1), ..., (x_n, y_n)\}$.

L'objectif de la SVM est de déterminer les paramètres de l'hyperplan séparant les données des deux classes ($+1$ et -1). Par exemple, dans un espace bidimensionnel $\mathbb{R}^2$, comme illustré dans la Figure 1, l'hyperplan est une ligne qui sépare les deux groupes d'échantillons. Dans un espace tridimensionnel $\mathbb{R}^3$, l'hyperplan est un plan en deux dimensions, comme illustré dans la Figure 2. Plus généralement, dans un espace de dimension p ($\mathbb{R}^p$), l'hyperplan est un sous-espace affine de dimension $p - 1$ qui sépare les deux groupes de données que l'on cherche à discriminer.

Une représentation visuelle de l'hyperplan séparateur est présentée dans les Figures 1 et 2, où le vecteur $w \in \mathbb{R}^p$ représente la normale de l'hyperplan et b est l'intercept. Les paramètres recherchés pour définir l'hyperplan sont donc $w \in \mathbb{R}^p$ et $b \in \mathbb{R}$.

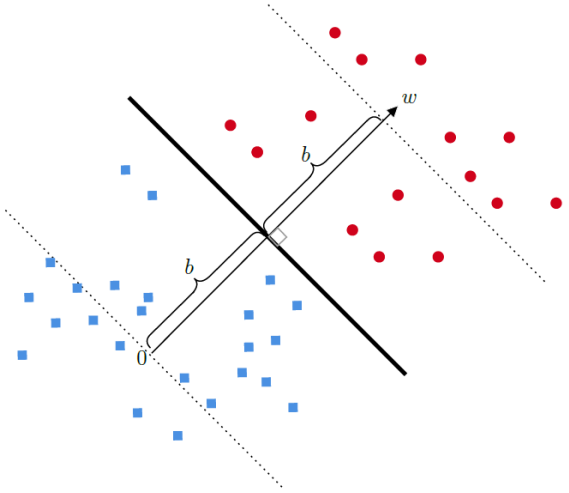


FIGURE 1 – Projection sur l'hyperplan séparatif

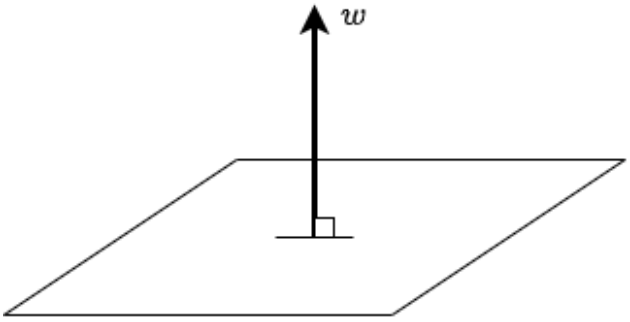


FIGURE 2 – L'hyperplan séparatif dans $\mathbb{R}^3$

Le produit scalaire $\langle x_i, x_j \rangle$ est une mesure de la similitude entre les vecteurs x_i et x_j, largement utilisée dans des concepts clés de la géométrie analytique tels que la projection et l'orthogonalité. Dans le cadre de la classification par SVM, le produit scalaire est essentiel dans la fonction de décision $f(x)$, qui évalue la similitude entre un vecteur de données $x \in \mathbb{R}^p$ et le vecteur normal w de l'hyperplan séparateur, décalé par l'intercept b.

$$\begin{aligned} f : \mathbb{R}^p &\longrightarrow \mathbb{R} \\ x &\longmapsto \langle w, x \rangle + b \end{aligned}$$

En d'autres termes, si la valeur de la fonction de décision $f(x)$ est positivement grande, cela signifie que le vecteur x est dans la direction du vecteur normal w de l'hyperplan, et donc du côté de la classe $y = +1$. En revanche, si la valeur de $f(x)$ est négativement grande, cela

indique que le vecteur x est dans la direction opposée à celle de w, et donc du côté de la classe $y = -1$. Lorsque $f(x)$ est égal à 0, cela signifie que le vecteur x se trouve sur la zone limite de l'hyperplan, qui est la frontière entre les deux classes.

La frontière entre les deux classes est établie par le paramètre b, qui détermine le décalage de l'hyperplan par rapport à l'origine. L'objectif de la classification par SVM est donc de trouver les valeurs optimales des paramètres w et b pour définir l'hyperplan qui sépare efficacement les deux classes dans l'espace des données.

Les vecteurs de données x_i qui appartiennent à la classe $y_i = +1$ doivent respecter l'équation (1) suivante :

$$\langle w, x_i \rangle + b \geq 0 \text{ pour tout } y_i = +1 \tag{1}$$

D'un autre côté, les vecteurs de données appartenant à la classe $y_i = -1$ doivent se conformer à l'équation (2) suivante :

$$\langle w, x_i \rangle + b < 0 \text{ pour tout } y_i = -1 \tag{2}$$

Les deux équations (1) et (2) peuvent être combinées :

$$y_i(\langle w, x_i \rangle + b) \geq 0 \tag{3}$$

La contrainte de base de l'approche SVM, telle qu'exprimée dans l'équation (3), doit être satisfaite par le jeu de données d'entraînement. Cependant, cette contrainte seule ne permet pas d'aboutir à une solution unique. Intuitivement, cela peut conduire à une infinité de solutions, comme illustré dans la figure 3. La formulation mathématique à une seule contrainte telle que présentée dans l'équation (3) mène vers un problème mal posé ; d'où l'intérêt de la notion de la marge qui rajoute une contrainte supplémentaire permettant au système de converger vers une solution unique.

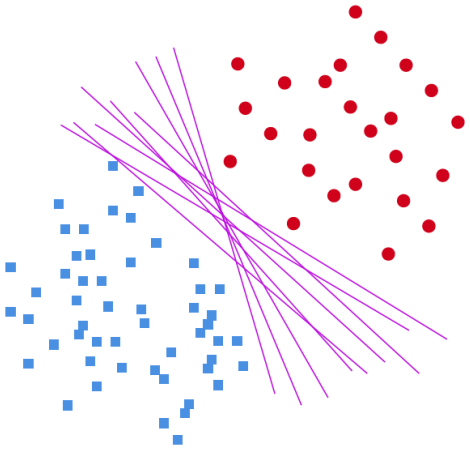


FIGURE 3 – Infinité de solutions possibles pour l'hyperplan séparatif.

2.1 La marge SVM

Pour un dataset d'entraînement $(x_1, y_1), ..., (x_n, y_n)$ linéairement séparable, la contrainte de l'équation (3) seule ne garantit pas la convergence vers une solution unique pour obtenir les paramètres w et b de l'hyperplan séparateur. C'est pourquoi la notion de marge est introduite pour imposer une contrainte supplémentaire liée à la maximisation de cette marge, afin de déterminer une solution unique.

La notion de marge est illustrée dans la figure 4. Elle représente la distance entre le vecteur x_a et l'hyperplan recherché, étant donné :

- x_a est le point du dataset d'entraînement le plus proche de l'hyperplan recherché.
- x'_a est la projection orthogonale du point x_a sur l'hyperplan.
- r est la distance orthogonale entre l'hyperplan et le point x_a.

Ainsi, pour les vecteurs de données appartenant à la classe $y_i = +1$, leur distance par rapport à l'hyperplan dans la direction de w doit être supérieure à r. De même, pour les vecteurs de données appartenant à la classe $y_i = -1$, leur distance par rapport à l'hyperplan dans la direction négative de w doit être supérieure à r. En résumé, pour l'ensemble des vecteurs de données x_i, la contrainte suivante doit être satisfaite :

$$y_i(\langle w, x_i \rangle + b) \geq r \tag{4}$$

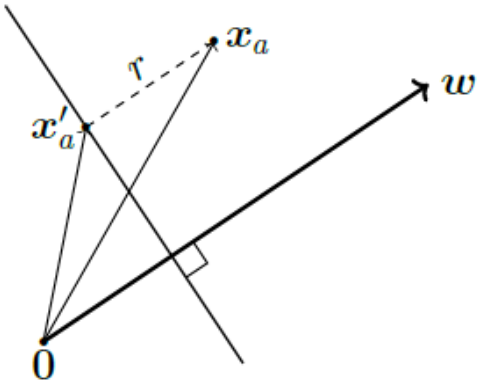


FIGURE 4 – La marge dans SVM.

Or, le problème d'optimisation revient à résoudre :

$$\underset{w,b,r}{\text{maximiser}} \quad r \tag{5}$$

$$\text{sujet à :} \quad y_i(\langle w, x_i \rangle + b) \geq r, \tag{6}$$

$$r > 0, \tag{7}$$

$$\|w\| = 1 \tag{8}$$

Si l'on relâche la contrainte $\|w\| = 1$ et que l'on choisit $r = 1$, cela n'aura aucune incidence sur la fonction d'optimisation. Le choix de cette échelle de $r = 1$ permet de garantir que la valeur de $\langle w, x_a \rangle + b = 1$ pour le vecteur x_a le plus proche de l'hyperplan recherché. Ainsi, la projection orthogonale de x_a, c'est-à-dire x'_a, se trouvera exactement sur la marge. Or,

$$\langle w, x'_a \rangle + b = 0 \tag{9}$$

En connaissant la longueur de w, nous pouvons utiliser le facteur d'échelle r pour calculer la distance absolue entre x_a et x'_a, qui est la projection orthogonale de x_a sur l'hyperplan recherché. En ajoutant les vecteurs, comme illustré dans la figure 4, nous pouvons déduire l'équation suivante :

$$x_a = x'_a + r\frac{w}{\|w\|} \Rightarrow x'_a = x_a - r\frac{w}{\|w\|} \tag{10}$$

Par remplacement du terme x'_a issu de l'équation (10) dans l'équation (9), on obtient :

$$\langle w, x_a - r\frac{w}{\|w\|}\rangle + b = 0 \Rightarrow \langle w, x_a\rangle + b - r\frac{\langle w, w\rangle}{\|w\|} = 0 \tag{11}$$

Puisque x_a est le plus proche vecteur à l'hyperplan recherché, cela implique que $\langle w, x_a\rangle + b = 1$ (car $r = 1$). Après remplacement dans l'équation (11), la marge à maximiser prend ainsi la formule suivante :

$$r = \frac{1}{\|w\|}$$

Le problème d'optimisation relatif à la maximisation de la marge tout en libérant w de la contrainte $\|w\| = 1$ prend la formule suivante :

$$\underset{w,b}{\text{maximiser}} \quad \frac{1}{\|w\|} \tag{12}$$

$$\text{sujet à :} \quad y_i(\langle w, x_i\rangle + b) \geq 1 \tag{13}$$

Le problème d'optimisation relatif à l'approche SVM avec le facteur de $\frac{1}{2}$ pour des fins d'optimisation par descente de gradient peut être formulé comme suit :

$$\underset{w,b}{\text{minimiser}} \quad \frac{1}{2}\|w\|^2$$

$$\text{sujet à :} \quad y_i(\langle w, x_i\rangle + b) \geq 1$$

2.2 Soft Margin SVM

Le modèle qui permet certaines erreurs de classification est connu sous le nom de *SVM Primal avec Marge Souple*. Ce concept est illustré dans la figure 5. L'idée clé est d'introduire une variable de "slack" (ou d'écart) ζ_i associée à chaque paire de données (x_i, y_i), permettant ainsi aux vecteurs x_i de ne pas nécessairement appartenir strictement à leur classe y_i.

En ajoutant la contrainte liée à la marge de tolérance, le problème d'optimisation prend la

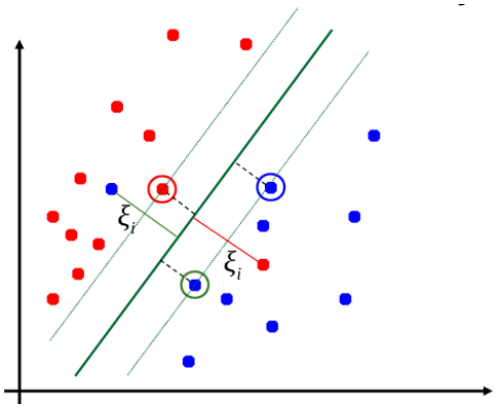


FIGURE 5 – Soft Margin SVM.

formule de Soft margin SVM suivante :

$$\underset{w,b,\zeta}{\text{minimiser}} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\zeta_i \tag{14}$$

$$\text{sujet à :} \quad y_i(\langle w, x_i\rangle + b) \geq 1 - \zeta_i, \tag{15}$$

$$\zeta_i \geq 0 \tag{16}$$

n représente le nombre total de vecteurs d'entrainement. Le paramètre C est un hyperparamètre qui contrôle le compromis entre la marge de tolérance et la classification incorrecte des données d'entraînement. Une valeur plus grande de C entraîne une classification plus rigoureuse des données, en minimisant les erreurs de classification mais pouvant conduire à une marge plus étroite. Une valeur plus petite de C permet une marge plus large, mais tolère davantage les erreurs de classification. En d'autres termes, un paramètre C plus élevé donne plus de poids aux erreurs de classification, ce qui peut entraîner un modèle plus ajusté aux données d'entraînement, tandis qu'un paramètre C plus faible donne moins de poids aux erreurs de classification, permettant ainsi une marge plus large et une plus grande tolérance aux erreurs. Le choix approprié de la valeur de C dépend du problème spécifique et des caractéristiques des données d'entraînement, et peut nécessiter des expérimentations pour trouver la meilleure valeur.

2.3 Dualité de Lagrange

En effet, pour résoudre des problèmes d'optimisation avec des contraintes, il existe diverses approches numériques, parmi lesquelles l'optimisation par les multiplicateurs de Lagrange est l'une des plus couramment utilisées.

Les variables primaires du problème primal d'optimisation de SVM, tel qu'illustré dans l'équation (14) pour le Soft Margin SVM, sont w, b et ζ. La fonction objectif du dual associé au problème primal de l'équation (14) pour le Soft Margin SVM prend la forme suivante :

$$\mathcal{L}(w, b, \zeta, \alpha, \gamma) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\zeta_i - \sum_{i=1}^{n}\alpha_i(y_i(\langle w, x_i\rangle + b) - 1 + \zeta_i) - \sum_{i=1}^{n}\gamma_i\zeta_i \tag{17}$$

étant donné α_i et γ_i sont les multiplicateurs de Lagrange.

Afin de trouver les valeurs des variables primaires qui maximisent la Lagrangienne, on cherche les points où les dérivées partielles par rapport à w, b et ζ sont nulles. En dérivant l'équation (17) par rapport à ces variables, on obtient :

$$\frac{\partial\mathcal{L}}{\partial w} = w^T - \sum_{i=1}^{n}\alpha_i y_i x_i^T \tag{18}$$

$$\frac{\partial\mathcal{L}}{\partial b} = \sum_{i=0}^{n}\alpha_i y_i \tag{19}$$

$$\frac{\partial\mathcal{L}}{\partial \zeta_i} = C - \alpha_i - \gamma_i \tag{20}$$

En annulant les dérivées des équations (18) et (19), on obtient les équations suivantes :

$$\frac{\partial\mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n}\alpha_i y_i x_i \tag{21}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} y_i \alpha_i = 0 \tag{22}$$

Par remplacement de l'équation (21) dans la fonction du dual (17), on obtient le dual suivant :

$$\begin{aligned} \mathfrak{D}(\zeta, \alpha, \gamma) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle + C \sum_{i=1}^{n} \zeta_i - \sum_{i=1}^{n} y_i \alpha_i \langle \sum_{j=1}^{n} y_j \alpha_j x_j, x_i \rangle \\ - b \sum_{i=1}^{n} y_i \alpha_i + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i \zeta_i - \sum_{i=1}^{n} \gamma_i \zeta_i \end{aligned} \tag{23}$$

$$\begin{aligned} \Rightarrow \mathfrak{D}(\zeta, \alpha, \gamma) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle - \sum_{i=1}^{n} y_i \alpha_i \langle \sum_{j=1}^{n} y_j \alpha_j x_j, x_i \rangle + C \sum_{i=1}^{n} \zeta_i \\ - b \sum_{i=1}^{n} y_i \alpha_i + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i \zeta_i - \sum_{i=1}^{n} \gamma_i \zeta_i \end{aligned} \tag{24}$$

$$\begin{aligned} \Rightarrow \mathfrak{D}(\zeta, \alpha, \gamma) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle - \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle + \sum_{i=1}^{n} \alpha_i \\ - \sum_{i=1}^{n} \alpha_i \zeta_i - \sum_{i=1}^{n} \gamma_i \zeta_i + C \sum_{i=1}^{n} \zeta_i - b \sum_{i=1}^{n} y_i \alpha_i \end{aligned} \tag{25}$$

De plus, nous avons à travers l'équation (22) : $\sum_{i=1}^{n} y_i \alpha_i = 0$. En remplaçant les équations (18) et (19) dans l'équation (25), le dual de Lagrange prend la forme suivante :

$$\mathfrak{D}(\zeta, \alpha, \gamma) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle + \sum_{i=1}^{n} \alpha_i + \sum_{i=1}^{n} (C - \alpha_i - \gamma_i) \zeta_i \tag{26}$$

2.4 Les noyaux

Il est vrai que le SVM est initialement conçu pour traiter des cas linéairement séparables. Cependant, il est possible de traiter des cas non linéairement séparables en introduisant la notion de noyaux (ou kernels en anglais).

Il convient de noter que dans l'équation (26), le seul produit scalaire qui apparaît est entre les vecteurs x_i et x_j. Par conséquent, si l'on considère une fonction $\phi(x_i)$ pour représenter x_i et ainsi appliquer une transformation (linéaire ou non linéaire) sur les x_i, le seul changement dans la fonction objectif SVM sera de remplacer la fonction de produit scalaire $\langle x_i, x_j \rangle$ par une autre fonction k. Ainsi, la notion de noyau permet d'appliquer indirectement une transformation sur $(x_1, y_1), ..., (x_n, y_n)$ à travers une fonction k qui substitue au produit scalaire $\langle x_i, x_j \rangle$. Cette option permet au SVM d'être plus flexible et de trouver des limites de décision non linéaires dans des espaces transformés par $\phi(x_i)$.

En d'autres termes, en utilisant SVM, il est possible de construire des classificateurs non linéaires pour les données $x_1, x_2, ..., x_n$ en remplaçant le produit scalaire par une fonction de noyau $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, où $\phi(x_i)$ peut être une fonction de transformation non linéaire. L'idée derrière la fonction de noyau est qu'elle peut être calculée de manière plus efficace que le produit scalaire $\langle \phi(x_i), \phi(x_j) \rangle$. La fonction de noyau k est utilisée comme paramètre pour entraîner le modèle de manière plus flexible, permettant ainsi au SVM de construire des limites de décision non linéaires dans l'espace transformé. Les fonctions du noyau les plus communément utilisées sont :

— Noyau linéaire : $k(x_i, x_j) = x_i^\top x_j + c$.

— Noyau polynomial : $k(x_i, x_j) = (\gamma x_i^\top x_j + c)^d$.

— Radial Basis Function (RBF) : $k(x_i, x_j) = e^{(-\gamma \|x_i - x_j\|^2)}$.

où c est un hyperparamètre qui s'applique aux noyaux linéaire et polynomial, d est un hyperparamètre spécifique au noyau polynomial et correspond au degré du polynôme. γ est un hyperparamètre dans la fonction du noyau RBF qui contrôle la forme et la portée de la fonction de noyau. Plus précisément, γ est un paramètre de régularisation qui détermine la

"largeur" des fonctions de base radiale autour des exemples d'entraînement. Une valeur plus grande de γ conduit à des fonctions de base radiale plus étroites et plus "pointues", tandis qu'une valeur plus petite de γ conduit à des fonctions de base radiale plus larges et plus "douces".

Chapitre 3 Réseaux de neurones à convolution (CNN)

Les travaux de LeCun *et al.* [33] ont donné naissance aux réseaux de neurones à convolution (CNN), une catégorie de réseaux de neurones qui impliquent des couches de convolution. De nos jours, les CNN sont considérés comme l'un des types de réseaux de neurones les plus largement utilisés, en particulier dans les domaines liés à la vision par ordinateur. Cette popularité a été rendue possible grâce au développement de la puissance de calcul des machines et à la disponibilité de grandes quantités de données.

Les CNN ont été largement appliqués dans des domaines tels que la reconnaissance des formes, le traitement automatique du langage naturel et la classification d'images, et de nombreuses recherches récentes [9, 19, 27] ont montré une amélioration significative des performances dans diverses applications grâce à l'utilisation de CNN.

La force des CNN réside dans l'introduction de la couche de convolution, qui permet l'extraction de caractéristiques (ou "features") à partir des données d'entrée. Cette capacité à extraire automatiquement des caractéristiques importantes à partir des données d'entrée sans avoir besoin d'une ingénierie manuelle des caractéristiques est l'un des avantages majeurs des CNN, ce qui les rend adaptés à une large gamme d'applications de traitement d'images et de signaux.

3.1 Les éléments constitutifs de CNN

Dans un réseau de neurones convolutionnels (CNN), une architecture typique comprend généralement des couches de convolution alternées avec des couches de pooling, suivies d'une ou plusieurs couches fully-connected à la fin. Cependant, il existe différents types de couches qui peuvent être utilisées dans un CNN en fonction des besoins spécifiques de la tâche.

Pendant l'entraînement, le CNN apprend à partir des données grâce à un algorithme de rétro-propagation, qui s'inspire du modèle biologique de l'apprentissage basé sur la réponse du cerveau humain. La structure hiérarchique multicouche du CNN lui permet d'extraire des caractéristiques à partir des couches intermédiaires, également connues sous le nom de "Features". Cette capacité d'extraction de caractéristiques à partir des couches intermédiaires simule l'apprentissage profond du cerveau humain, qui apprend de manière dynamique à partir des données d'entrée brutes, comme les photons lumineux.

Il est important de noter que des améliorations significatives des performances des CNN ont été réalisées récemment grâce à de nombreux travaux de recherche. Les avancées continues dans le domaine permettent d'améliorer constamment les performances et les capacités des réseaux de neurones convolutionnels pour une large gamme de tâches d'apprentissage automatique.

3.1.1 Couche de convolution

La couche de convolution est une composante clé d'un réseau de neurones convolutionnels (CNN), et elle est composée d'un ensemble de noyaux de convolution. Cette couche permet d'extraire des caractéristiques en réduisant la dimensionnalité de l'image d'entrée.

Les images d'entrée contiennent des informations chromatiques stockées dans différents canaux. Par exemple, une image en couleur RVB possède trois canaux distincts : rouge, vert et bleu. Comme illustré dans la figure 6, chaque noyau convolue avec les images en utilisant un ensemble spécifique de poids et en multipliant ses éléments avec les éléments correspondants du bloc convolué pour chaque canal constitutif de l'image d'entrée. L'opération de convolution est définie par la formule suivante :

$$f_l^k(p,q) = \sum_c \sum_{x,y} i_c(x,y) e_l^k(u,v)$$

$i_c(x,y)$ correspond à l'élément de l'indice (x,y) du $c^{\text{ème}}$ canal de l'image d'entrée. $e_l^k(u,v)$ correspond à l'élément de l'indice (u,v) du $k^{\text{ème}}$ noyau de convolution de la $l^{\text{ème}}$ couche.

$f_l^k(p,q)$ correspond à l'élément de l'indice (p,q) de la matrice de sortie de la convolution (*feature-map*) liée à la $l^{\text{ème}}$ couche et au $k^{\text{ème}}$ noyeau.

Le résultat de sortie de la convolution (*feature-map*) F_l^k à travers le $k^{\text{ème}}$ noyau au niveau de la $l^{\text{ème}}$ couche est une matrice F_l^k de dimension (P,Q) présentée comme suit :

$$F_l^k = [f_l^k(1,1), ..., f_l^k(p,q), ..., f_l^k(P,Q)]$$

L'opération de convolution peut être classée en différents types selon la taille des filtres et de la direction de la convolution. Par exemple :

— Convolution 1D : Elle est utilisée pour traiter des données unidimensionnelles, telles que des séquences de texte ou de signaux temporels. Les filtres sont appliqués en une seule direction, généralement de gauche à droite.

— Convolution 2D : Elle est utilisée pour traiter des données bidimensionnelles, telles que des images en niveaux de gris ou des cartes de chaleur. Les filtres sont appliqués dans deux directions, généralement en utilisant une fenêtre de taille fixe (par exemple, 3x3 ou 5x5) pour parcourir l'image.

— Convolution 3D : Elle est utilisée pour traiter des données tridimensionnelles, telles que des volumes d'images ou des données volumétriques en imagerie médicale. Les filtres sont appliqués dans trois directions, généralement en utilisant une fenêtre de taille fixe (par exemple, 3x3x3) pour parcourir le volume.

— Convolution transposée : Elle est utilisée pour effectuer l'opération inverse de la convolution, et elle est souvent utilisée dans des architectures de réseaux de neurones comme les générateurs dans les réseaux de neurones génératifs adversariaux (GANs) pour la génération d'images de haute qualité.

Ces différents types de convolution offrent une flexibilité dans la manière dont les filtres sont appliqués aux données d'entrée, ce qui permet d'adapter les CNN à différentes tâches et types de données.

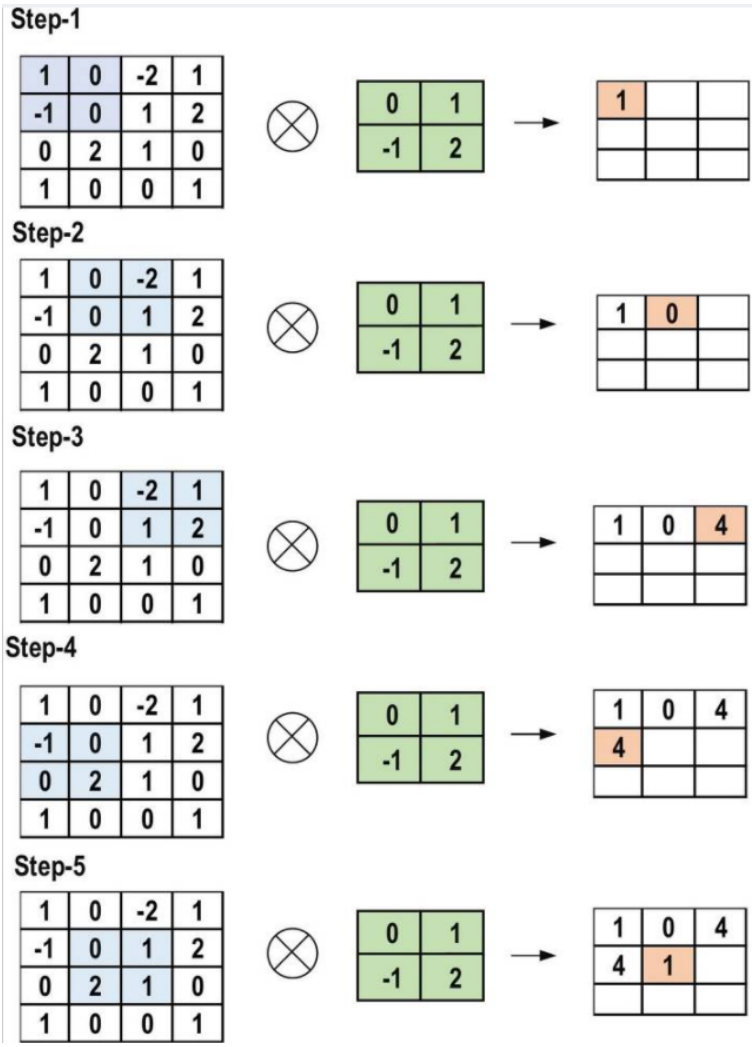


FIGURE 6 – L'opération de la convolution[9].

3.1.2 Couche de sous-échantillonnage (Pooling layer)

En effet, le sous-échantillonnage, également connu sous le nom de *pooling*, est une opération locale importante dans les CNN. Son objectif est de réduire la taille de la *feature-map* générée par la couche de convolution. Une couche de sous-échantillonnage est généralement ajoutée après une couche de convolution dans un CNN pour optimiser le coût calculatoire de l'entraînement. Cette opération de sous-échantillonnage est définie par l'équation suivante :

$$Z_l^k = g_p(F_l^k)$$

Z_l^k représente le résultat de l'opération du sous-échantillonnage *Pooled Feature-map* de la $l^{\text{ème}}$ couche pour le $k^{\text{ème}}$ *Feature-map* d'entrée F_l^k. g_p représente le type de l'opération du sous-échantillonnage.

Il existe plusieurs méthodes de sous-échantillonnage dans les CNN, dont les plus couramment utilisées sont le Max Pooling et l'Average Pooling. Le mécanisme de ces deux méthodes est illustré dans la figure 7. Dans le Max Pooling, la valeur maximale dans une fenêtre donnée de la *feature-map* est sélectionnée pour créer la matrice de *Pooled Feature-map*. En revanche, dans l'Average Pooling, la valeur moyenne dans la fenêtre est considérée pour créer la *Pooled Feature-map*.

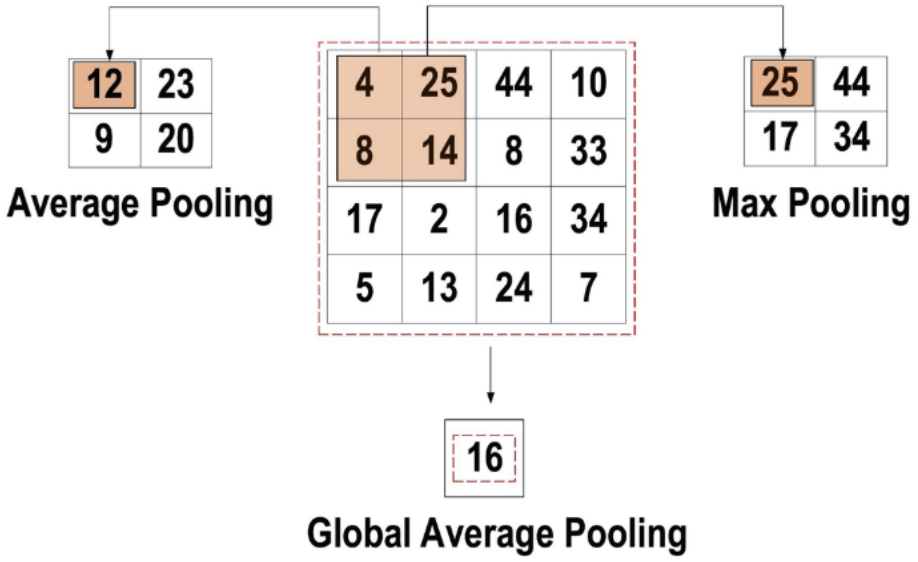


FIGURE 7 – L'opération du sous-échantillonnage[9].

3.1.3 La fonction d'activation

La fonction d'activation a pour rôle de servir de fonction de décision et d'aider à l'apprentissage de modèles complexes. Elle permet au réseau de neurones d'exprimer des relations non linéaires, ce qui lui permet de mieux s'adapter aux résultats et d'améliorer la précision. Pour une feature-map convoluée, la fonction d'activation est définie par :

$$T_l^k = g_a(F_l^k)$$

F_l^k est la sortie de la convolution assignée à la fonction d'activation g_a qui ajoute la non-linéarité au modèle et retourne une sortie transformée T_l^k pour la $l^{\text{ème}}$ couche associée au $k^{\text{ème}}$ noyau de convolution.

Il existe différentes fonctions d'activation dans la littérature, parmi lesquelles les plus couramment utilisées sont :

- **La fonction *binary step* :** elle produit une sortie binaire. La fonction attribue la valeur 1 lorsque l'entrée dépasse un seuil limite, et 0 lorsque l'entrée ne le dépasse pas. Mathématiquement, elle peut être représentée comme suit : $g_a(x) = 0$ si $x < 0$; $g_a(x) = 1$ si $x \geq 0$.
- **La fonction *sigmoïd* :** elle a une forme de courbe en S et peut être représentée mathématiquement comme suit : $g_a(x) = 1/(1 + e^{-x})$.
- **La fonction *tanh* :** elle est similaire à la fonction sigmoïde, car elle prend des nombres réels en entrée, mais sa sortie est limitée à des valeurs entre -1 et 1. Sa représentation mathématique est : $g_a(x) = (e^x - e^{-x})/(e^x + e^{-x})$.
- **La fonction *ReLU* :** c'est la fonction la plus couramment utilisée dans le contexte des réseaux de neurones convolutifs (CNN). Elle convertit les valeurs négatives de l'entrée en 0, et laisse les valeurs positives inchangées. Son faible coût calculatoire est l'un de ses principaux avantages par rapport aux autres fonctions d'activation. Sa représentation mathématique est : $g_a(x) = \max(0, x)$.
- Des variantes de ReLU : il existe plusieurs variantes de ReLU, telles que ReLU modifiée (Leaky ReLU), ReLU exponentielle, ReLU paramétrique, etc., qui ont des propriétés différentes et sont utilisées dans diverses applications.

La figure 8 présente une illustration du concept des fonctions d'activation mentionnées. Ces fonctions sont principalement utilisées pour appliquer des transformations non linéaires sur la matrice de convolution. Parmi celles-ci, les fonctions ReLU et ses variantes sont préférées, car elles sont connues pour aider à optimiser le calcul du gradient dans les opérations d'apprentissage[9].

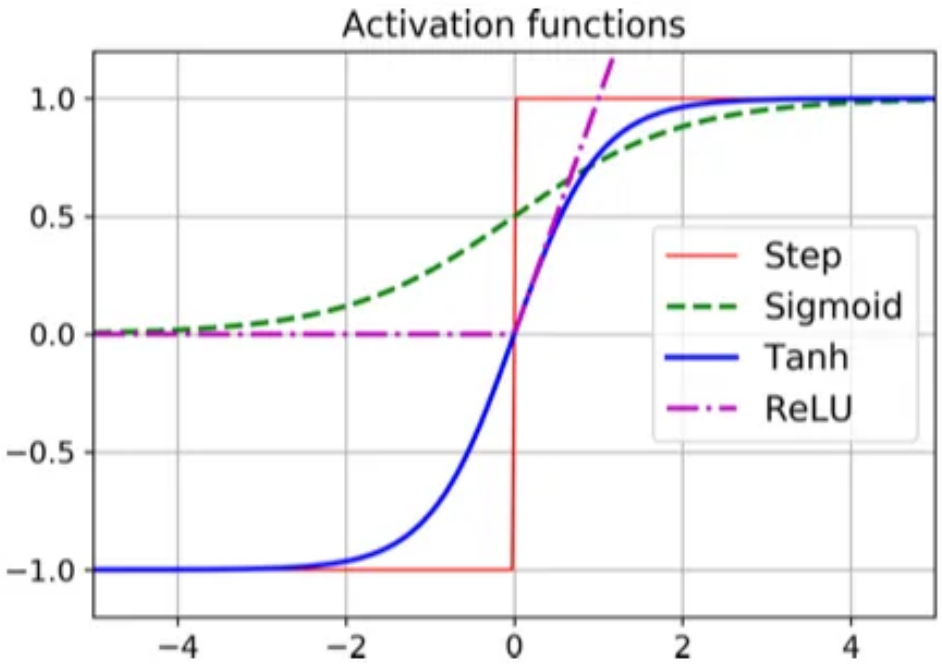


FIGURE 8 – Les fonctions d'activation.

3.1.4 Normalisation de lot (Batch normalization)

La normalisation de lot (Batch Normalization) est une méthode utilisée pour standardiser et centrer les matrices de feature-maps d'entrée générées à partir de la convolution dans un réseau de neurones. Elle facilite l'entraînement du réseau en présentant deux avantages majeurs. Tout d'abord, elle augmente le taux d'apprentissage (learning rate), ce qui permet au réseau de converger plus rapidement lors de l'entraînement. De plus, elle réduit le poids des paramètres d'initialisation du réseau, ce qui contribue à une meilleure stabilité de l'apprentissage. En conséquence, le réseau CNN peut apprendre plus rapidement et avec une consommation de ressources réduite.

La normalisation de lot pour une feature-map transformée F_l^k est formulée mathématiquement comme suit :

$$N_l^k = \frac{F_l^k - \mu_B}{\sigma_B^2 + \varepsilon}$$

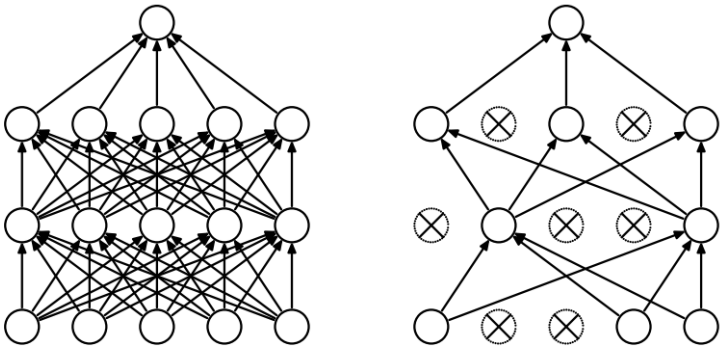
où :

N_l^k représente la *feature-map* normalisée. F_l^k représente la *feature-map* d'entree. μ_B et σ_B^2 représentent la moyenne et la variance de la feature-map d'entrée. ε est une petite valeur ajoutée pour assurer la stabilité numérique et éviter la division par zéro lors du calcul de la normalisation.

3.1.5 Dropout

Le Dropout est une technique de régularisation utilisée pour réduire le sur-apprentissage lors de l'entraînement d'un modèle. Elle consiste à désactiver temporairement certains neurones, ainsi que toutes les connexions entrantes et sortantes, comme illustré dans la figure 9. Cette désactivation aléatoire de certaines connexions conduit à la création de plusieurs architectures de réseau "aminées", et finalement, un réseau représentatif est sélectionné avec de petits poids.

L'efficacité empirique du Dropout a été démontrée par plusieurs études. Srivastava *et al.* [42] ont montré une nette amélioration des performances des CNN lorsque le Dropout est utilisé. La figure 10 illustre clairement les avantages de l'utilisation du Dropout, avec une amélioration significative des performances des CNN observée lorsque le Dropout est appliqué.



(a) Standard Neural Net (b) After applying dropout.

FIGURE 9 – Dropout dans un réseau de neurones artificiels[42].

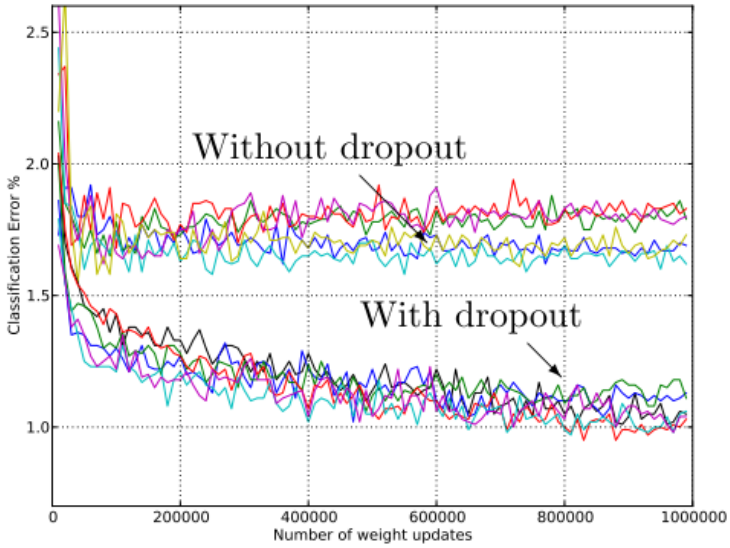


FIGURE 10 – L'avantage empirique du Dropout[42].

3.1.6 Fully connected layer

La couche de fully connected est généralement située à la dernière couche d'un CNN. Elle prend un vecteur en entrée et produit un nouveau vecteur en sortie, comme illustré dans la figure 11. Pour cela, elle applique une combinaison linéaire des valeurs d'entrée, suivie éventuellement d'une fonction d'activation.

Trois paramètres sont nécessaires pour définir une couche fully connected : la taille de la couche, c'est-à-dire le nombre de neurones qu'elle contient, le nombre d'entrées qui représente la dimension de l'entrée du réseau, et le nombre de sorties qui représente la dimension de la sortie produite par la couche fully connected. Ces paramètres permettent de spécifier la structure et la taille de la couche fully connected, ce qui influence directement la capacité du réseau à apprendre des représentations complexes et à effectuer des prédictions précises.

3.2 Les architectures CNN

Depuis 1989, diverses améliorations ont été apportées à l'architecture des réseaux de neurones convolutionnels (CNN). Ces améliorations peuvent être regroupées en catégories telles que l'optimisation des paramètres, la régularisation, la reformulation structurelle, et d'autres.

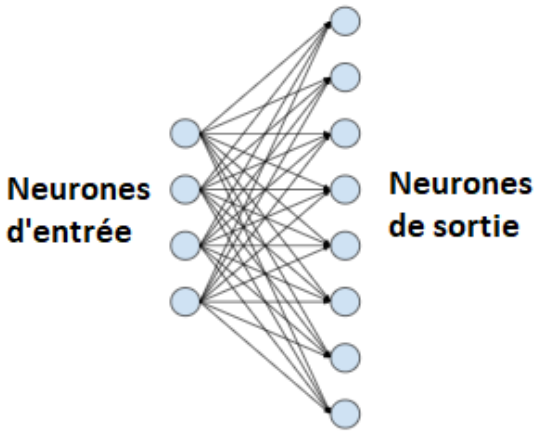


FIGURE 11 – Couche *Fully connected*.

Cependant, il est observé que la principale source d'amélioration des performances des CNN provient de la restructuration des unités de traitement et de la conception de nouveaux blocs. La plupart des innovations dans les architectures des CNN sont axées sur la profondeur et l'exploitation spatiale. En fonction du type de modifications apportées à l'architecture, les CNN peuvent être globalement classés en sept catégories différentes[30], à savoir :

— l'exploitation spatiale,

— la profondeur,

— les trajets multiples,

— la largeur,

— l'exploitation de *Feature-maps*,

— le renforcement des canaux,

— CNN basés sur l'attention.

Les différentes architectures de CNN connues sont présentées dans la figure 12 avec leurs catégories respectives.

Récemment, les progrès architecturaux des CNN se sont concentrés sur la conception de nouveaux blocs pour améliorer la représentation du modèle en exploitant les feature-maps ou en manipulant l'image d'entrée avec l'ajout de canaux artificiels. Une autre tendance émergente est la conception d'architectures légères qui maintiennent de bonnes performances tout en

étant adaptées aux matériels limités en ressources [9]. Un exemple notable est l'architecture GoogleNet, qui utilise des petits réseaux et remplace la convolution conventionnelle par une technique appelée "point-wise group convolution" pour rendre le processus de calcul plus efficace.

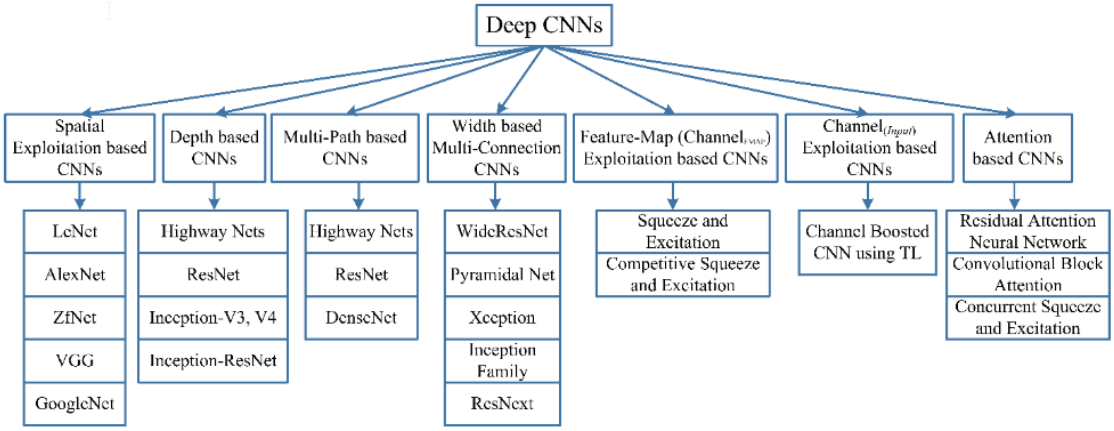


FIGURE 12 – Les différentes architectures développées dans la littérature[30].

Les détails complets des architectures mentionnées ci-dessus ne sont pas cités ici. Pour obtenir plus d'informations, il est recommandé de consulter les ressources en ligne abondantes. Pour approfondir le sujet, la référence [9] peut également être consultée.

Chapitre 4 Hybride CNN-SVM

En plus de l'étape de pré-traitement des images d'entrée, l'extraction des paramètres des images est un élément clé pour le succès de la classification. L'objectif de cette étape est d'extraire les caractéristiques les plus discriminantes des données d'entrée parmi les différentes classes, tout en conservant les caractéristiques invariantes au sein de la même classe. Cela permet au modèle de capturer les informations essentielles pour la tâche de classification, en identifiant les motifs significatifs dans les données d'entrée qui permettent de distinguer les différentes classes. Une extraction efficace des paramètres des images permet au modèle d'apprendre des représentations de haute qualité, ce qui contribue à améliorer la performance de classification du modèle.

L'article de Szarvas *et al.* [44] a proposé et évalué l'utilisation de réseaux de neurones convolutifs (CNN) et de machines à vecteurs de support (SVM) dans un système de détection des piétons. En plus de chercher à atteindre la plus grande précision possible du système, l'un des objectifs de ce travail était de fournir une comparaison rigoureuse entre les performances des CNN et des SVM, qui étaient deux des classifieurs les plus populaires dans les systèmes de détection de piétons à l'époque.

L'approche proposée par Szarvas *et al.* [44] consiste à combiner les CNN et les SVM pour former une architecture appelée "*CNN-features+SVM combination*", représentant ainsi une première tentative de combinaison de ces deux approches. Plus spécifiquement, les paramètres des images d'entrée sont extraits à l'aide du CNN, puis les SVM sont entraînés à partir de ces paramètres extraits. Les résultats expérimentaux obtenus par Szarvas *et al.* [44] démontrent que la précision de cette combinaison est supérieure à celle du CNN de base, mettant ainsi en évidence l'importance de l'extraction des paramètres dans un système de classification.

Les résultats de cette étude ont montré que l'extraction des paramètres des images d'entrée est inévitable pour obtenir une détection de piétons de haute précision dans leur système.

Ainsi, cette recherche pionnière a mis en avant l'importance de l'extraction des paramètres dans le contexte de la détection de piétons, et a ouvert la voie à de futures recherches sur l'utilisation de combinaisons d'approches pour améliorer les performances des systèmes de détection d'objets.

Les travaux de Lauer *et al.* [31] ont étendu l'utilisation de SVM sur les paramètres extraits par CNN dans le domaine de la reconnaissance des chiffres manuscrits. Ils ont proposé une approche hybride combinant CNN et SVM dans un système de reconnaissance des chiffres manuscrits, basé sur une architecture appelée "Trainable Feature Extractor - SVM" (TFE-SVM). Cette méthode a été testée sur un dataset de référence bien connu, MNIST.

L'idée centrale de l'approche proposée est d'extraire les paramètres des images en utilisant un "Trainable Feature Extractor" (TFE) basé sur une architecture CNN, comme illustré dans la figure 13. Les sorties de la couche C5 du CNN sont soit dirigées vers les 10 sorties pour la phase d'apprentissage du CNN, soit utilisées comme paramètres d'images pour l'entraînement du SVM.

Les résultats expérimentaux rapportés par Lauer *et al.* [31] montrent que le TFE-SVM a surpassé à la fois l'architecture CNN "LeNet5" et les SVM formés sur l'ensemble du dataset MNIST. Le taux d'erreur obtenu par SVM était de 1,4%, celui de LeNet5 était de 0,71%, tandis que celui obtenu avec TFE-SVM était de 0,56%. Ces résultats démontrent l'efficacité de l'approche TFE-SVM dans le contexte de la reconnaissance des chiffres manuscrits, et soulignent l'avantage de combiner les CNN et les SVM pour améliorer les performances de détection d'objets.

Les recherches dans le domaine de la reconnaissance des chiffres manuscrits continuent à viser des niveaux de précision compétitifs. Cependant, les architectures hybrides combinant CNN et SVM restent principalement limitées à la reconnaissance des chiffres latins. Les travaux de Ahlawat et Choudhary [1], Niu et Suen [37] proposent un modèle Hybride CNN-SVM pour résoudre le problème de reconnaissance des chiffres manuscrits latins. Ce modèle utilise le CNN comme extracteur automatique de paramètres et le SVM comme classifieur final. Les

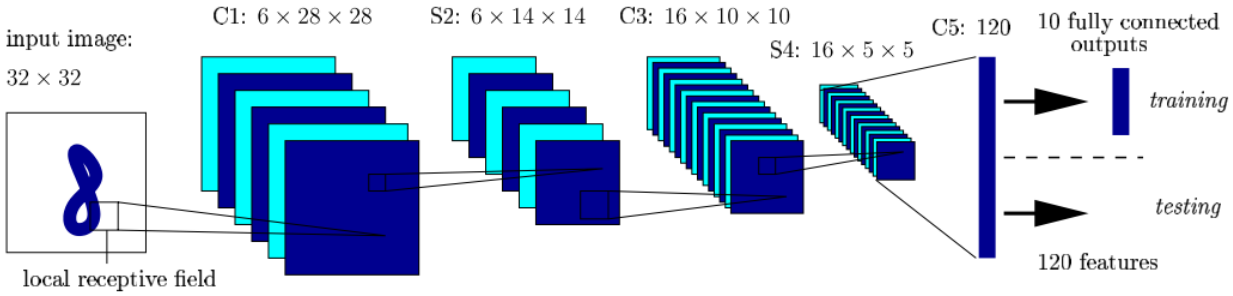


FIGURE 13 – L'architecture Trainable Feature Extractor (TFE)[31].

chercheurs ont mené des études importantes pour évaluer l'efficacité et la faisabilité de cette approche hybride appliquée sur MNIST, en se concentrant sur deux aspects : la précision de la reconnaissance et les performances de fiabilité. Les résultats rapportés montrent que cette approche Hybride CNN-SVM a atteint des taux de reconnaissance respectifs de 98,88% et 99,81%, ce qui représente une amélioration de 1,03% (98,88% - 97,85%) par rapport à l'utilisation de SVM seul, et de 0,4% (99,81% - 99,41%) par rapport à l'utilisation de CNN seul. Il convient de noter toutefois que ces résultats sont basés sur l'évaluation sur MNIST, qui est considérée comme une base de données simple et de nombreuses approches atteignent des niveaux de précision comparables.

En dehors du domaine de la reconnaissance des chiffres manuscrits, le modèle Hybride CNN-SVM a également été appliqué à d'autres domaines tels que la reconnaissance automatique du manuscrit arabe [4, 24], l'imagerie médicale [43], la reconnaissance des formes [26], la classification des signaux d'électromyographie (EMG) [46], etc.

Malgré de nombreuses recherches démontrant l'amélioration du taux de reconnaissance des chiffres manuscrits de MNIST grâce à l'utilisation du modèle Hybride CNN-SVM, son avantage par rapport à l'utilisation de CNN et SVM seuls n'est pas clair lorsqu'il est appliqué à des ensembles de données plus hétérogènes.

Chapitre 5 Reconnsaissance du manuscrit

La reconnaissance automatique des caractères manuscrits revêt une grande importance en raison de ses nombreuses applications pratiques et de ses implications financières. L'industrie exige de plus en plus des taux de reconnaissance élevés avec une grande fiabilité. Dans cette section, nous nous intéressons aux techniques récentes appliquées dans le domaine de la reconnaissance des caractères manuscrits latins et arabes.

5.1 Les caractères latins

Le jeu de données EMNIST, introduit par Cohen *et al.* [16] en 2017, est composé d'un grand nombre d'échantillons et de collections d'images manuscrites pour les chiffres et les caractères latins, comme décrit en détail dans la section 6.1 des datasets utilisés. Il comprend les ensembles de données suivants :

- **EMNIST MNIST** ayant une structure identique à MNIST et comporte des images des chiffres manuscrits réparties dans 10 classes.
- **EMNIST Letters** est représenté par 26 classes de caractères latins. On n'y distingue pas les majuscules des minuscules.
- **EMNIST Balanced** comprend 47 classes de chiffres et de lettres. Pour certaines lettres, on distingue les majuscules des minuscules.
- **EMNIST By_Class** et **EMNIST By_Merge** contiennent tous deux les mêmes images des chiffres et lettres manuscrits mais ne diffèrent que par le nombre de classes attribuées selon la distinction entre les lettres majuscules et les lettres minuscules.

Les datasets constitutifs de EMNIST présentent ainsi des datasets référentiels plus exigeants.

MNIST, qui se compose uniquement de chiffres manuscrits latins, a été largement utilisé dans de nombreux travaux d'apprentissage automatique, rapportant des taux de reconnaissance supérieurs à 99%. Cependant, avec de nombreux travaux atteignant un taux d'erreur aussi

bas que 99,8%, MNIST est considéré comme étant résolu. Pour cette raison, Cohen *et al.* ont introduit en avril 2017 la base de données MNIST étendue (EMNIST), qui comprend à la fois des chiffres et des lettres manuscrites et qui a la même structure que la base de données MNIST. EMNIST est ainsi devenu le référentiel par défaut pour de nombreux travaux de reconnaissance automatique des chiffres et des lettres latines.

Les résultats de plusieurs travaux sur EMNIST seront présentés dans le tableau 1. De plus, dans l'article original de Cohen *et al.* [16], les auteurs ont inclus des résultats de classification en utilisant un classificateur linéaire ainsi qu'une technique appelée OPIUM (Online Pseudo-Inverse Update Method). Ils rapportent les précisions obtenues pour chacun des datasets composant EMNIST.

Bien que la base de données EMNIST soit moins étudiée dans la littérature par rapport à MNIST, plusieurs travaux ont été réalisés depuis sa création, appliquant différentes approches sur ses datasets constitutifs. Par exemple, Ghadekar *et al.* [25] ont utilisé la transformée en ondelettes discrète (DWT) et la transformée en cosinus discrète (DCT) pour l'extraction des caractéristiques avant d'utiliser KNN et SVM pour la classification des images issues d'EMNIST. Ils ont rapporté que SVM surpassait KNN dans tous les scénarios, atteignant une précision allant jusqu'à 89,51% pour les EMNIST Letters et 97,74% pour les EMNIST Digits. Dufourq et Bassett [21] ont atteint une précision de 88,3% pour EMNIST Balanced et de 99,3% pour EMNIST Digits en utilisant la technique de l'apprentissage profond.

TABLE 1 – Résultats de la littérature pour les différents datasets de EMNIST

Technique	Référence	By_Class	By_Merge	Balanced	Letters	MNIST
Classifieur linéeaire	Cohen *et al.* [16]	51,80%	50,51%	50,93%	55,78%	84,70%
OPIUM	Cohen *et al.* [16]	69,71%	72,57%	78,02%	85,15%	95,90%
DWT-DCT + SVM	Ghadekar *et al.* [25]	-	-	-	89,51%	97,74%
EDEN	Dufourq et Bassett [21]	-	-	88,3	89,51%	99,30%

5.2 Les caractères arabes

La reconnaissance des chiffres et des alphabets latins manuscrits a fait l'objet de nombreuses études au cours des dernières décennies, avec de nombreuses méthodes proposées atteignant un taux de reconnaissance très élevé. Cependant, les progrès de la recherche sont moins marqués pour les écritures autres que latines.

5.2.1 Les lettres arabes

De nos jours, la langue arabe (al 'arabīya) est la langue officielle de plus de 20 pays et parlée par plus de 300 millions de personnes dans le monde. En plus de cela, de nombreuses langues utilisent les mêmes alphabets arabes : le persan et ses variétés. La langue arabe est écrite horizontalement de la droite vers la gauche. Elle est composée de vingt-huit alphabets primaires sans caractères minuscules ni majuscules. La forme du caractère dépend de sa position dans le mot.

En raison de son utilisation généralisée, la reconnaissance automatique de l'écriture manuscrite en arabe suscite un intérêt croissant au sein de la communauté de recherche. Il s'agit d'un sujet majeur dans le domaine de la recherche sur la langue arabe en général.

Pour entraîner, tester et valider les modèles de reconnaissance automatique de l'écriture manuscrite en arabe, il est nécessaire de disposer de jeux de données de référence. Il existe plusieurs ensembles de données couramment utilisés dans la littérature, parmi lesquels on peut citer :

— **La base de données IFN/ENIT** issue des travaux de Dimauro *et al.* [20], développée par l'Institut fur Nachrichtentechnik et l'Ecole Nationale d'Ingénieurs de Tunis. Elle comporte 26549 images de noms de villes/villages tunisiens écrits par 411 personnes.

— **Center for Pattern Recognition and Machine Intelligence (CENPARMI)** Al-Ohali *et al.* [2] dispose d'un certain nombre de bases de données dont certaines sont destinées à la reconnaissance des chèques manuscrits en arabe.

— **Arabic Handwritten Characters Dataset (AHCD)** issue des travaux de El-Sawy

et al. [23]. Le dataset comporte des images de 32x32 pixels des lettres arabes.

Le travail novateur de Sarfraz *et al.* dans le domaine de la reconnaissance automatique du texte arabe utilise l'apprentissage profond. Le système développé est basé sur une police unique (police *Naskh*) et prend en charge plusieurs tailles de police. Les images d'entrée sont en niveaux de gris et ont une taille de 640x480 pixels, contenant près de 200 caractères. Pour l'entraînement du modèle, Sarfraz *et al.* utilisent une base d'entraînement composée de 149 images réparties sur 33 classes de caractères (alphabets). En plus des 28 classes de caractères habituelles, les distinctions graphiques de certains alphabets en fonction de leur position dans le mot sont prises en compte et incluses dans l'ensemble d'entraînement.

Le système proposé par Sarfraz *et al.* est composé de plusieurs modules, notamment le prétraitement, la segmentation, l'extraction des paramètres et la reconnaissance. Le système a atteint une précision de 73%. Cependant, il présente certaines limitations liées à la contrainte de la police unique ainsi qu'à la taille relativement faible des ensembles de données d'entraînement et de test par rapport aux bases de données actuelles.

La digitalisation de l'information a permis la création de datasets référentiels dans le domaine des lettres arabes, rendant ainsi ces ressources accessibles à la communauté de chercheurs. Cette disponibilité de données a stimulé la recherche dans ce domaine, avec de nombreux travaux successifs cherchant à obtenir des taux de reconnaissance compétitifs entre eux. Par exemple, Sahloul *et al.* [39] ont présenté une approche basée sur les caractéristiques structurelles, statistiques et morphologiques des caractères, atteignant une précision moyenne de 88% en utilisant l'ensemble de données CENPRMI.

Elleuch *et al.* [24] ont été les pionniers à appliquer l'approche Hybride CNN-SVM dans la reconnaissance des lettres manuscrites arabes, en utilisant une approche similaire à celle présentée dans leur travail. Les ensembles d'entraînement et de test étaient issus des bases de données HACDB et IFN/ENIT. Les résultats obtenus ont montré que le modèle Hybride CNN-SVM a significativement réduit l'erreur de classification par rapport à CNN, passant de 14,71% à 5,83% sur la base de données HACDB (66 classes). Cette conclusion a consolidé

empiriquement l'hypothèse de l'avantage de l'approche Hybride CNN-SVM par rapport aux méthodes de l'état de l'art.

El-Sawy *et al.* [23] ont utilisé une architecture CNN formée et testée sur le dataset Arabic Handwritten Character Dataset (AHCD), composé de 16800 caractères arabes manuscrits. Les résultats expérimentaux ont montré une amélioration significative par rapport à d'autres algorithmes de classification, avec un taux de classification de 94,9% sur les données de test.

Shams *et al.* [41] ont présenté un système Hybride CNN-SVM pour la classification des caractères manuscrits arabes, atteignant un taux de reconnaissance de 95,07%. De plus, Altwaijry et Al-Turaiki [8] ont appliqué CNN sur le dataset Arabic Handwritten Character Dataset (AHCD) et ont rapporté un taux de reconnaissance de 97%. Ces travaux démontrent les avancées significatives dans le domaine de la reconnaissance des lettres arabes en utilisant des approches basées sur les réseaux de neurones convolutionnels (CNN) et l'approche Hybride CNN-SVM.

5.2.2 Les chiffres arabes orientaux

Les chiffres arabes orientaux font référence à une variante graphique du système de numération indo-arabe, utilisée dans la partie orientale du monde arabe.

Les pionniers de l'utilisation de l'apprentissage profond dans la reconnaissance des chiffres arabes manuscrits sont Das *et al.* [18]. Ils ont constitué une base de données de 3000 échantillons de chiffres en utilisant des spécimens de caractères manuscrits numérisés optiquement, comprenant les 10 symboles des chiffres arabes. Les échantillons ont été redimensionnés en images de 32x32 pixels et convertis en images binaires par seuillage. Les résultats expérimentaux ont montré un taux de reconnaissance de 94,93% évalué après triple validation croisée.

Une approche similaire a été proposée par Pan *et al.* [38] pour la reconnaissance des chiffres persans qui ont une ressemblance avec les chiffres arabes. Ils ont utilisé une approche basée sur les réseaux de neurones convolutifs (CNN) et l'algorithme K-SVD pour créer un dictionnaire

pour les représentations creuses. Les données du dictionnaire issues de K-SVD ont été utilisées comme première couche du CNN, qui agit comme le classifieur final. Les expériences ont montré une précision de 99,22% pour la base de données de chiffres manuscrits persans du CENPARMI, comparativement à SVM.

Lekhal *et al.* [34] ont proposé une méthode basée sur les k-plus proches voisins (KNN) et le perceptron multicouche (MLP) pour la reconnaissance des chiffres arabes. Ils ont atteint un taux de reconnaissance de 99% en utilisant MLP et 97% en utilisant KNN sur une petite base de données de 600 chiffres arabes, comprenant 200 images de test et 400 images d'entraînement.

Takruri *et al.* [45] ont proposé un système de reconnaissance à trois niveaux basé sur Fuzzy C-Means, SVM et unique pixels. L'approche a été testée sur un ensemble de données public comprenant 3510 images, avec 40% pour les tests et 60% pour l'entraînement. Les auteurs ont rapporté une précision de 88

El-Sawy *et al.* [22] ont utilisé un modèle CNN basé sur l'architecture LeNet-5, comprenant 7 couches au total, dont 3 couches convolutives, 2 couches de Max Pooling et 2 couches Fully connected. Le modèle a été entraîné sur l'ensemble de données MADBase, et les auteurs ont rapporté un taux de reconnaissance de 88% sur l'ensemble de test.

Ashiquzzaman et Tushar [10] ont proposé une approche basée sur CNN utilisant plusieurs couches de convolution avec l'activation ReLU, et utilisant Dropout comme couche de régularisation. La sortie est ensuite introduite dans une couche Fully connected avec une activation softmax pour obtenir une prédiction pour chaque classe. L'approche a été appliquée sur le dataset *CMATERDB Arabic handwritten digit dataset* et a permis d'atteindre une précision de 97,4

Les auteurs de l'article Loey *et al.* [36] ont utilisé une nouvelle approche d'apprentissage non supervisé basée sur un encodeur automatique empilé (SAE) pour reconnaître les chiffres manuscrits arabes. Ils ont appliqué cette approche CNN sur le dataset MADBase, qui est le même que celui utilisé par El-Sawy *et al.* [22]. Leur approche a montré des améliorations

significatives par rapport à d'autres algorithmes de classification, avec une précision moyenne de 98,5%.

Dans une autre étude, Al-Wajih et Ghazali [3] ont appliqué différents algorithmes basés sur le "Local Binary Pattern (LBP)" pour la reconnaissance des chiffres arabes, obtenant des résultats encourageants, avec un taux de réussite le plus élevé atteignant 99,26%.

Enfin, Alkhawaldeh *et al.* [7] ont proposé une nouvelle approche appelée "ensemble deep transfer learning" (EDTL) et l'ont appliquée sur le dataset MADBase. L'EDTL est formé sur de grands ensembles de données pour extraire les caractéristiques pertinentes qui sont ensuite utilisées comme entrées dans un classifieur de réseau de neurones artificiels entièrement connecté. Les résultats expérimentaux ont montré des performances significatives, avec une précision de 99,83% par rapport aux méthodes de base.

Chapitre 6 Méthodologie

Le travail vise principalement à explorer empiriquement l'approche hybride combinant CNN et SVM sur un large éventail de datasets variés en nature de données, en taille d'échantillons, en dimensions des images d'entrée, ainsi qu'en nombre de classes. Dans le modèle hybride que nous proposons, SVM est utilisé comme classifieur final et CNN comme extracteur automatique de caractéristiques (ou réducteur de dimension) à partir d'images brutes.

Nous étudierons les avantages plausibles de notre approche proposée à travers une application sur des datasets variés constitués des alphabets latins, alphabets arabes, chiffres latins et chiffres arabes. Nous évaluerons ainsi dans quelle mesure l'approche hybride peut être avantageuse et chercherons à obtenir des taux de reconnaissances compétitifs comparativement aux travaux de la littérature à travers cette même approche.

Dans ce chapitre, nous présentons les différents datastes utilisés pour atteindre nos objectifs de recherche. Ensuite, on abordera l'approche proposée.

6.1 Datasets

L'importance de bons datasets référentiels est critique, en particulier dans le domaine de l'apprentissage automatique. Ces datasets référentiels fournissent un moyen rapide, quantitatif et équitable d'analyser et de comparer différentes approches et techniques d'apprentissage. Cela permet aux chercheurs d'avoir rapidement un aperçu des performances et des particularités des méthodes et des algorithmes.

Il existe plusieurs datasets référentiels qui sont largement utilisés dans le domaine de l'apprentissage machine et sont devenus très compétitifs. Pour les besoins de notre recherche, on se fie à 3 collections de datasets, tel que illustré dans le tableau 2 et qui répondent à nos trois objectifs suivants :

— EMNIST[16] pour l'application sur les alphabets et chiffres latins. EMNIST est composé de plusieurs sous-datasets.

— Arabic Handwritten Characters Dataset (AHCD)[23] pour l'application sur les lettres des alphabets arabes.

— MADBase [3] pour l'application sur les chiffres manuscrits arabes.

TABLE 2 – Les datasets utilisés.

Dataset	MNIST	Letters	Balanced	By_Merge	By_Class	AHCD	MADBase
Entrainement	60,000	88,800	112,800	697,932	697,932	13,440	60,000
Test	10,000	14,800	18,800	116,932	116,932	3,360	10,000
Total	70,000	103,600	131,600	814,255	814,255	16,800	70,000
Nombre de classes	10	26	47	47	62	28	10

6.1.1 EMNIST

Avant l'apparition de EMNIST, l'ensemble de données MNIST fut considéré pendant longtemps comme étant la base de données d'essai standard dans de nombreux travaux de la littérature pour les systèmes d'apprentissage, de classification et de vision par ordinateur. La taille et ses exigences de stockage relativement petites, ainsi que l'accessibilité et la facilité d'utilisation de la base de données elle-même, contribuent à son adoption généralisée. Il existe dans la littérature une quantité importante de travaux testés sur MNIST et signalant un taux d'erreur inférieur à 1 %. Le fait que beaucoup de ces travaux approchent un taux d'erreur de 0,2% suggère que cette valeur peut être irréductible pour ce problème. Pour cette raison, MNIST est considéré comme déjà résolu.

Cohen *et al.* [16] ont introduit en avril 2017 la base de données MNIST étendue (EMNIST) [4], composée à la fois de chiffres et de lettres manuscrits, et partageant la même structure que la base de données MNIST. Cette dernière a été dérivée d'un ensemble de données plus vaste connu sous le nom de base de données spéciale NIST 19 qui contient des chiffres, des lettres manuscrites majuscules et minuscules. EMNIST suit le même paradigme de conversion

3. https ://datacenter.aucegypt.edu/shazeem/
4. https ://www.nist.gov/itl/products-and-services/emnist-dataset

utilisé pour créer l'ensemble de données MNIST. Le résultat est un ensemble de datasets qui partagent la même structure d'image et les mêmes paramètres que MNIST d'origine, permettant une compatibilité directe avec tous les classificateurs et systèmes validés et testé sur MNIST. La figure 14 illustre l'organisation de EMNIST et présente les caractéristiques principales des datasets constitutifs de la collection : la nature des classes incluses et le nombre d'échantillons par classe dans chacun des 6 datasets ainsi que le nombre de vecteurs d'entraînement et de test pour chaque ensemble de données.

Les ensembles de données **EMNIST By_Class** et **EMNIST By_Merge** contiennent tous deux les mêmes images manuscrites mais ne diffèrent que par le nombre de classes attribuées. L'origine de cette disparité dans le nombre de classes réside dans l'organisation des catégories relatives à la distinction entre les lettres majuscules et les lettres minuscules tel que montré dans la figure 14.

L'ensemble de données **EMNIST Balanced** contient l'ensemble de données appartenant à 47 classes. Il a été conçu afin d'éviter les biais d'erreurs de classification résultant uniquement d'une mauvaise classification entre les lettres majuscules et minuscules. La figure 15 montre un échantillon d'images que comporte EMNIST Balanced.

L'ensemble de données **EMNIST Letters** cherche à réduire davantage les erreurs résultant d'une confusion de classe en fusionnant toutes les images des classes majuscules et minuscules. La figure 16 montre un échantillon d'images contenus dans le dataset EMNIST Letter.

L'ensemble de données **EMNIST MNIST** est destiné à remplacer directement le dataset MNIST.

6.1.2 AHCD

Pour les besoins de notre recherche quant à la reconnaissance du manuscrit des lettres arabes, on se fie à la base Arabic Handwritten Characters Dataset (AHCD)[23]. El-Sawy *et al.* ont développé AHCD afin d'améliorer le résultat de la précision de la reconnaissance automatique du manuscrit arabe. L'ensemble de données est composé de 16800 caractères écrits par 60

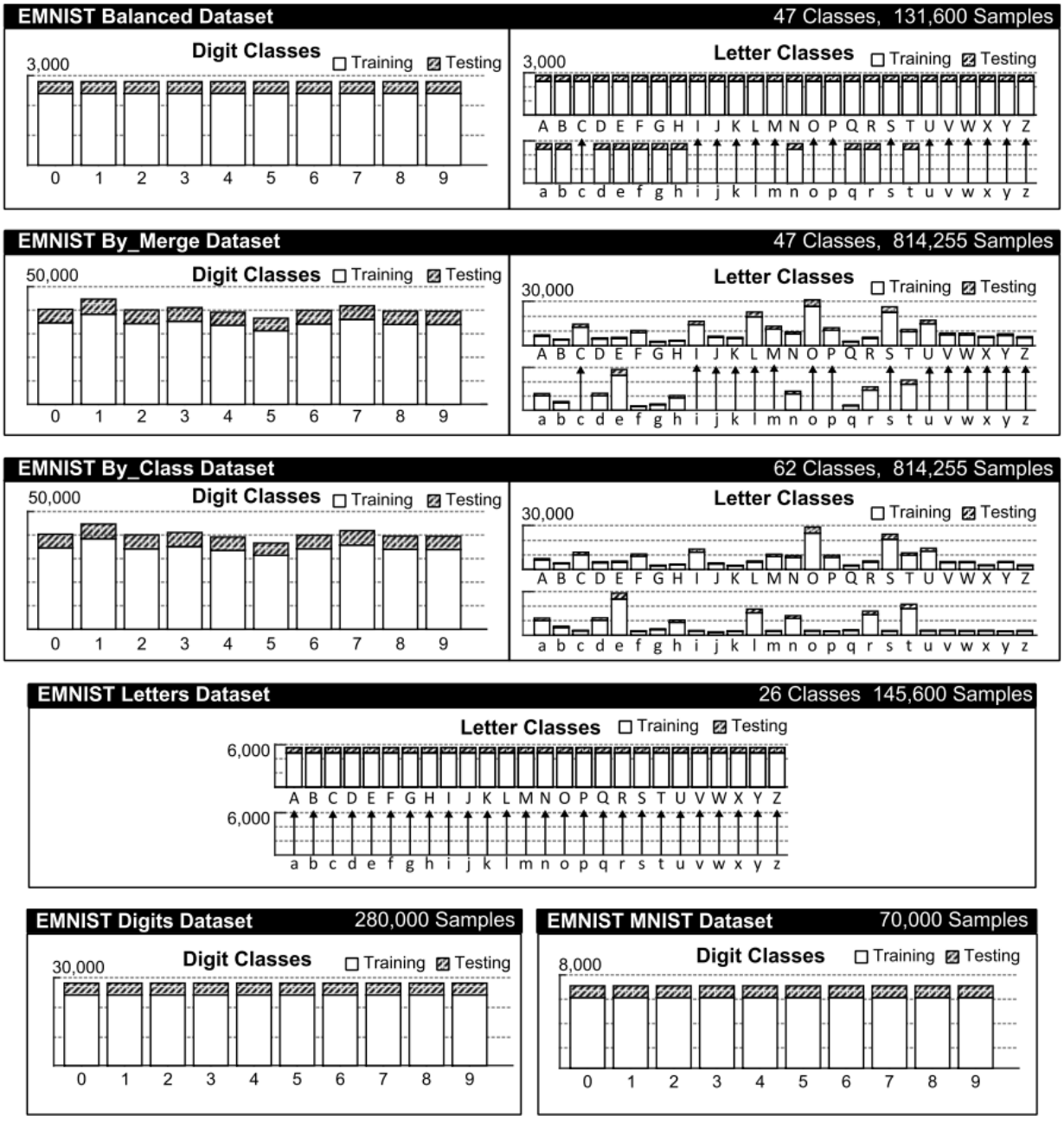


FIGURE 14 – Les datasets constitutifs de EMNIST[16].

participants. Ensuite, les manuscrits sont numérisés à la résolution de 300 dpi. Chaque participant a écrit dix fois chaque caractère (de Alif à Yaa). L'étape de pré-traitement a été réalisée et consiste à appliquer la conversion des images numérisées en images en niveaux de gris allant de 0 à 255, le filtrage et le lissage. Il y a ensuite l'étape de la segmentation de l'image dont la méthode n'a pas été précisée. Des méthodes de pré-traitement ont été

FIGURE 15 – Échantillon des images de EMNIST Balanced.

appliquées pour réduire l'effet du bruit et pour augmenter la lisibilité de l'image d'entrée. La base de données au final est partitionnée en deux ensembles : un ensemble d'apprentissage constitué de 13440 images, soit 480 images par alphabet et un ensemble de test constitué de 3360 images soit 120 images par alphabet. Le dataset est disponible gratuitement sur le site de Kraggle [5]. Un échantillon de cette source de données est exposé dans la figure 17

6.1.3 MADBase

On désigne par le terme 'chiffres arabes' la variante de la présentation graphique du système de numération indo-arabe, utilisé par la partie orientale du monde arabe. La figure 18 illustre cette variante graphique dans l'écriture des chiffres arabes en Moyen Orient.

5. https ://www.kaggle.com/datasets/mloey1/ahcd1

FIGURE 16 – Échantillon des images de EMNIST Letters.

Arabic Digit	English Digit	Image
١	1	
٢	2	
٣	3	
٤	4	
٥	5	
٦	6	
٧	7	
٨	8	
٩	9	
٠	0	

FIGURE 18 – Le système de numération indo-arabe en Moyen-Orient[10].

FIGURE 17 – Échantillon des images de AHCD.

En tant que variante du système de numération indo-arabe, ces chiffres tirent leur origine des chiffres utilisés en Inde. C'est pourquoi, en arabe, on les appelle aussi chiffres indiens (hindi).

Sherif Abdelazeem et Ezzat El-Sherif [6], ont publié une base de données de chiffres manuscrits arabe (ADBase) et une version modifiée appelée (MADBase). ADBase et MADBase sont composés de 70 000 chiffres écrits par 700 scripteurs. La base de données est partitionnée en deux ensembles : un ensemble d'entrainement (60000 chiffres à 6000 images par classe) et un ensemble de test (10000 chiffres à 1000 images par classe). Le MADBase est une version modifiée de ADBase qui a le même format que MNIST. Ces deux bases de données ont servi de données référentielles pour de nombreux travaux dans la littérature scientifique [3, 6, 7, 11, 14, 36] Il sont des datasets importants dans le domaine de la reconnaissance automatique des chiffres arabes. MADBase est disponible et accessible gratuitement à la communauté des chercheurs. La figure 19 montre des exemples d'images d'entraînement et de test de la base MADBase que l'on utilise dans nos expérimentations.

6. https ://datacenter.aucegypt.edu/shazeem/

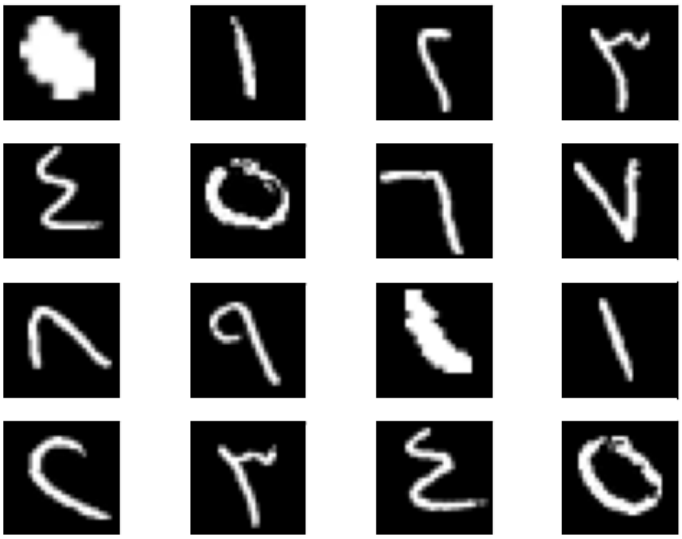


FIGURE 19 – Échantillon des images de MADBase.

6.2 Approche proposée

Notre approche vise à explorer l'approche hybride combinant CNN et SVM dans des datasets du manuscrit variés et hétérogènes qui sont illustrés dans la section précédente. Premièrement, nous expérimentons les deux classificateurs CNN et SVM avant d'appliquer notre approche hybride, qui prend CNN comme extracteur automatique de caractéristiques à partir d'images brutes et avant d'utiliser SVM dans la classification finale.

6.2.1 SVM

Initialement, SVM est appliqué sur l'ensemble des datasets mentionnés dans la section précédente. Le noyau gaussien (RBF) est utilisé par défaut. La fonction RBF du noyau SVM sur deux vecteurs de données x et x' est définie comme suit :

$$e^{-\gamma\|x-x'\|^2}$$

γ est un hyperparamètre de la fonction du noyau RBF que nous fixons a priori par la formule

suivante :

$$\gamma = n\text{Var}[X]$$

Var$[X]$ représente la variance du dataset d'entrainement X et n le nombre de variables aléatoires correspondant au nombre de pixels dans l'image d'entrée.

C et γ sont deux hyperparamètres de la SVM qui ont été détaillés dans la section dédiée à la SVM, comme mentionné dans la section 2.

Pour chaque dataset, nous répétons la classification SVM sous différentes valeurs de C, où C est choisi parmi les valeurs $1, 3, 5, 10$. Les variantes de SVM correspondantes sont étiquetées comme SVM_C, où C représente la valeur attribuée au paramètre C.

6.2.2 CNN

Nous allons appliquer différentes architectures de CNN étiquetées CNN_1, CNN_2, CNN_3, CNN_4 et CNN_5 pour chaque dataset. Les structures de ces architectures sont illustrées dans le tableau 3. Plus précisément :

- CNN_1 correspond à une structure simplifiée de CNN en 5 couches, telle que présentée et utilisée par Pan *et al.* [38].
- CNN_2 correspond à l'architecture LeNet-5 [33], avec comme seule différence que le nombre d'unités de sortie est ajusté pour correspondre au nombre de classes du dataset d'entrée.
- CNN_2 correspond à l'architecture LeNet-5 [33], avec comme seule différence que le nombre d'unités de sortie est ajusté pour correspondre au nombre de classes du dataset d'entrée.
- CNN_4 correspond à la même architecture de CNN proposée par El-Sawy *et al.* [23].
- CNN_5 est une variante de CNN_4 où le Dropout est appliqué sur les couches intermédiaires, et elle a été envisagée et appliquée pour certains datasets.

Nous présentons dans le tableau 3 les différents paramètres de CNN : le Kernel qui est une matrice correspondant au noyau de convolution ; le Stride fait référence au pas que le

TABLE 3 – Architectures CNN utilisées

Couche	CNN_1	CNN_2	CNN_3	CNN_4	CNN_5
1	INPUT	INPUT	INPUT	INPUT	INPUT
2	CONV2D filters :50 kernel :5x5 stride :2	CONV2D filters :6 kernel :5x5 stride :1	CONV2D filters :32 kernel :5x5 stride :1	CONV2D filters :80 kernel :5x5 stride :1	CONV2D filters :80 kernel :5x5 stride :1
	-	-	-	-	Dropout
3	CONV2D filters :50 kernel :5x5 stride :2	AveragePool	MaxPool stride :2	MaxPool stride :2	MaxPool stride :2
4	FC :100	CONV2D filters :16 kernel :5x5 stride :1	CONV2D filters :48 kernel :5x5 stride :1	CONV2D filters :64 kernel :5x5 stride :1	CONV2D filters :64 kernel :5x5 stride :1
	-	-	-	-	Dropout
5	FC :OUTPUT	AveragePool	MaxPool stride :2	MaxPool stride :2	MaxPool stride :2
6	—	FC :120	FC :256	FC :1024	FC :1024
7	—	FC :84	FC :84	-	-
8	—	FC :OUTPUT	FC :OUTPUT	FC :OUTPUT	FC :OUTPUT

noyau de convolution considère dans son parcours de la matrice d'entrée ; le Pooling dont l'objectif principal est de réduire le nombre d'éléments des matrices de la couche d'entrée ; le AveragePooling qui renvoie la moyenne de toutes les valeurs de la partie traitée par le noyau et l'insère dans la position correspondante de la matrice de sortie ; le MaxPooling qui renvoie la valeur maximale de la partie de la matrice traitée par le noyau ; la couche Fully connected (FC) qui sert à convertir la matrice d'entrée en vecteurs entièrement connectés comme un réseau de neurones.

Nous avons choisi les différentes architectures énumérées dans le tableau 3 en raison de leur efficacité démontrée dans de nombreux travaux de recherche pour les mêmes datasets que nous utilisons dans notre étude. Cela nous permettra de comparer nos résultats avec ceux de la littérature qui sont basés sur les mêmes architectures, afin d'évaluer dans quelle mesure le modèle hybride basé sur le CNN sous-jacent peut améliorer la précision par rapport à

l'architecture CNN sous-jacente.

Pour toutes les architectures mentionnées dans le tableau 3, le modèle est entraîné sur plusieurs epochs jusqu'à ce que le processus d'apprentissage converge, en suivant les bonnes pratiques de formation des CNN.

6.2.3 Hybride CNN-SVM

Les images normalisées et centrées sont utilisées comme couche d'entrée dans notre modèle Hybride CNN-SVM, comme illustré dans la figure 20. Dans ce modèle, la dernière couche du CNN est remplacée par un classifieur SVM. Le SVM prend en entrée les sorties de la couche "Fully connected" qui précède les sorties de décision du CNN pour l'entraînement par SVM. Ainsi, les décisions obtenues sont le résultat de cette approche hybride.

Toutes les combinaisons possibles entre CNN_i et SVM_j seront expérimentés. Ainsi, l'expérimentation permettra d'avoir le plus d'éléments de comparaison des résultats de Hybride CNN-SVM à ceux obtenus avec CNN et SVM utilisés chacun individuellement.

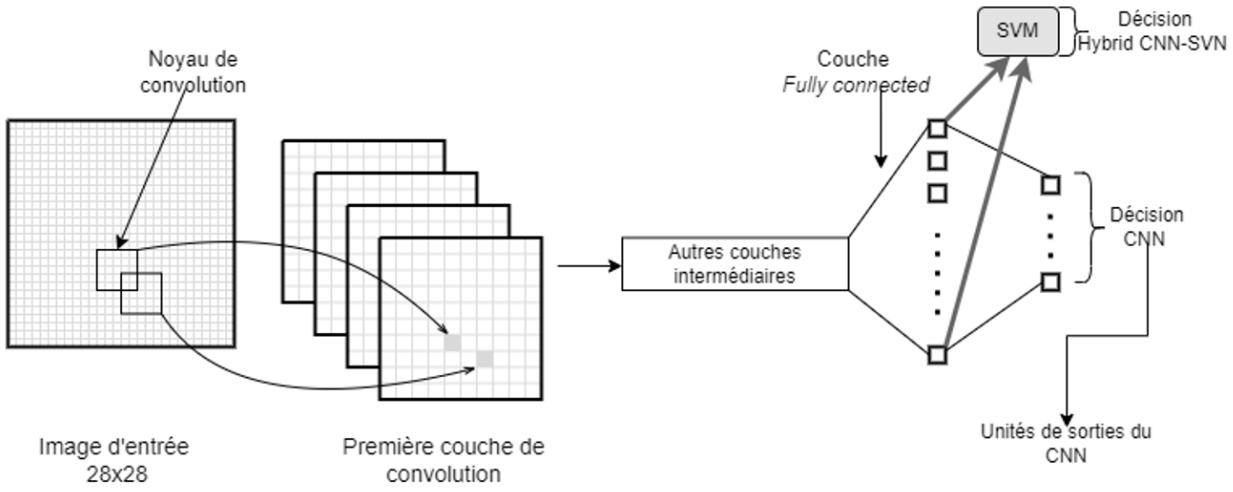


FIGURE 20 – Le modèle Hybride CNN-SVM.

En plus de rechercher des performances optimales et de déterminer la combinaison optimale dans l'approche hybride, qui sont des objectifs courants dans la littérature, notre travail vise à évaluer empiriquement si les performances de l'approche hybride sont statistiquement

significatives par rapport à l'utilisation individuelle du CNN et du SVM. Avant d'élaborer nos hypothèses, il est important de définir certaines notions :

— SVM_i désigne le modèle SVM ayant comme paramètre $C = i$ étant donné $i \in \{1, 3, 5, 10\}$.

— CNN_i désigne l'architecture CNN correspondante tel que illustré dans le tableau 3.

— HYBRID_i désigne l'approche hybride à base de CNN_i.

— $\text{HYBRID}_{i,j}$ désigne le modèle combinant CNN_i et SVM_j.

— $\mu_{\text{SVM}}(d)$ désigne la moyenne de taux de reconnaissance par SVM pour le dataset $\boldsymbol{d}$.

— $\mu_{\text{HYBRID}_i}(d)$ désigne la moyenne des résultats obtenus des taux de reconnaissance par l'approche hybride à base de CNN_i pour le dataset $\boldsymbol{d}$.

— $\mu_{\text{HYBRID}}(d)$ désigne la moyenne de taux de reconnaissance obtenus à travers toutes les combinaisons hybrides pour le dataset $\boldsymbol{d}$.

Pour chaque dataset d, les résultats obtenus à travers l'approche Hybride CNN-SVM et les deux approches sous-jacentes seront examinés afin de valider les hypothèses suivantes :

— H_1 **: L'approche Hybride CNN-SVM améliore de manière significative le taux de reconnaissance par rapport à l'utilisation de SVM seul ?**

  — $\text{H}_1 - 0 : \mu_{\text{HYBRID}}(d) = \mu_{\text{SVM}}(d)$

  — $\text{H}_1 - 1 : \mu_{\text{HYBRID}}(d) > \mu_{\text{SVM}}(d)$

— H_2 **: L'utilisation de SVM sur les résultats de l'extraction des caractéristiques à partir de CNN_i permet-elle d'améliorer le taux de reconnaissance par rapport à l'utilisation de SVM sur les ensembles de données sans extraction des caractéristiques ?**

  — $\text{H}_{2,i} - 0 : \mu_{\text{HYBRID}_i}(d) = \mu_{SVM}(d)$

  — $\text{H}_{2,i} - 1 : \mu_{\text{HYBRID}_i}(d) > \mu_{SVM}(d)$

— $\mathtt{H}_3$ **: L'approche hybride HYBRID$_i$, avec l'utilisation de SVM sur les résultats d'extraction de caractéristiques de CNN$_i$, présente-t-elle une augmentation statistiquement significative du taux de reconnaissance par rapport à l'utilisation de CNN$_i$ seul ?**

— $\mathtt{H}_{3,i} - 0 : \mu_{\mathrm{HYBRID}_i}(d) = \mu_{\mathrm{CNN}_i(d)}$

— $\mathtt{H}_{3,i} - 1 : \mu_{\mathrm{HYBRID}_i}(d) > \mu_{\mathrm{CNN}_i(d)}$

Nous utiliserons le test t de Welch (Welch's t-test) pour valider les hypothèses $\mathtt{H}_1$ et $\mathtt{H}_3$. En revanche, pour l'élaboration de l'hypothèse $\mathtt{H}_2$, nous utiliserons le test de comparaison d'observations pairées. Un niveau de signification alpha de 0.05 sera considéré par défaut pour la vérification de nos hypothèses. Cependant, nous évaluerons les hypothèses énumérées en fonction des p-values obtenues pour chaque dataset. Enfin, nous terminerons notre analyse par une étude comparative des résultats obtenus avec ceux de la littérature pour les différents datasets.

6.3 Conclusion

En somme, nous avons utilisé trois collections de datasets, à savoir EMNIST, Arabic Handwritten Characters Dataset (AHCD) et MADBase, pour mener nos recherches. Nous avons d'abord réalisé la classification des images manuscrites de ces datasets en utilisant des SVM avec une variation du paramètre C. Ensuite, nous avons utilisé différentes architectures de CNN pour effectuer la classification. Enfin, nous avons combiné nos modèles SVM et CNN dans une approche Hybride CNN-SVM pour tous nos datasets. Pour valider notre hypothèse sur l'apport de l'approche hybride, nous avons utilisé le test t de Welch et le test de comparaison d'observations pairées. Nous procéderons ensuite à une étude comparative des résultats obtenus avec ceux de la littérature pour chaque type de manuscrit étudié.

Chapitre 7 Expérimentations et discussion

Pour les besoins de l'expérimentation, nous avons utilisé les librairies de Keras [7] et LibSVM [8].

Les combinaisons hybrides sont formées par le biais de nombreux paramètres variés tant pour CNN que SVM constitutifs. Les modèles CNN_1, CNN_2 et CNN_3 ont été expérimentés avec leurs architectures hybrides les impliquant sur l'ensemble des jeux de données. L'expérimentation sur AHCD et MADBase a été étendue pour impliquer CNN_4 et CNN_5 avec les approches hybrides associées. CNN_5 a été expérimentée sur EMNIST Balanced parmi le reste des sous-ensembles de EMNIST.

En observant les résultats expérimentaux affichés dans le tableau 5, nous constatons que les valeurs de variance σ^2 pour les approches hybrides combinées sont généralement plus élevées par rapport aux variances des taux des approches hybrides regroupées selon l'architecture CNN sous-jacente commune. Les taux obtenus par les approches hybrides ont tendance à se regrouper en fonction de l'architecture CNN utilisée. De plus, les taux issus des approches hybrides sont davantage influencés par les architectures CNN sous-jacentes que par les paramètres de SVM.

Cela nous permet de considérer la moyenne du groupe HYBRID_i dans l'évaluation comparative avec les SVM et les CNN sous-jacents. Cette approche nous permettra d'évaluer de manière individuelle l'avantage du modèle hybride par rapport à SVM et CNN sous-jacent pour chaque modèle, à travers les hypothèses $H_{2,i}$ et $H_{3,i}$ autour de HYBRID_i

7. https ://keras.io/
8. https ://www.csie.ntu.edu.tw/ cjlin/libsvm/

TABLE 4 – Les taux de reconnaissance en (%) par SVM et CNN sur les bases de test.

Technique	SVM_1	SVM_3	SVM_5	SVM_{10}	μ_{SVM}	σ^2	CNN_1	CNN_2	CNN_3	CNN_4	CNN_5
MNIST	98,34	98,73	98,76	93,60	97,36	6,31	99,10	98,96	99,37	-	-
Letters	88,98	90,10	90,29	90,27	89,91	0,39	90,50	90,97	91,53	-	-
Balanced	84,28	85,52	85,62	85,37	85,20	0,38	85,38	86,38	86,65	-	88,96
By_Merge	84,19	85,54	85,67	85,55	85,24	0,49	89,30	89,88	89,47	-	-
By_Class	82,36	83,38	83,32	-	83,65	1,82	85,42	86,08	85,64	-	-
AHCD	65,29	71,36	72,08	72,23	70,24	11,03	82,70	77,79	84,40	93,69	95,23
MADBase	98,45	98,62	98,68	98,67	98,60	0,01	98,65	98,71	99,1	99,07	99,17

7.1 Caractères latins : EMNIST

Les résultats obtenus pour les chiffres manuscrits de MNIST (voir tableau 4) montrent que la variance des taux de reconnaissance obtenus par SVM, $\sigma^2 = 6,31$, est élevée par rapport aux autres datasets de EMNIST. Cette variance élevée est principalement due au résultat obtenu par SVM_{10}, qui présente un taux de reconnaissance de 93,6%, nettement inférieur à la moyenne μ_{SVM}(MNIST). Cependant, si on exclut le taux obtenu par SVM_{10} dans l'analyse des hypothèses, cela n'affecte pas la conclusion sur l'avantage du modèle hybride par rapport à SVM pour MNIST.

Le taux issu de SVM_{10} est le plus bas parmi les taux obtenus dans nos expérimentations pour MNIST, donc son exclusion ne biaise pas la conclusion en faveur de SVM. En effet, en excluant SVM_{10}, on obtient une moyenne des taux de reconnaissance pour SVM de $\mu_{SVM}(\text{MNIST}) = 98.61\%$ avec une variance de $\sigma^2 = 0,055$.

D'autre part, la moyenne des taux de reconnaissance pour les approches hybrides combinées, $\mu_{HYBRID}(\text{MNIST}) = 99,29\%$ (voir tableau 5), présente une augmentation de 0,68% par rapport au taux obtenu par SVM en excluant SVM_{10}. Cette petite amélioration a permis de valider l'hypothèse H_1 pour MNIST, comme indiqué dans le tableau 6, avec une p-value de 0,01.

En ce qui concerne les autres datasets constitutifs de EMNIST, l'approche hybride est clairement avantageuse par rapport à SVM. Les résultats globaux des approches hybrides affichés dans le tableau 5 montrent que les moyennes μ_{HYBRID} sont nettement supérieures aux taux

TABLE 5 – Les taux de reconnaissance en (%) par Hybride CNN-SVM sur les bases de test.

Technique	MNIST	Letters	Balanced	By_Merge	By_Class	AHCD	MADBase
$\text{HYBRID}_{1,1}$	99,30	92,38	87,89	89,81	86,82	87,58	98,96
$\text{HYBRID}_{1,3}$	99,27	92,32	87,71	89,77	86,38	89,58	98,89
$\text{HYBRID}_{1,5}$	99,20	92,23	87,49	89,64	86,17	89,70	98,86
$\text{HYBRID}_{1,10}$	99,18	92,24	87,39	89,45	85,85	89,31	98,88
μ_{HYBRID_1}	99,24	92,29	87,62	89,67	86,30	89,53	98,90
σ^2	0,003	0,005	0,05	0,026	0,16	0,04	0,002
$\text{HYBRID}_{2,1}$	99,16	92,2	87,87	89,87	86,37	80,47	98,82
$\text{HYBRID}_{2,3}$	99,23	92,22	88,09	89,90	86,40	83,69	98,82
$\text{HYBRID}_{2,5}$	99,20	92,25	88,08	89,87	86,39	84,34	98,85
$\text{HYBRID}_{2,10}$	99,21	92,27	88.00	89,75	86,25	84,52	98,87
μ_{HYBRID_2}	99,20	92,23	88,01	89,85	86,35	84,18	98,84
σ^2	0,001	0,001	0,01	0,004	0,005	0,19	0,006
$\text{HYBRID}_{3,1}$	99,43	92,79	88,22	89,91	86,56	90,65	99,16
$\text{HYBRID}_{3,3}$	99,47	92,84	88,03	89,82	86,34	91,51	99,10
$\text{HYBRID}_{3,5}$	99,30	92,38	99,47	92,66	87,91	91,60	99,09
$\text{HYBRID}_{3,10}$	99,42	92,58	87,75	89.60	86,01	91,72	99,09
μ_{HYBRID_3}	99,45	92,71	87,97	89,77	86,28	91,61	99,11
σ^2	0,001	0,015	0,04	0,017	0,053	0,01	0,001
$\text{HYBRID}_{4,1}$	-	-	-	-	-	94,64	99,15
$\text{HYBRID}_{4,3}$	-	-	-	-	-	94,70	99,16
$\text{HYBRID}_{4,5}$	-	-	-	-	-	94,55	99,14
$\text{HYBRID}_{4,10}$	-	-	-	-	-	94,43	99,14
μ_{HYBRID_4}	-	-	-	-	-	94,56	99,15
σ^2	-	-	-	-	-	0,02	9,16E-5
$\text{HYBRID}_{5,1}$	-	-	89,93	-	-	96,63	99,32
$\text{HYBRID}_{5,3}$	-	-	89,77	-	-	97,00	99,35
$\text{HYBRID}_{5,5}$	-	-	89,63	-	-	96,34	99,35
$\text{HYBRID}_{5,10}$	-	-	89,34	-	-	96,90	99,33
μ_{HYBRID_5}	-	-	89,67	-	-	96,74	99,34
σ^2	-	-	0,063	-	-	0,12	0.0002
μ_{HYBRID}	99,29	92,41	88,32	89,76	86,31	91,32	99,06
σ^2	0,012	0,056	0,7	0,01	0,06	20,23	0,034

obtenus par SVM (voir tableau 4) pour tous les datasets constitutifs de EMNIST. Cependant, l'amélioration par rapport à SVM varie selon le dataset utilisé, étant plus marquée pour les datasets plus complexes en termes de nombre de classes.

Pour Letters, l'approche hybride combinée a permis d'obtenir une moyenne des taux de reconnaissance de 92,41%, ce qui représente une augmentation de 2,5% par rapport à μ_{SVM}. Pour Balanced, l'augmentation est de 3,12% par rapport à μ_{SVM}. Pour By_Merge, l'aug-

TABLE 6 – Les seuils observés relativement aux hypothèses étudiées.

	MNIST	Letters	Balanced	By_Merge	By_Class	AHCD	MADBase
$\mathtt{H}_1$:	**0,01***	0,002	0,0002	0,0005	0,005	2,26E-5	0,0001
$\mathtt{H}_{2,1}$:	0,02	0,003	0,005	0,0008	0,001	0,0002	0,015
$\mathtt{H}_{2,2}$:	0,02	0,002	0,001	0,0005	0,005	1,36E-06	0,007
$\mathtt{H}_{2,3}$:	0,01	0,002	0,003	0,0007	0,005	5,95E-05	0,002
$\mathtt{H}_{2,4}$:	-	-	-	-	-	0,0001	0,001
$\mathtt{H}_{2,5}$:	-	-	0,0008	-	-	0,0001	0,0003
$\mathtt{H}_{3,1}$:	0,006	8,04E-06	0,004	0,01	0,01	0,0001	0,0007
$\mathtt{H}_{3,2}$:	0,0002	2,04E-06	0,001	**0,2**	0,002	0,0007	0,0009
$\mathtt{H}_{3,3}$:	0,0006	0,00014	0,003	0,009	0,005	3,55E-05	**0,29**
$\mathtt{H}_{3,4}$:	-	-	-	-	-	0,004	0,0002
$\mathtt{H}_{3,5}$:	-	-	0,0007	-	-	0,009	9,8E-5

Note : *$\mathtt{H}_1$ pour MNIST est abordée sans considérer le résultat de SVM_{10}

mentation est encore plus significative, avec un écart de 4,52% par rapport à μ_{SVM}. Ces résultats sont corroborés par les valeurs de seuils observées, qui confirment la validation. Ceci est étayé par des seuils de validation hautement probants pour la confirmation de $\mathtt{H}_1$ dans tous les ensembles de données de EMNIST, comme illustré de manière concluante dans le tableau 6.

Nous pouvons observer que l'application du SVM sur les résultats de l'extraction issus de CNN entraîne une précision nettement supérieure par rapport à l'application du SVM sur des ensembles de données sans extraction de caractéristiques, et ce, indépendamment de l'architecture CNN utilisée, y compris pour toutes les architectures hybrides expérimentées. Cette constatation est étayée par un test d'hypothèses statistiques sur les $\text{H}_{2,i}$, qui révèle des seuils observés significatifs et inférieurs à $\alpha = 0,05$, comme illustré dans le tableau 6, pour toutes les approches HYBRID_i.

Les résultats affichés dans les tableaux 4 et 5 montrent que les modèles hybrides ont tendance à augmenter de manière significative le taux de reconnaissance par rapport au CNN sous-jacent. En particulier, l'architecture CNN_3 a donné les meilleurs taux de reconnaissance parmi les trois architectures CNN expérimentées sur les ensembles de données MNIST, Let-

ters et Balanced.

Lorsqu'elle est combinée avec le SVM, l'architecture HYBRID_3, qui utilise CNN_3 comme base, obtient des résultats encore meilleurs. Par exemple, pour l'ensemble de données MNIST, comme illustré dans le tableau 5, $\text{HYBRID}_{3,3}$ atteint un taux de reconnaissance de 99,47%, dépassant de 0,10% le taux de 99,37% obtenu par CNN_3. Cette amélioration est encore plus marquée pour des ensembles de données plus complexes, tels que Letters et Balanced. Pour Letters, $\text{HYBRID}_{3,2}$ permet une amélioration de 1,31% par rapport à CNN_3, tandis que pour Balanced, $\text{HYBRID}_{3,1}$ offre une amélioration de 1,57% par rapport à CNN_3.

Les résultats des tests d'hypothèses statistiques $\text{H}_{3,i}$ confirment de manière significative que l'approche hybride, par rapport au CNN sous-jacent, conduit à une augmentation statistiquement significative du taux de reconnaissance, indépendamment du SVM utilisé, avec des niveaux de signification inférieurs à $0,05$ pour tous les ensembles de données EMNIST. De plus, empiriquement, il est observé que pour chaque architecture hybride HYBRID_i impliquant l'architecture CNN_i, la moyenne des taux de reconnaissance obtenus est nettement supérieure à celle du CNN_i sous-jacent.

7.2 Caractères arabes

7.2.1 Lettres arabes (AHCD)

Les résultats obtenus par SVM pour AHCD, comme indiqué dans le tableau 4, varient de 65,29% à 72,23%. En utilisant SVM, nous avons obtenu un taux de reconnaissance moyen de $\mu_{\text{SVM}}(\text{AHCD}) = 70,24\%$ avec une variance de $\sigma^2 = 11,03$. La valeur élevée de la variance σ^2 des taux obtenus par SVM pour AHCD s'explique par le fait que SVM_1 présente un taux nettement inférieur à la moyenne, soit 65,29

La moyenne des taux de reconnaissance des approches hybrides combinées pour AHCD, comme montré dans le tableau 5, est de $\mu_{\text{HYBRID}}(\text{AHCD}) = 91,32\%$ avec une variance de $\sigma^2 = 20,23$. La variance élevée $\sigma^2 = 20,23$ pour les approches hybrides combinées s'explique par le fait que ces combinaisons sont formées à partir de nombreux paramètres variés pour

les CNN et SVM constitutifs. L'expérimentation sur AHCD a impliqué l'utilisation de 5 architectures de CNN, ce qui a influencé grandement les variations des taux de reconnaissance obtenus par les approches hybrides. Il est intéressant de noter que les taux obtenus par les modèles hybrides tendent à se regrouper selon l'architecture CNN utilisée. En observant les valeurs de variance σ^2 regroupées selon l'architecture CNN employée, on constate qu'elles oscillent entre 0,01 et 0,20, ce qui indique une baisse significative de la variance pour chaque HYBRID_i.

Les approches hybrides présentent un avantage significatif par rapport à SVM, comme le montrent les résultats globaux. Les taux de reconnaissance des approches hybrides HYBRID_i varient de 84,18% à 96,74% (voir tableau 5), tandis que les taux de SVM varient de 65,29% à 72,23% (voir tableau 4). L'augmentation du taux de reconnaissance par les approches hybrides par rapport à SVM pour AHCD est donc très significative. Cela est confirmé par le test d'hypothèse H_1, qui montre un niveau de signification nettement inférieur à 0,05, comme illustré dans le tableau 6.

L'application de SVM sur les résultats d'extraction issus de CNN permet d'obtenir une précision nettement supérieure par rapport à SVM appliqué sur des jeux de données sans extraction de caractéristiques. Cette constatation est valable indépendamment de l'architecture CNN utilisée, notamment pour toutes les architectures hybrides expérimentées. Cela est confirmé par les tests d'hypothèses statistiques sur les $H_{2,i}$, qui montrent des niveaux de signification inférieurs à 0,05, comme illustré dans le tableau 6, pour toutes les approches HYBRID_i.

Nos résultats montrent également que les modèles hybrides augmentent de manière significative les taux de reconnaissance par rapport aux CNN sous-jacents. En effet, nous observons une augmentation moyenne de 6,34% pour HYBRID_1, de 5,46% pour HYBRID_2, de 6,97% pour HYBRID_3, de 0,89% pour HYBRID_4, et de 1,48% pour HYBRID_5, par rapport aux CNN sous-jacents. Empiriquement, pour chaque architecture HYBRID_i impliquant CNN_i, la moyenne des taux obtenus est nettement supérieure à celle du CNN_i sous-jacent. De plus, l'augmentation du taux de reconnaissance par l'approche hybride par rapport au CNN sous-jacent

est statistiquement significative, indépendamment du SVM utilisé. Cela est confirmé par les tests d'hypothèses statistiques sur les $H_{3,i}$, qui montrent des niveaux de signification très faibles et inférieurs à 0,05.

7.2.2 Chiffres arabes (MADBase)

Pour les chiffres arabes, nous avons obtenu un taux de reconnaissance moyen μ_{SVM} de $98,60\%$ avec une variance $\sigma^2 = 0,011$. En revanche, la moyenne des taux de reconnaissance pour les approches hybrides combinées $\mu_{\text{HYBRID}}(\text{MADBase})$ est de $99,06\%$ avec une variance $\sigma^2 = 0,034$ (voir tableau 5). La faible valeur de la variance, tant pour les modèles hybrides combinés que pour SVM, s'explique par le fait que les variations des paramètres n'ont pas semblé affecter de manière significative le taux de reconnaissance final pour MADBase.

Il est possible de constater l'avantage des approches hybrides par rapport à SVM à travers nos résultats. Même si la différence de moyennes entre $\mu_{\text{HYBRID}}(\text{MADBase})$ et $\mu_{\text{SVM}}(\text{MADBase})$ est minime, elle est suffisante pour conclure que l'augmentation du taux de reconnaissance par l'approche hybride par rapport à SVM pour MADBase est statistiquement significative. Cela est confirmé par le test statistique d'hypothèse H_1, qui fournit un niveau de signification (0,001) nettement inférieur à 0,05 pour MADBase, comme illustré dans le tableau 6. De plus, ce constat est valable indépendamment de l'architecture CNN utilisée, pour toutes les architectures hybrides expérimentées. Ceci est également confirmé par les tests d'hypothèses statistiques sur les $H_{2,i}$, qui fournissent des niveaux de signification inférieurs à 0,05 pour toutes les approches hybrides, comme illustré dans le tableau 6.

Nous pouvons observer à travers les résultats présentés dans les tableaux 4 et 5 que les modèles hybrides améliorent généralement les taux de reconnaissance par rapport au CNN sous-jacent. En moyenne, nous constatons une augmentation de 0,16% par rapport au CNN sous-jacent. Plus spécifiquement, nous observons une augmentation moyenne de 0,25% pour Hybrid_1, de 0,13% pour Hybrid_2, de 0,01% pour Hybrid_3, de 0,078% pour Hybrid_4, et de 0,17% pour Hybrid_5, par rapport au CNN sous-jacent.

Il convient de noter que CNN_5 affiche le meilleur taux de reconnaissance de 99,17% parmi

toutes les autres architectures de CNN, comme indiqué dans le tableau 4. Cependant, lorsque cette même architecture est combinée avec le SVM dans le modèle HYBRID_5, elle donne un résultat encore meilleur parmi toutes les combinaisons possibles et expérimentées, avec un taux moyen de 99,34% (voir tableau 5).

Le modèle HYBRID_3 n'a pas pu démontrer son avantage par rapport à l'approche CNN_3 selon le test statistique $H_{3,3}$ tel qu'illustré dans le tableau 6. Bien que HYBRID_3 affiche un taux de reconnaissance moyen de 99,11% (voir tableau 5), cela ne diffère que de 0,01% par rapport à CNN3 (voir tableau 4) qui a donné un taux de 99,10%. Cette augmentation n'a pas permis de valider l'hypothèse $H_{3,3}$ concernant la signification de l'augmentation par HYBRID_3 par rapport à CNN_3.

Cependant, pour les autres approches hybrides, les tests statistiques ont montré de manière concluante l'avantage de l'approche hybride par rapport au CNN sous-jacent (voir tableau 6). Empiriquement, pour chaque architecture hybride HYBRID_i impliquant CNN_i, la moyenne des taux obtenus est supérieure au taux obtenu par CNN_i sous-jacent. Bien que cette augmentation puisse être minime, elle est statistiquement significative en raison de la faible valeur de la variance pour les HYBRID_i. Ainsi, l'augmentation du taux de reconnaissance à travers l'approche hybride comparativement au CNN sous-jacent est statistiquement significative, indépendamment de SVM utilisé.

Les taux de reconnaissance des chiffres manuscrits arabes suivent le même schéma que les résultats obtenus sur MNIST. Les taux de reconnaissance les plus élevés sont observés pour les chiffres manuscrits, qu'ils soient en variante latine ou arabe. Cela s'explique par le fait que ces datasets ont un nombre de classes plus faible par rapport aux datasets impliquant des alphabets et des structures plus complexes.

7.3 Synthèse

Les taux de reconnaissance les plus élevés pour MNIST et MADBase sont observés par rapport aux autres ensembles de données. Cette tendance s'explique par le fait que MNIST et MADBase sont des ensembles de données moins complexes en termes de nombre de classes.

TABLE 7 – Tableau comparatif des résultats pour EMNIST

Référence	Technique	By_Class	By_Merge	Balanced	Letters	Digits
Cohen *et al.* [16]	Classifieur linéeaire	51,80%	50,51%	50,93%	55,78%	84,70%
Cohen *et al.* [16]	OPIUM	69,71%	72,57%	78,02%	85,15%	95,90%
Ghadekar *et al.* [25]	DWT-DCT + SVM				89,51%	97,74%
Dufourq et Bassett [21]	EDEN			88,3%	89,51%	99,3%
Biswas et Islam [13]	CNN					99,53
Cavalin et Oliveira [15]	CNN			87,18%	93,63%	99.46%
Ahlawat et Choudhary [1]	CNN					97,85%
Niu et Suen [37]	CNN					99,81%
Notre travail	Hybride CNN-SVM	86,82%	89,9%	89,93%	92,84%	99,47%

TABLE 8 – Tableau comparatif des résultats pour les caractères arabes

Référence	Dataset	Méthode	Test
Ashiquzzaman et Tushar [10]	CMATERDB 3.3.1	MLP	97,4
Pan *et al.* [38]	CENPARMI	CNN	99,22
Alkhawaldeh [6]	ADBase	LeNet + LSTM	98,92
Alkhateeb [5]	ADBase	CNN	94,3
Can et Kabadayı [14]	ADBase	Autoencoder et Softmax	99,34
Loey *et al.* [36]	MADBase	Autoencoder et Softmax	95,5
El-Sawy *et al.* [22]	MADBase	CNN Net-5	88
Notre approche	**MADBase**	**$\mathbf{CNN}_5$**	**99,17**
Notre approche	**MADBase**	**$\mathbf{HYBRID}_{5,3}$**	**99.35**
El-Sawy *et al.* [23]	AHCD database	CNN	94,9
Shams *et al.* [41]	AHCD database	DCNN-SVM	95,07
Altwaijry et Al-Turaiki [8]	AHCD database	CNN	94,9
Notre approche	**AHCD database**	**$\mathbf{CNN}_5$**	**95,23**
Notre approche	**AHCD database**	**$\mathbf{HYBRID}_{5,3}$**	**97,00**

De manière générale, les résultats globaux montrent clairement l'avantage des approches hybrides par rapport à SVM. Cette conclusion est confirmée par les valeurs des seuils observés dans les tests d'hypothèses H_1 et $H_{2,i}$, qui fournissent des niveaux de signification concluants, comme illustré dans le tableau 6, pour tous les ensembles de données étudiés. Cet avantage est particulièrement souligné dans le cas de l'application sur AHCD, où l'écart moyen des taux entre SVM et les approches hybrides combinées est plus notable. En conclusion, il est empiriquement évident que l'intérêt des approches hybrides par rapport à SVM est plus marqué lorsque les ensembles de données sont complexes en termes de nombre de classes.

L'augmentation statistiquement significative du taux de reconnaissance par l'approche hybride par rapport au CNN sous-jacent est indépendante du SVM utilisé, comme confirmé par les valeurs des seuils observés dans les tests d'hypothèses $H_{3,i}$ qui sont concluantes. Selon nos expérimentations, l'approche Hybride CNN-SVM est empiriquement avantageuse, car elle tend à produire de meilleurs résultats que le CNN et le SVM pris séparément.

Pour les ensembles de données MNIST et Letters, nous observons que CNN_3 donne les meilleurs taux de reconnaissance parmi les modèles CNN. De plus, l'approche hybride impliquant CNN_3 produit également les meilleurs taux de reconnaissance pour ces mêmes ensembles de données. Un constat similaire est observé pour CNN_5 lorsqu'il est appliqué sur AHCD, MADBase et Balanced. Cependant, l'approche hybride HYBRID_5 impliquant CNN_5 produit significativement les meilleurs taux de reconnaissance pour ces ensembles de données.

Empiriquement, pour chaque architecture hybride HYBRID_i impliquant CNN_i, la moyenne des taux obtenus est nettement supérieure à celle du CNN sous-jacent CNN_i. Cela démontre que les performances du modèle hybride varient en fonction des paramètres de ses composantes, c'est-à-dire le CNN et le SVM.

Outre l'évaluation intrinsèque des résultats qui a mis en évidence les avantages de l'approche hybride, nos taux de reconnaissance obtenus sont compétitifs par rapport aux résultats de la littérature, comme illustré dans les tableaux 7 et 8.

Pour l'ensemble de données EMNIST, Biswas et Islam [13] ont obtenu un taux de reconnaissance de 99,53% pour les chiffres manuscrits MNIST en utilisant CNN. Pour Letters, Cavalin et Oliveira [15] ont obtenu un taux de reconnaissance de 93,63% en utilisant CNN. Nos meilleurs résultats avec l'approche Hybride CNN-SVM sont légèrement inférieurs à ces résultats. Cependant, lorsque notre approche hybride est appliquée à un ensemble de données Balanced plus large et plus hétérogène, nous obtenons un meilleur résultat que celui de Cavalin et Oliveira, avec un taux de reconnaissance de 88,22% par rapport à leur taux de 87,18%. L'application sur EMNIST nous a permis de consolider l'avantage de notre approche hybride qui a pu maintenir sa performance en termes de taux de reconnaissance par rapport

à CNN et à SVM.

L'avantage de notre approche hybride est encore mieux mis en évidence par une évaluation comparative avec les résultats de certains travaux de la littérature menés sur le même jeu de données AHCD, comme illustré dans le tableau 8. Le taux de reconnaissance obtenu avec notre architecture CNN (CNN_4), qui correspond à la même architecture utilisée par El-Sawy *et al.* [23], est de 93,69%, soit inférieur au taux rapporté par El-Sawy *et al.*. Cependant, en appliquant le Dropout aux couches intermédiaires du CNN dans notre architecture CNN_5, nous avons obtenu un taux de reconnaissance de 95,23%, soit supérieur à celui obtenu avec CNN_4 et rapporté par El-Sawy *et al.*. Cela met en avant l'avantage d'utiliser le Dropout dans l'architecture du CNN.

Notre meilleure architecture CNN_5, combinée avec SVM, a permis d'améliorer le taux de reconnaissance de 1,78%, atteignant ainsi un taux de 97% avec notre approche hybride $HYBRID_{5,3}$ (voir tableau 5). Ce taux dépasse de manière significative les taux rapportés par El-Sawy *et al.* [23], Altwaijry et Al-Turaiki [8] et Shams *et al.* [41]. En effet, Shams *et al.* ont expérimenté une approche hybride combinant CNN et SVM, étiquetée DCNN-SVM, et ont rapporté un taux de reconnaissance de 95,07%, qui est inférieur à notre taux de 97%.

Nous avons démontré, à travers notre approche hybride, que les résultats obtenus pour MADBase sont nettement supérieurs à ceux rapportés par El-Sawy *et al.* [22] et Loey *et al.* [36], comme illustré dans le tableau 8. Dans leurs travaux sur la reconnaissance des chiffres arabes manuscrits, El-Sawy *et al.* ont utilisé la variante LeNet-5 de CNN et ont obtenu un taux de reconnaissance de 88%. De même, Loey *et al.* [36] ont utilisé un Autoencodeur et ont obtenu un taux de reconnaissance de 95,5%. En revanche, notre approche hybride $HYBRID_{5,3}$ (voir tableau 5) a surpassé les résultats de El-Sawy *et al.* et Loey *et al.* pour le même dataset MADBase. Globalement, notre approche Hybride CNN-SVM s'est avérée empiriquement avantageuse pour la reconnaissance des chiffres arabes, en surpassant à la fois les résultats de CNN et SVM, ainsi que les résultats rapportés par El-Sawy *et al.* et Loey *et al.* pour le même dataset MADBase.

Chapitre 8 Conclusion et perspectives

En conclusion, ce mémoire a pour objectif de présenter une nouvelle approche dans le domaine de la classification, et de l'explorer empiriquement dans le contexte de la reconnaissance du manuscrit. Le modèle Hybride CNN-SVM proposé combine l'utilisation de CNN pour l'extraction des caractéristiques et SVM comme classifieur final. Notre étude aborde le modèle sous deux perspectives : sa capacité à reconnaître les caractères manuscrits et sa performance en termes de taux de reconnaissance comparativement à CNN et SVM.

Nous avons réalisé nos expérimentations sur des datasets référentiels comprenant des chiffres et des lettres, largement utilisés et bien documentés dans la littérature. Nos sources de données sont des images manuscrites contenant à la fois des chiffres et des lettres, dans les deux types d'écritures, à savoir le latin et l'arabe.

Nos résultats expérimentaux ont révélé que les performances du modèle hybride, composé du CNN et du SVM, varient en fonction des paramètres de ces composantes. Parmi toutes les combinaisons expérimentées, notre approche Hybride CNN-SVM a démontré les meilleurs taux de reconnaissance pour l'ensemble de nos datasets, comparativement aux méthodes sous-jacentes que sont CNN et SVM.

Notre modèle Hybride CNN-SVM offre des performances de reconnaissance compétitives par rapport à la littérature pour les datasets étudiés. Cependant, notre contribution se démarque principalement dans l'amélioration du taux de reconnaissance pour les caractères manuscrits arabes, un domaine moins exploré que celui des caractères latins. Notre approche a permis d'atteindre la meilleure précision pour les chiffres arabes, avec un taux de reconnaissance de 99,35%, surpassant ainsi les résultats rapportés dans les travaux récents. Pour les lettres arabes (AHCD), notre approche a également significativement amélioré le taux de classification, atteignant un taux de 97%, ce qui est nettement supérieur aux taux rapportés dans les récents travaux. Cela confirme l'avantage empirique de notre approche.

L'application de notre approche Hybride CNN-SVM sur des datasets variés a permis de conclure qu'elle est une méthode prometteuse dans le domaine de la classification en général. Cependant, nous avons également constaté que ce modèle hybride s'est révélé particulièrement pertinent et efficace dans le domaine de la reconnaissance de l'écriture manuscrite.

Néanmoins, il est essentiel d'étendre l'architecture proposée afin de pouvoir traiter des mots manuscrits dans différentes langues et ainsi améliorer le taux de reconnaissance. Cela pourrait impliquer l'ajout de nouvelles couches ou de nouvelles techniques d'extraction de caractéristiques dans le modèle CNN, ainsi que l'exploration de paramètres optimaux pour le SVM, tels que le choix du noyau et les paramètres de régularisation.

En somme, il existe plusieurs pistes d'amélioration potentielles pour l'architecture proposée, notamment l'extension pour traiter des mots manuscrits dans différentes langues, l'optimisation des architectures CNN et SVM, et l'exploration de nouvelles techniques d'apprentissage automatique. Ces améliorations pourraient contribuer à améliorer le taux de reconnaissance du modèle et à renforcer sa pertinence dans le domaine de la reconnaissance de l'écriture manuscrite.

Références

[1] S. Ahlawat et A. Choudhary : Hybrid cnn-svm classifier for handwritten digit recognition. *Procedia Computer Science*, 167:2554–2560, 2020.

[2] Y. Al-Ohali, M. Cheriet et C. Suen : Databases for recognition of handwritten arabic cheques. *Pattern Recognition*, 36(1):111–121, 2003.

[3] E. Al-Wajih et R. Ghazali : An enhanced lbp-based technique with various size of sliding window approach for handwritten arabic digit recognition. *Multimedia Tools and Applications*, 80(16):24399–24418, 2021.

[4] A. A. A. Ali et S. Mallaiah : Intelligent handwritten recognition using hybrid cnn architectures based-svm classifier with dropout. *Journal of King Saud University-Computer and Information Sciences*, 34(6):3294–3300, 2022.

[5] J. H. Alkhateeb : Handwritten arabic digit recognition using convolutional neural network. *International Journal of Communication Networks and Information Security*, 12(3):411–416, 2020.

[6] R. S. Alkhawaldeh : Arabic (indian) digit handwritten recognition using recurrent transfer deep architecture. *Soft Computing*, 25(4):3131–3141, 2021.

[7] R. S. Alkhawaldeh, M. Alawida, N. F. F. Alshdaifat, W. Alma'aitah et A. Almasri : Ensemble deep transfer learning model for arabic (indian) handwritten digit recognition. *Neural Computing and Applications*, 34(1):705–719, 2022.

[8] N. Altwaijry et I. Al-Turaiki : Arabic handwriting recognition system using convolutional neural network. *Neural Computing and Applications*, 33(7):2249–2261, 2021.

[9] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie et L. Farhan : Review of deep learning : Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8(1):1–74, 2021.

[10] A. Ashiquzzaman et A. K. Tushar : Handwritten arabic numeral recognition using deep learning neural networks. *In 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, p. 1–4. IEEE, 2017.

[11] H. M. Balaha, H. A. Ali et M. Badawy : Automatic recognition of handwritten arabic characters : a comprehensive review. *Neural Computing and Applications*, 33 (7):3011–3034, 2021.

[12] A. Baldominos, Y. Saez et P. Isasi : A survey of handwritten character recognition with mnist and emnist. *Applied Sciences*, 9(15):3169, 2019.

[13] A. Biswas et M. S. Islam : An efficient cnn model for automated digital handwritten digit classification. *Journal of Information Systems Engineering and Business Intelligence*, 7(1):42–55, 2021.

[14] Y. S. Can et M. E. Kabadayi : Automatic cnn-based arabic numeral spotting and handwritten digit recognition by using deep transfer learning in ottoman population registers. *Applied Sciences*, 10(16):5430, 2020.

[15] P. Cavalin et L. Oliveira : Confusion matrix-based building of hierarchical classification. *In Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications : 23rd Iberoamerican Congress, CIARP 2018, Madrid, Spain, November 19-22, 2018, Proceedings 23*, p. 271–278. Springer, 2019.

[16] G. Cohen, S. Afshar, J. Tapson et A. Van Schaik : Emnist : Extending mnist to handwritten letters. *In 2017 international joint conference on neural networks (IJCNN)*, p. 2921–2926. IEEE, 2017.

[17] C. Cortes et V. Vapnik : Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[18] N. Das, A. F. Mollah, S. Saha et S. S. Haque : Handwritten arabic numeral recognition using a multi layer perceptron. *arXiv preprint arXiv :1003.1891*, 2010.

[19] S. Deepak et P. Ameer : Brain tumor classification using deep cnn features via transfer learning. *Computers in biology and medicine*, 111:103345, 2019.

[20] G. Dimauro, S. Impedovo, R. Modugno et G. Pirlo : A new database for research on bank-check processing. *In Proceedings eighth international workshop on frontiers in handwriting recognition*, p. 524–528. IEEE, 2002.

[21] E. Dufourq et B. A. Bassett : Eden : Evolutionary deep networks for efficient machine learning. *In 2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, p. 110–115. IEEE, 2017.

[22] A. El-Sawy, H. El-Bakry et M. Loey : Cnn for handwritten arabic digits recognition based on lenet-5. *In International conference on advanced intelligent systems and informatics*, p. 566–575. Springer, 2016.

[23] A. El-Sawy, M. Loey et H. El-Bakry : Arabic handwritten characters recognition using convolutional neural network. *WSEAS Transactions on Computer Research*, 5 (1):11–19, 2017.

[24] M. Elleuch, R. Maalej et M. Kherallah : A new design based-svm of the cnn classifier architecture with dropout for offline arabic handwritten recognition. *Procedia Computer Science*, 80:1712–1723, 2016.

[25] P. Ghadekar, S. Ingole et D. Sonone : Handwritten digit and letter recognition using hybrid dwt-dct with knn and svm classifier. *In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, p. 1–6. IEEE, 2018.

[26] A. D. Jia, B. Z. Li et C. C. Zhang : Detection of cervical cancer cells based on strong feature cnn-svm network. *Neurocomputing*, 411:112–127, 2020.

[27] B. Kayalibay, G. Jensen et P. van der Smagt : Cnn-based segmentation of medical imaging data. *arXiv preprint arXiv :1701.03056*, 2017.

[28] D. Keysers : Comparison and combination of state-of-the-art techniques for handwritten character recognition : topping the mnist benchmark. *arXiv preprint arXiv :0710.2231*, 2007.

[29] M. Khairandish, M. Sharma, V. Jain, J. Chatterjee et N. Jhanjhi : A hybrid

cnn-svm threshold segmentation approach for tumor detection and classification of mri brain images. *IRBM*, 2021.

[30] A. Khan, A. Sohail, U. Zahoora et A. S. Qureshi : A survey of the recent architectures of deep convolutional neural networks. *Artificial intelligence review*, 53(8):5455–5516, 2020.

[31] F. Lauer, C. Y. Suen et G. Bloch : A trainable feature extractor for handwritten digit recognition. *Pattern Recognition*, 40(6):1816–1824, 2007.

[32] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard et L. D. Jackel : Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[33] Y. LeCun, L. Bottou, Y. Bengio et P. Haffner : Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[34] F. Lekhal, M. El Hitmy et O. Melhaoui : Arabic numerals recognition based on an improved version of the loci characteristic. *Int. J. Comput. Appl*, 24:36–41, 2011.

[35] C.-L. Liu, K. Nakashima, H. Sako et H. Fujisawa : Handwritten digit recognition : benchmarking of state-of-the-art techniques. *Pattern recognition*, 36(10):2271–2285, 2003.

[36] M. Loey, A. El-Sawy et H. El-Bakry : Deep learning autoencoder approach for handwritten arabic digits recognition. *arXiv preprint arXiv :1706.06720*, 2017.

[37] X.-X. Niu et C. Y. Suen : A novel hybrid cnn–svm classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4):1318–1325, 2012.

[38] W. Pan, T. D. Bui et C. Y. Suen : Isolated handwritten farsi numerals recognition using sparse and over-complete representations. *In 2009 10th international conference on document analysis and recognition*, p. 586–590. IEEE, 2009.

[39] A. Sahloul, C. Suen *et al.* : Off-line system for the recognition of handwritten arabic character. *In Fourth international conference on computer science & information technology*, p. 227–244, 2014.

[40] M. Sarfraz, S. N. Nawaz et A. Al-Khuraidly : Offline arabic text recognition system. *In 2003 International Conference on Geometric Modeling and Graphics, 2003. Proceedings*, p. 30–35. IEEE, 2003.

[41] M. Shams, A. Elsonbaty, W. ElSawy *et al.* : Arabic handwritten character recognition based on convolution neural networks and support vector machine. *arXiv preprint arXiv :2009.13450*, 2020.

[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever et R. Salakhutdinov : Dropout : a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[43] X. Sun, J. Park, K. Kang et J. Hur : Novel hybrid cnn-svm model for recognition of functional magnetic resonance images. *In 2017 IEEE international conference on systems, man, and cybernetics (SMC)*, p. 1001–1006. IEEE, 2017.

[44] M. Szarvas, A. Yoshizawa, M. Yamamoto et J. Ogata : Pedestrian detection with convolutional neural networks. *In IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, p. 224–229. IEEE, 2005.

[45] M. Takruri, R. Al-Hmouz et A. Al-Hmouz : A three-level classifier : fuzzy c means, support vector machine and unique pixels for arabic handwritten digits. *In 2014 World Symposium on Computer Applications & Research (WSCAR)*, p. 1–5. IEEE, 2014.

[46] S. A. Tuncer et A. Alkan : Classification of emg signals taken from arm with hybrid cnn-svm architecture. *Concurrency and Computation : Practice and Experience*, 34 (5):e6746, 2022.

ANNEXE A : Hybride CNN-SVM : vers une meilleure classification des données manuscrites

Benkadja, A., Biskri, I., & Ghazzali, N. (2022, July). Hybride CNN-SVM : vers une meilleure classification des données manuscrites. In *JADT 2022-Proceedings of the 16th International Conference on Statistical Analysis of Textual Data (Vol. 1, pp. 119-126). Vadistat press & Edizioni Erranti.*

Hybride CNN-SVM : vers une meilleure classification des données manuscrites

Abdallah Benkadja, Ismaïl Biskri, Nadia Ghazzali

Université du Québec à Trois-Rivières – abdellah.benkadja@uqtr.ca
ismail.biskri@uqtr.ca nadia.ghazzali@uqtr.ca

Abstract
We present in this paper a hybrid model combining the CNN convolutional neural network with the SVM classifier. Experiments carried out on handwritten digits and letters from the EMNIST dataset show higher recognition rates than those obtained with CNN and SVM alone. We found by varying some parameters of CNN and SVM that it is possible to further improve the recognition rates of the hybrid CNN-SVM approach.
Keywords: Hybrid CNN-SVM, Classification, EMNIST

Résumé
Nous présentons dans ce papier un modèle hybride combinant le réseau de neurones à convolutions CNN avec le classifieur SVM. Des expérimentations menées sur des chiffres et des lettres manuscrits issus du dataset EMNIST permettent d'observer des taux de reconnaissance supérieurs à ceux obtenus avec CNN et SVM seuls. Nous avons constaté en faisant varier certains paramètres de CNN et de SVM qu'il est possible d'améliorer davantage les taux de reconnaissance de l'approche hybride CNN-SVM.

Mots clés: Méthode hybride CNN-SVM, Classification, EMNIST

1. Introduction

La reconnaissance automatique du manuscrit revêt un intérêt majeur dans le domaine du traitement de l'image et de la reconnaissance des formes. Un plus grand taux de reconnaissance est toujours demandé par la recherche en apprentissage automatique et en intelligence artificielle et même par des applications concrètes en industrie.

De nombreuses études consacrent des approches d'apprentissage automatique dans la reconnaissance des caractères manuscrits et la reconnaissance des formes (LeCun *et al.*, 1998; Cortes, Vapnik, 1995; Baldominos *et al.*, 2019 ; Liu *et al.*, 2003; Keysers, 2007). Parmi les approches les plus utilisées, nous avons les réseaux de neurones à convolution (Convolutional Neural Network) CNN (LeCun *et al.*, 1998) et les machines à vecteurs de support (Support Vector Machine) SVM (Cortes, Vapnik, 1995). CNN est une classe de réseaux de

neurones artificiels avancés, largement utilisés dans les problèmes de vision par ordinateur et de traitement de l'image. SVM est une technique d'apprentissage supervisé pour des fins de classification, largement utilisée dans la recherche d'information, la vision par ordinateur, la biologie, etc.

Malgré leur efficacité démontrée dans le domaine de la reconnaissance de forme, plus particulièrement, la reconnaissance des chiffres et des lettres manuscrits, on constate, plus récemment, l'émergence d'une approche hybride combinant CNN et SVM dans laquelle la dernière couche de CNN est remplacée par le classifieur SVM, avec les travaux de (Khairandish *et al.*, 2021) sur la classification d'images IRM de tumeurs cérébrales et de classification du manuscrit sur la classification de caractères manuscrits (Ali, Mallaiah, 2021 ; Ahlawat, Choudhary, 2020 ; Niu, Suen, 2012). Dans la reconnaissance des chiffres manuscrits provenant de MNIST (http://yann.lecun.com/exdb/mnist/), Ahlawat et Choudhary (2020), et Niu et Suen (2012) rapportent pour cette approche hybride des taux de reconnaissance respectifs de 98,88 et 99,81, une augmentation respectivement de 1,03 % (98,88 % - 97,85 %) par rapport à SVM seul et de 0,4 % (99,81 % - 99,41 %) par rapport à CNN seul.

Bien que ces travaux tendent à démontrer que Hybride CNN-SVM améliore le taux de reconnaissance des chiffres manuscrits de MNIST, l'avantage de son utilisation, comparativement à CNN et SVM seuls, n'est pas clair quand appliquée à des datasets plus hétérogènes. En effet, la majorité des travaux publiés sur Hybride CNN-SVM donnent des résultats d'expérimentation, seulement, sur les chiffres manuscrits issus de (MNIST).

Notre travail de recherche porte sur une étude comparative des résultats de reconnaissance des caractères manuscrits, à partir de datasets plus variés, obtenus avec CNN, SVM et Hybride CNN-SVM. Un large éventail de paramètres tant pour CNN que pour SVM est considéré.

Les jeux de données utilisés pour l'exploration empirique et l'étude comparative sont ceux de Extended MNIST (EMNIST) (https://www.nist.gov/itl/products-and-services/emnist-dataset) (Cohen *et al.*, 2017). Ceux-ci comportent plusieurs ensembles de données qui varient en nombre de classes et en taille d'échantillons d'entrainement et de test. Nous nous limitons, pour les besoins de cette recherche, aux trois jeux de données suivants : EMNIST Letters, EMNIST Balanced et EMNIST MNIST (voir tableau 1). Ces derniers contiennent des données qui ont le même format. Ce sont des images de 28x28 pixels sur une échelle de gris allant de 0 à 255. EMNIST MNIST est un dataset de chiffres, à 10 classes. EMNIST Letters est représenté par 26 classes de caractères latins. On n'y distingue pas les majuscules des minuscules. EMNIST Balanced comprend 47 classes de chiffres et de lettres (pour certaines lettres, on distingue les majuscules des minuscules).

Tableau 1. Datasets utilisés

Dataset	EMNIST MNIST	EMNIST Letters	EMNIST Balanced
Entrainement	60 000	88 800	112 800
Test	10 000	14 800	18 800
Total	70 000	103 600	131 600
Nombre de classes	10	26	47

2. Hybride CNN-SVM

Comme notre travail porte sur Hybride CNN-SVM, nous présentons brièvement SVM et CNN ainsi que leurs différents paramètres en lien avec les expérimentations de notre étude.

SVM vise à obtenir les paramètres de l'hyper-plan ou un ensemble d'hyper-plans dans un espace à dimensions relatives au jeu de données utilisé (Cortes et Vapnik 1995). Dans le travail en cours, le noyau gaussien (RBF) est utilisé sous différentes variations de γ et **C**.

γ est un hyperparamètre que nous devons définir avant d'entraîner le modèle. Il détermine la forme de l'hyperplan qui est la frontière de décision. Une valeur élevée de γ signifie plus de courbure. Une valeur basse de γ signifie moins de courbure. γ est obtenu au moyen de la formule $\gamma=(n*var(X))$, **var(X)** étant la variance du dataset d'entrainement et n le nombre de variables aléatoires représentant les valeurs des pixels des images en entrée. La fonction RBF du noyau gaussien utilisée dans notre travail est définie par l'équation suivante : $\exp(-\gamma\|x-x'\|^2)$. Le paramètre **C** indique la marge de tolérance à l'erreur pour les données d'entrainement. Une valeur élevée de **C** est susceptible de mener à un surapprentissage du modèle qui peut biaiser les résultats de test. Inversement, une très petite valeur de C amènera l'optimiseur à rechercher un hyperplan de séparation à plus grande marge pour les datasets d'entrainement. Enfin la fonction de décision One-Vs-Rest (**OVR**) permet de diviser l'ensemble des données multi-classes en plusieurs problèmes de classification binaire.

Dans le cas de CNN (voir le tableau 2), nous utilisons trois architectures différentes, étiquetées *CNN-1*, *CNN-2* et *CNN-3*. *CNN-1* correspond à la même structure CNN simplifiée à 5 couches qui a été présentée dans (Pan *et al.*, 2009). *CNN-2* utilise la même architecture LeNet-5 présentée dans (LeCun *et al.* 1998), à la seule différence que le nombre d'unités de sortie correspond au nombre de classes du dataset d'entrée. *CNN-3* se base sur la même structure que *LeNet-5* avec quelques paramètres modifiés.

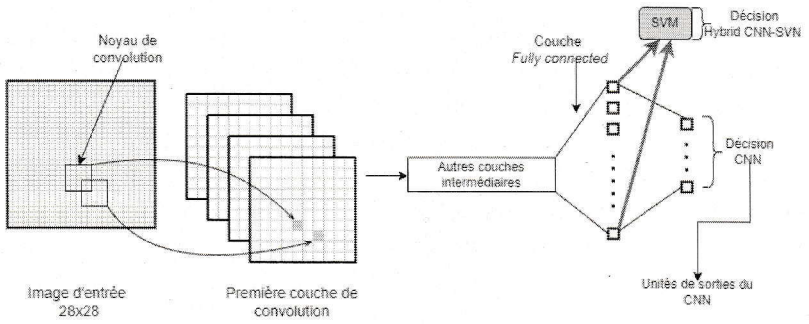
Tableau 2. Architectures CNN utilisées

Couche	CNN-1	CNN-2	CNN-3
1	INPUT	INPUT	INPUT
2	CONV2D filters : 50 kernel : 5 x 5 stride : 2	CONV2D filters : 6 kernel : 5 x 5 stride : 1	CONV2D filters : 32 kernel : 5 x 5 stride : 1
3	CONV2D filters : 50 kernel : 5 x 5 stride : 2	AveragePool	MaxPool stride : 2
4	FC : 100	CONV2D Filters : 16 Kernel : 5 x 5 Stride : 1	CONV2D filters : 48 kernel : 5 x 5 stride : 1
5	FC : OUTPUT	AveragePool	MaxPool Stride : 2
6		FC : 120	FC : 256
7		FC : 84	FC : 84
8		FC : OUTPUT	FC : OUT-PUT

Nous présentons dans le tableau 2 les différents paramètres de CNN : (i) le *Kernel* qui est une matrice correspondant au noyau de convolution, (ii) le *Stride* qui fait référence au pas que le noyau de convolution considère dans son parcours de la matrice d'entrée, (iii) le *Pooling* dont l'objectif principal est de réduire le nombre d'éléments des matrices de la couche d'entrée, (iv) le *AveragePooling* qui renvoie la moyenne de toutes les valeurs de la partie traitée par le noyau et l'insère dans la position correspondante de la matrice de sortie, (v) le *MaxPooling* qui renvoie la valeur maximale de la partie de la matrice traitée par le noyau et (vi) la couche *Fully connected* (FC) qui sert à convertir la matrice d'entrée en vecteurs entièrement connectés comme un réseau de neurones.

Tel que présenté dans la figure 1, les images d'entrée, normalisées et centrées, représentent la couche d'entrée de notre modèle hybride CNN-SVM. La dernière couche de CNN est remplacée par le classifieur SVM. Ce dernier prend les sorties de la couche *Fully connected* qui précède les sorties de décision de CNN comme un nouveau vecteur de caractéristiques pour l'entrainement par SVM. Les décisions obtenues sont dès lors le résultat de la méthode hybride.

Figure 1. Hybride CNN-SVM



L'objectif du travail étant d'apporter le plus d'éléments de comparaison des résultats de Hybride CNN-SVM à ceux obtenus avec CNN et SVM utilisé chacun individuellement, un large éventail de paramètres tant pour CNN que pour SVM seront expérimentés en leur assignant différentes valeurs. En outre, notre expérimentation empirique s'effectuera sur des jeux de données variés en nombre de classes et en tailles d'échantillons.

3. Expérimentation

Pour les besoins de l'expérimentation, nous avons implémenté le modèle hybride en utilisant les librairies de Keras (Chollet, 2015) et LibSVM (Chang *et al.*, 2011). Dans le tableau 3, nous présentons une synthèse des résultats de nos expérimentations ainsi que des résultats trouvés dans la littérature. La première étape de nos expérimentations consiste à effectuer la classification des images manuscrites de nos trois datasets (Tableau 1) au moyen de SVM en faisant varier le paramètre C, et au moyen des trois architectures CNN. Ensuite, nous utilisons le modèle Hybride CNN-SVM en faisant varier, également, le paramètre C et en considérant les trois architectures CNN.

Nous constatons que pour les chiffres manuscrits (MNIST) les taux de reconnaissance sont très élevés (**98,34-99,47%**). Les résultats montrent que CNN-3 a donné le meilleur taux de reconnaissance parmi les 3 autres architectures de CNN. Cette même architecture CNN-3, quand combinée à SVM dans Hybride CNN-SVM permet un meilleur taux de reconnaissance supérieur de 0.71% par rapport à SVM et de **0,10%** par rapport à CNN-3. C'est un constat semblable dans le cas l'Hybride CNN-SVM utilisant CNN-1 et CNN-2. Néanmoins, cette amélioration du taux de reconnaissance reste minime puisque inférieure à 1 %. Dans le cas des lettres (EMNIST Letters), nous observons une augmentation plus significative du taux de reconnaissance de l'approche Hybride CNN-SVM. Cette augmentation est observée peu importe l'architecture CNN utilisée. Hybride-CNN-1-SVM augmente le taux de reconnaissance par rapport à CNN-1 de **1,88%**. Hybride-CNN-2-SVM augmente le taux de reconnaissance par rapport à CNN-2 de **1,28%**. Hybrid-CNN-3-SVM augmente le taux de reconnaissance par rapport à CNN-3 de **1,31%**. L'amélioration du taux de reconnaissance d'Hybride CNN-3-SVM est encore plus significative comparativement à celui de SVM. L'augmentation est supérieure à 2%. Des trois architectures hybrides, Hybride-CNN-3-SVM est celle qui permet d'obtenir le meilleur taux de reconnaissance. Dans le cas de EMNIST Balanced qui contient et des chiffres et des lettres, le nombre de classes est plus important. Cela se traduit empiriquement par un taux de reconnaissance moins élevé. Néanmoins, nous remarquons qu'avec les trois combinaisons hybrides, une augmentation du taux de reconnaissance comparativement à CNN et à SVM. Le meilleur taux de reconnaissance est obtenu avec Hybride-CNN-3-SVM avec 1,57% de plus par

rapport à CNN-3 et 2,6% par rapport à SVM.

Par ailleurs, nous observons que pour l'ensemble de nos datasets, CNN-3 permet d'obtenir des taux de reconnaissance supérieurs à ceux de CNN-1 et CNN-2. Hybride CNN-3-SVM donne, également, les meilleurs taux de reconnaissance comparativement aux variantes Hybrides CNN-1-SVM et CNN-2-SVM. Cela démontre que les performances du modèle hybride varient en fonction des paramètres de ses composantes que sont CNN et SVM. Globalement, l'approche l'Hybride CNN-SVM est empiriquement avantageuse puisqu'elle tend, globalement, selon nos expérimentations, à donner des résultats meilleurs que CNN et SVM.

Biswas et Islam (2021) ont obtenu un taux de reconnaissance des chiffres au moyen de CNN égal à 99,53%. Cavalin et Oliviera (2019) ont obtenu un taux de reconnaissance des lettres au moyen de CNN égal à 93,63%. Nos meilleurs résultats avec Hybride CNN-SVM leur sont légèrement inférieurs. Néanmoins, quand appliqué à un jeu de données plus large et plus hétérogène, notre approche hybride donne un meilleur résultat que celui de Cavalin et Oliveira avec 88,22% de taux de reconnaissance contre 87,18%.

Tableau 3. Résultats des expérimentations (en %)

Technique	**Paramètre C de SVM**	**MNIST (Chiffres)**	**Letters (Lettres)**	**Balanced**
SVM	1	98,34	88,98	84,28
	3	98,73	90,10	85,52
	5	**98,76**	**90,29**	**85,62**
CNN-1		99,10	90,50	85,38
CNN-2		98,96	90,97	86,38
CNN-3		**99,37**	**91,53**	**86,65**
Hybride CNN-1-SVM	1	99,30	92,38	87,89
	3	99,27	92,32	87,71
	5	99,20	92,23	87,49
Hybride CNN-2-SVM	1	99,16	92,20	87,87
	3	99,23	92,22	88,09
	5	99,20	92,25	88,08
Hybride CNN-3-SVM	1	99,43	92,79	**88,22**
	3	**99,47**	**92,84**	88,03
	5	**99,47**	92,66	87,91
CNN (Biswas, Islam, 2021)		99,53		
SVM (Ahlawat, Choudhary, 2020)		97,85		
CNN (Cavalin, Oliveira, 2019)		99,46	93,63	87,18
Hybride CNN-SVM (Ahlawat, Choudhary, 2020)		98,88		
Hybride CNN-SVM (Niu, Suen, 2012)		99,81		

4. Conclusion et perspectives

Le modèle Hybride CNN-SVM proposé dans ce travail est abordé selon deux perspectives : Sa capacité à reconnaitre les caractères manuscrits et celle à maintenir sa performance en termes de taux de reconnaissance par rapport à CNN et à SVM. Notre approche utilise CNN dans l'extraction des descripteurs des caractères manuscrits avant d'utiliser SVM comme classifieur final. Nos expérimentations ont été menées sur des données variées composées de chiffres et de lettres issues de EMNIST.

Parmi toutes les combinaisons expérimentées, nos résultats empiriques montrent que l'approche Hybride CNN-3-SVM donne les meilleures classifications pour tous nos datasets. Une expérimentation et une validation avec des ensembles de données plus complexes et plus variées pourrait encore davantage consolider les résultats de sa performance.

Bien entendu, nous pensons que l'expérimentation avec la variation des paramètres de SVM et CNN reste au centre de l'amélioration des performances du modèle hybride. Nous en avons exploré quelques-uns dans ce travail. D'autres seront considérés dans nos travaux futur. Parmi ceux-là, il y a les paramètres des couches intermédiaires de CNN, et ceux du noyau SVM, etc.

Bibliographie

Ahlawat S. et Choudhary A. (2020). "Hybrid CNN-SVM Classifier for Handwritten Digit Recognition." In *Procedia Computer Science*, Volume 167. Elsevier. 2554-2560.

Ali A.A.A. et Mallaiah S. (2021). Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with dropout. *Journal of King Saud University-Computer and Information Sciences*.

Baldominos A., Saez Y. et Isasi P. (2019). "A Survey of Handwritten Character Recognition with MNIST and EMNIST." *Journal of Applied Sciences* 9(15). MDPI.

Biswas A. et Islam M. S. (2021). "An Efficient CNN Model for Automated Digital Handwritten Digit Classification." *Journal of Information Systems Engineering and Business Intelligence*, 7(1), 42-55.

Cavalin P. et Oliveira L. (2019). "Confusion Matrix-Based Building of Hierarchical Classification". In proceedings of the 2018 Iberoamerican Congress on Pattern Recognition. LNCS. Springer. 271–78.

Chang C. C. et Lin C. J. (2011). LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3), 1-27.

Chollet F. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras

Cohen G., Afshar S., Tapson J. et Van Schaik A. (2017). "EMNIST: Extending MNIST to Handwritten Letters." In *Proceedings of the International Joint Conference on Neural Networks*. pp. 2921-2926

Cortes C. et Vapnik V. (1995). "Support-Vector Networks." *Machine Learning* 20(3): 273–97. Springer.

Khairandish M.O., Sharma M., Jain V., Chatterjee J. M. et Jhanjhi N. Z. (2021). "A Hybrid CNN-SVM Threshold Segmentation Approach for Tumor Detection and

Classification of MRI Brain Images." *IRBM*. Elsevier.
Keysers D. (2007). "Comparison and Combination of State-of-the-Art Techniques for Handwritten Character Recognition: Topping the MNIST Benchmark." arXiv preprint *arXiv:0710.2231*
LeCun Y., Cortes C. et Burges C. J. C. (1998). "MNIST Handwritten Digit Database" http://yann.lecun.com/exdb/mnist/.
Liu C. L., Nakashima K., Sako H. et Fujisawa H. (2003). "Handwritten Digit Recognition: Benchmarking of State-of-the-Art Techniques." *Pattern Recognition* 36(10). Elsevier. 2271–2285.
Niu X. X. et Suen C. Y. (2012). "A Novel Hybrid CNN-SVM Classifier for Recognizing Handwritten Digits." *Pattern Recognition* 45(4). Elsevier. 1318–1325.
Pan W. M., Bui T. D. et Suen C.Y. (2009). "Isolated Handwritten Farsi Numerals Recognition Using Sparse and Over-Complete Representations." In *Proceedings of the International Conference on Document Analysis and Recognition*, ICDAR, IEEE Computer Society, 486–90.

ANNEXE B : Statistical profiling of Hybride CNN-SVM effectiveness

Article soumis le 24 février 2023

Benkadja, A., Ben Ayed, A., Biskri, I., & Ghazzali, N. Statistical profiling of Hybride CNN-SVM effectiveness. In *New Frontiers in Textual Data Analysis (M. Misuraca & G. Giordano, editors).Springer Nature.* Publié par de la série de *Studies in Classification, Data Analysis, and Knowledge Organization.*

Statistical profiling of Hybrid CNN-SVM effectiveness

No Author Given

Abstract Explaining decisions taken by a deep neural network is a challenging task. Even though significant advances have been made in this context during the last two decades, it remains a hot research problem. Combining a deep neural network with a classic AI model can be helpful in the context of model explainability. However, it may reduce its discriminating power. This paper performs statistical profiling on the effectiveness of hybrid deep learning models: a hybrid model of convolutional neural networks and support vector machines is used as a use case. We assess its performance on some state-of-the-art Latin and Arabic handwritten datasets. The proposed statistical study focuses on the hybrid deep learning model's ability to recognize handwritten characters and maintain its performance in recognition rates compared to state-of-art CNN and SVM architectures. Obtained results show that the hybrid neural network achieves better overall classification accuracy. The findings of this research prove the effectiveness of hybrid deep neural networks and draw new bridges toward their explainability.

1 Introduction

Machine learning has become a crucial technology in various industries and has many applications ranging from computer vision and natural language processing to predictive policing and healthcare [19]. Artificial Intelligence (AI) models can be grouped into several families based on their underlying architecture, functionality, and applications [19]. The most important ones are :

1. *Deep Learning Models*: They have been the driving force behind many breakthroughs in AI in recent year. Their architecture, made of multiple neuronal layers, is designed to model complex relationships and non-linear interactions between input and output variables.

2. *Generative Models*: They can generate new data much of a muchness to the fed training data. They are used in most of applications generating new images, music, or text.
3. *Reinforcement Learning Models*: They can learn from interactions with their environment and make decisions based on their experience. This has the potential to be applied to a wide range of real-world problems, such as autonomous systems and robotics.
4. *Transfer Learning Models*: They allow for efficient and effective training by transferring knowledge from one task to another related task. This can be a cost-effective solution for many real-world computer vision and NLP applications.

Many generative, reinforcement learning and transfer learning models rely on deep neural network architectures. Indeed, generative models, such as GANs [5] and VAEs [5], use multiple layers of neural networks to learn the underlying patterns and relationships in the data. Also, reinforcement learning models such as Deep Q-Network (DQN) [5], Policy Gradients [5], A3C (Asynchronous Advantage Actor-Critic) [5], and Proximal Policy Optimization (PPO) [5] use deep neural network architectures to learn from interactions with their environment and make decisions based on their experience. Also, state-of-the-art transfer learning models such as VGG [5], ResNet [5], BERT [5], GPT-2/3 [5] make use of deep neural architectures to transfer knowledge from one task to another related one.

Despite the wide range of deep neural network applications, they are considered to be black box models as they do not provide explicit and intuitive explanations for their predictions. one bridge toward complex deep learning explainability is to mix those models with classic approaches. Even though the classic model to mix with is a black box, like SVMs, explaining their results remains easier than deep neural networks. Moreover, recent research has been focusing on developing methods to make SVMs more interpretable, such as visualizing the decision boundary or incorporating domain knowledge into the model to help explain predictions. Once the explainability is met, the hybrid model's main challenge is maintaining its discriminative power. For this purpose, we perform statistical profiling of deep neural networks' effectiveness using a hybrid model of convolutional neural networks and support vector machines as a use case.

The rest of this paper is broken down as follows: the second section describes related work on hybrid CNN-SVM architectures. The third one details the experimental protocol. The fourth section reports and discusses experimental results and the fifth section puts forth conclusions.

2 Related work

Hybrid CNN-SVM models were tested on the MNIST handwriting recognition dataset in [1], and [16]. Although authors tended to demonstrate that Hybrid CNN-

Table 1 Datasets used in experiments

Dataset	MNIST	Letters	Balanced	By Merge	By Class	AHCD	MADBase
Training	60,000	88,800	112,800	697,932	697,932	13,440	60,000
Test	10,000	14,800	18,800	116,932	116,932	3,360	10,000
Total	70,000	103,600	131,600	814,255	814,255	16,800	70,000
Classes	10	26	47	47	62	28	10

SVM improves the recognition rate of MNIST handwritten digits, the advantage of its use, compared to CNN and SVM alone, is not clear when applied to more heterogeneous datasets. [22] also proposed and evaluated a hybrid CNN-SVM model in a pedestrian detection system. Conducted research in [13] shows that a hybrid CNN-SVM architecture called *TFE-SVM* outperformed LeNet5 when tested on the MNIST dataset. Note that the scope of the works involving hybrid architectures combining CNN and SVM remains mostly limited to the domain of Latin digits. In this work, we statistically assess the perofrmance of CNN-SVM architectures using Arabic hanwritten digits datasets.

3 Experimental protocol

3.1 Datasets

In this paper, we use three standard handwriting datasets, namely EMNIST, AHCD, and MADBase illustrated in Table 1. Those datasets have yet to be explored or only slightly explored and are considerably more complex in the number of classes and nature of data.

3.1.1 EMNIST

The EMNIST is a 28x28 pixel image format dataset. It is derived from the NIST Special Database 19 [8]. Some handwritten characters from this database are shown in the Fig. 1. Note that the ByClass, and ByMerge splits provide the full complement of the NIST Special Database 19.

3.1.2 Arabic Handwritten Characters Dataset (AHCD)

The AHCD dataset is made of 16,800. It was proposed by [11]. It is quite challenging regarding the style of writing, thickness, dots number and position etc. Some handwritten characters of AHCD are shown in the Fig. 2.

Fig. 1 Samples of the EMNIST dataset.

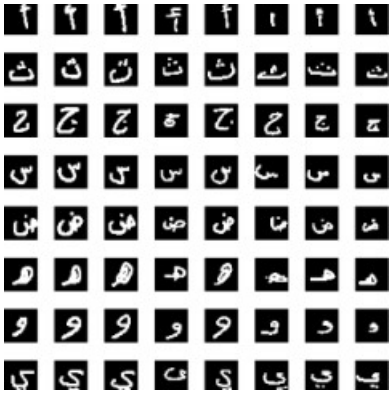


Fig. 2 Samples from the AHCD dataset.

Arabic Digit	English Digit	Image
١	1	
٢	2	
٣	3	
٤	4	
٥	5	
٦	6	
٧	7	
٨	8	
٩	9	
٠	0	

Fig. 3 Scale and rotation-invariance problems in Arabic handwritten digits.

3.1.3 MADBase

The MADBase database was proposed by [21]. It is a collection of handwritten Eastern Arabic numerals, also called Arabic-Hindu numerals or Indo–Arabic numerals. They are used to represent numerical digits in the Middle Eastern countries. Fig. 3 illustrates this variant. MADbase includes 70 000 arabo-indian numerals. The latter ones are more challenging than those used in [21] due to the rotation-invariance and scale problems.

3.2 Evaluation methodology

We use the EMNIST, AHCD, and MADBase datasets to assess the performance of a Hybrid CNN-SVM model. Classification precision rates of the latter one are compared to those relative to classic CNN and SVM classifiers.

3.2.1 The SVM architecture

The Gaussian kernel (RBF) is used by default. The RBF function of the SVM kernel on two feature vectors x and x' is defined as follows:

$$e^{-\gamma\|x-x'\|^2} \tag{1}$$

where γ is an hyperparameter of the RBF kernel function. It determines the shape of the hyperplane which is the decision boundary. It is defined as:

$$\gamma = n\mathrm{Var}[X] \tag{2}$$

$\mathrm{Var}[X]$ represents the variance of the training dataset, and n is the number of random variables corresponding to the number of pixels in the input image. The tolerance margin C is another critical parameter in the behavior of SVM: it indicates the error for the training data. A high value of C will likely lead to an over-learning of the model, which may bias the recognition results. Conversely, a minimum value of C will lead the optimizer to search for a higher margin separation hyperplane for the training datasets. Later, we describe different experiments with four values of C; $C \in \{1, 3, 5, 10\}$. The corresponding SVM variants are labeled SVM_C; $C \in \{1, 3, 5, 10\}$.

3.2.2 The CNN architecture

We will apply five different architectures of CNN, respectively labeled CNN_1, CNN_2, CNN_3, CNN_4, and CNN_5, whose structures are described in Table 2. CNN_1 corresponds to the 5-layer CNN structure presented and used by Pan et al. [17]. CNN_2 corresponds to the LeNet-5 architecture [14], the only difference being that the number of output units corresponds to the number of classes in the input dataset. CNN_3 is based on the same structure as LeNet-5 with some modified parameters, as illustrated in Table 2. CNN_4 corresponds to the same CNN architecture proposed by El-Sawy et al. [11]. For the CNN_5, we use the same architecture as CNN4 architecture, and we apply Dropout on the intermediate layers.

The parameter Kernel in Table 2 refers to a small matrix of weights used to perform convolution in a CNN. The size of the kernel and the number of kernels used are critical design choices that can impact the model's performance.The stride is a hyper-

Table 2 CNN architectures used in this paper

Layer	CNN_1	CNN_2	CNN_3	CNN_4	CNN_5
1	INPUT	INPUT	INPUT	INPUT	INPUT
2	CONV2D filters:50 kernel:5x5 stride:2	CONV2D filters:6 kernel:5x5 stride:1	CONV2D filters:32 kernel:5x5 stride:1	CONV2D filters:80 kernel:5x5 stride:1	CONV2D filters:80 kernel:5x5 stride:1
	-	-	-	-	Dropout
3	CONV2D filters:50 kernel:5x5 stride:2	AveragePool	MaxPool stride:2	MaxPool stride:2	MaxPool stride:2
4	FC:100	CONV2D filters:16 kernel:5x5 stride:1	CONV2D filters:48 kernel:5x5 stride:1	CONV2D filters:64 kernel:5x5 stride:1	CONV2D filters:64 kernel:5x5 stride:1
	-	-	-	-	Dropout
5	FC:OUTPUT	AveragePool	MaxPool stride:2	MaxPool stride:2	MaxPool stride:2
6	—	FC:120	FC:256	FC:1024	FC:1024
7	—	FC:84	FC:84	-	-
8	—	FC:OUTPUT	FC:OUTPUT	FC:OUTPUT	FC:OUTPUT

parameter in a CNN that determines the step size with which the kernel moves over the input feature map during the convolution operation. Pooling is a down-sampling operation commonly used in Convolutional Neural Networks (CNNs) to reduce the spatial dimensionality of the feature maps. Max pooling and average pooling, whose parameters in Table 2 are respectively MaxPool and AveragePool.

3.2.3 The hybrid CNN SVM architecture

The matrix describing the feature vectors of the normalized and centered images is fed to the input layer of our hybrid CNN-SVM model, as illustrated in Fig. 4. The last layer of CNN is replaced by the SVM classifier. The latter takes the outputs of the CNN Fully-connected layer during the training process. The final decisions are then the result of the hybrid method.

The following terminology is used for the rest of this paper:

- SVM_i denotes the SVM model having as parameter $C = i$ given $i \in \{1, 3, 5, 10\}$.
- CNN_i denotes the corresponding CNN architecture as illustrated in table 2.
- HYBRID_i denotes the CNN_i based hybrid approach.
- $\text{HYBRID}_{i,j}$ denotes the model combining CNN_i and SVM_j.
- $\mu_{\text{HYBRID}}(d)$ denotes the average recognition rate by all hybrid approaches for the dataset $\boldsymbol{d}$.

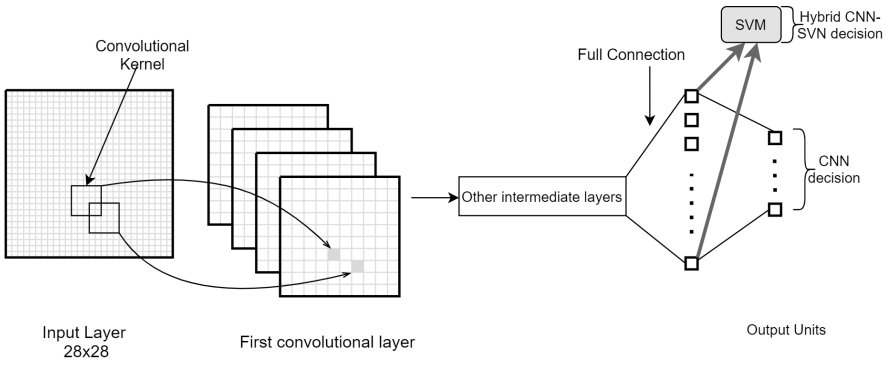


Fig. 4 Hybrid CNN-SVM model

- $\mu_{\text{SVM}}(d)$ denotes the average recognition rate by SVM for the dataset $\boldsymbol{d}$.
- $\mu_{\text{HYBRID}_i}(d)$ denotes the average of the results obtained from recognition rates by the hybrid approach based on CNN_i for dataset $\boldsymbol{d}$.

- **H-1: Does Hybrid CNN-SVM significantly improve the recognition rate compared to SVM**?
  - $\text{H}_1 - 0$: $\mu_{\text{HYBRID}}(d) = \mu_{\text{SVM}}(d)$
  - $\text{H}_1 - 1$: $\mu_{\text{HYBRID}}(d) > \mu_{\text{SVM}}(d)$
- **H-2: Does SVM, when applied on extraction results from CNN$_i$, improve accuracy compared to SVM applied on datasets without feature extraction?**
  - $\text{H}_{2,i} - 0$: $\mu_{\text{HYBRID}_i}(d) = \mu_{SVM}(d)$
  - $\text{H}_{2,i} - 1$: $\mu_{\text{HYBRID}_i}(d) > \mu_{SVM}(d)$
- **H-3: Does the hybrid HYBRID$_i$ approach increase accuracy significantly over CNN$_i$ subjasent?**
  - $\text{H}_{3,i} - 0$: $\mu_{\text{HYBRID}_i}(d) = \mu_{\text{CNN}_i(d)}$
  - $\text{H}_{3,i} - 1$: $\mu_{\text{HYBRID}_i}(d) > \mu_{\text{CNN}_i(d)}$

The Welch's t-test is used for the validation of H_1 and H_3. The Paired t-test is used in the elaboration of H_2. A comparative study with the results from the literature for the different datasets will be performed in the coming section.

4 Experimental results

Hybrid combinations are formed through many varied parameters for both CNN and SVM. Experimental results are illustrated in Tables 3 and 4. The experimentation on AHCD and MADBase has been extended to involve CNN_4 and CNN_5 with

associated hybrid approaches. CNN_1, CNN_2 and CNN_3 were experimented together with their hybrid architectures involving them on the whole datasets. CNN_5 was experimented on Balanced dataset among other subdatasets of EMNIST.

Table 3 Recognition rates by SVM and CNN.

Technique	SVM_1	SVM_3	SVM_5	SVM_{10}	μ_{SVM}	σ^2	CNN_1	CNN_2	CNN_3	CNN_4	CNN_5
AHCD	65.29	71.36	72.08	72.23	70.24	11.03	82.70	77.79	84.40	93.69	95.23
MADBase	98.45	98.62	98.68	98.67	98.60	0.01	98.65	98.71	99.1	99.07	99.17
MNIST	98.34	98.73	98.76	93.60	97.36	6.31	99.10	98.96	99.37	-	-
Letters	88.98	90.10	90.29	90.27	89.91	0.39	90.50	90.97	91.53	-	-
Balanced	84.28	85.52	85.62	85.37	85.20	0.38	85.38	86.38	86.65	-	88.96
By Merge	84.19	85.54	85.67	85.55	85.24	0.49	89.30	89.88	89.47	-	-
By Class	82.36	83.38	83.32	-	83.65	1.82	85.42	86.08	85.64	-	-

Table 4 Recognition rates by Hybrid CNN-SVM.

	$HYBRID_1$		$HYBRID_2$		$HYBRID_3$		$HYBRID_4$		$HYBRID_5$			
	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
AHCD	89.53	0.04	84.18	0.19	91.61	0.01	94.56	0.02	96.74	0.12	91.32	20.23
MADBase	98.90	0.002	98.84	0.0006	99.11	0.001	99.15	9.16E-05	99.34	0.0002	99.06	0.034
MNIST	99.29	0.003	99.20	0.001	99.45	0.001	-	-	-		99.29	0.012
Letters	92.29	0.005	92.23	0.001	92.71	0.015	-	-	-	-	92.41	0.056
Balanced	87.62	0.05	88.01	0.01	87.97	0.04	-	-	89.67	0.063	88.32	0.7
By Merge	89.67	0.026	89.85	0.004	89.77	0.017	-	-	-	-	89.76	0.01
By Class	86.30	0.16	86.35	0.005	86.28	0.053	-	-	-	-	86.31	0.06

Table 5 p-values for the tested hypothesis.

	MNIST	Letters	Balanced	By Merge	By Class	AHCD	MADBase
H_1:	0.01	0.002	0.0002	0.0005	0.005	1.97E-11	0.0001
$H_{2,1}$:	0.02	0.003	0.005	0.0008	0.001	0.0002	0.015
$H_{2,2}$:	0.02	0.002	0.001	0.0005	0.005	1.36E-06	0.007
$H_{2,3}$:	0.01	0.002	0.003	0.0007	0.005	5.95E-05	0.002
$H_{2,4}$:	-	-	-	-	-	0.0001	0.001
$H_{2,5}$:	-	-	0.0008	-	-	0.0001	0.0003
$H_{3,1}$:	0.006	8.04E-06	0.004	0.01	0.01	0.0001	0.0007
$H_{3,2}$:	0.0002	2.04E-06	0.001	**0.2**	0.002	0.0007	0.0009
$H_{3,3}$:	0.0006	0.00014	0.003	0.009	0.005	3.55E-05	**0.29**
$H_{3,4}$:	-	-	-	-	-	0.004	0.0002
$H_{3,5}$:	-	-	0.0007	-	-	0.009	9.8E-5

We observe, through our experimental results, as shown in Table 4, that the values of the variance σ^2, for the combined recognition rates of all hybrid models, is higher comparing to the variances of the recognition rates of each hybrid model. We also

observe that the recognition rates by hybrid approaches seem to be influenced more by the underlying CNN architectures than by the SVM parameters. This observation allows us to guarantee that the comparison of the recognition rates of SVM and CNN with the average of the recognition rates of the HYBRID_i, is representative of their comparison with the recognition rates of all the Hybrid CNN-SVM architectures.

Table 6 Recognition rate comparison with previously conducted research on the EMNIST dataset

Authors	Technique	By Class	By Merge	Balanced	Letters	MNIST
Cohen et al. [8]	Linear Classifier	51.80%	50.51%	50.93%	55.78%	84.70%
Cohen et al. [8]	OPIUM	69.71%	72.57%	78.02%	85.15%	95.90%
Ghadekar et al. [12]	DWT-DCT + SVM	-	-	-	89.51%	97.74%
Dufourq and Bassett [9]	EDEN	-	-	88.3%	89.51%	99.3%
Biswas and Islam [6]	CNN	-	-	-	-	99.53
Cavalin and Oliveira [7]	CNN	-	-	87.18%	93.63%	99.46%
Ahlawat and Choudhary [1]	CNN	-	-	-	-	97.85%
Niu and Suen [16]	CNN	-	-	-	-	99.81%
Our approach	Hybrid CNN-SVM	86.82%	- 89.9% -	89.93%	92.84%	99.47%

Table 7 Recognition rate comparison with previously conducted research on the AHCD and MADBase datasets

Authors	Dataset	Technique	Recognition rate
Alkhawaldeh [3]	ADBase	LeNet + LSTM	98.92
Alkhateeb [2]	ADBase	CNN	94.3
Loey et al. [15]	MADBase	Autoencoder + Softmax	95.5
El-Sawy et al. [10]	MADBase	CNN Net-5	88
Our approach	**MADBase**	**CNN_5**	**99.17**
Our approach	**MADBase**	**$\text{HYBRID}_{5,3}$**	**99.35**
El-Sawy et al. [11]	AHCD database	CNN	94.9
Shams et al. [20]	AHCD database	DCNN-SVM	95.07
Altwaijry and Al-Turaiki [4]	AHCD database	CNN	94.9
Our approach	**AHCD database**	**CNN_5**	**95.23**
Our approach	**AHCD database**	**$\text{HYBRID}_{5,3}$**	**97.00**

The advantage of hybrid models compared to SVM is evident from our overall experimental results. This is further confirmed by the p-values observed for the hypotheses tests H_1 and $H_{2,i}$ which provide clearly highly conclusive levels of significance as illustrated in the Table 5 for all experimented datasets. The advantage is more significant with the application on AHCD where the difference in average rates is more evident between SVM and the combined hybrid approaches. Empirically, we can conclude that the advantage of hybrid approaches compared to SVM is more highlighted when it comes to complex datasets in term of number of classes.

CNN_3 gives the best recognition rates among the 3 CNN architectures applied to MNIST, Letters and Balanced. This same CNN_3 architecture, when combined with SVM, in this case the HYBRID_3 model, gives even better results. $\text{HYBRID}_{3,3}$ gives a recognition rate of 99.47% which is higher by 0.10% than the recognition

rate obtained by CNN_3. For Letters, $HYBRID_{3,2}$ provided an improvement of 1.31% compared to CNN_3. For Balanced, $HYBRID_{3,1}$ provided an improvement of 1.57% compared to CNN_3. For AHCD, the CNN_5 gives a recognition rate of 95.23%. When combined with SVM, the rate is improved by 1.78%, thus giving a rate of 97%. This rate is the best of overall results with AHCD. For MADBase, CNN_5 gives the best rate of 99.17% among all other CNN architectures. This same CNN, when combined with SVM, under the model $HYBRID_{5,3}$, gives the best result among all other combinations with the recognition rate of 99.35%. Empirically, for each $HYBRID_i$, the average of the obtained recognition rate is significantly higher than the recognition rate obtained by the underlying CNN_i. This is further confirmed by the p-values observed for $H_{3,i}$ hypothesis tests in the Table 5.

A comparison with state-of-the-art reported work on the EMNIST dataset, as illustrated by table 6, shows that the recognition rates of our hybrid CNN-SVM algorithms are slightly lower than classic classifiers used by Biswas and Islam [6] and Cavalin and Oliveira [7]. Biswas and Islam [6] achieved a recognition rate on MNIST using CNN of 99.53%. Cavalin and Oliveira [7] obtained a recognition rate for Letters by means of CNN of 93.63%. However, the hybrid CNN-SVM model remains competitive when dealing with letters and digits. Also, it outperforms all the other models when considering all the remaining splits, especially the Balanced, the By Merge, and the By Class ones, which contain much more characters and balanced classes. This confirms the general insight from the analysis of the previous table, postulating that the hybrid CNN-SVM outperforms non-hybrid models (CNN, SVM, and other models like linear classifiers, OPIUM, etc.).

As for arabic characters, a comparison of the hybrid architecture's performance to classic deep neural network architectures, as illustrated by table 7, shows that the hybrid approach always reaches a better recognition rate.

5 Conclusion

In this work, we performed statistical profiling of deep neural networks' effectiveness. We compared the performance of a hybrid model of convolutional neural networks and support vector machines to state-of-the-art CNN and SVM architectures. Conducted experiments on reference Latin and Arabic handwritten datasets confirm that the hybrid convolutional network outperforms and reaches better precision rates. Based on this funding, we can use hybrid neural network architectures instead of pure neural network models. This has two advantages: 1) we can achieve better precision rates, and 2) we can explain obtained results. For instance, authors in [23] combines a convolutional neural network with a decision tree. System output can be explained by a set of if/else-if rules.

Notes that even if we mix our deep neural network with a black box model, which is the case for SVMs, explainability can be achieved due to previously conducted research to whiten those black boxes [18]. In other words, applying a Granular

Computing technique, described in [18] on the SVM, which is the output of the hybrid CNN-SVM architecture, makes the hybrid neural network explainable.

Currently, we are testing different deep neural network architectures with a bunch of classic AI models to draw on an overall portrait of hybrid deep neural network explainability.

Acknowledgements The authors would like to thank Natural Sciences and Engineering Research Council of Canada (**NSERC**) for financing this work.

References

[1] S. Ahlawat and A. Choudhary. Hybrid cnn-svm classifier for handwritten digit recognition. *Procedia Computer Science*, 167:2554–2560, 2020.

[2] J. H. Alkhateeb. Handwritten arabic digit recognition using convolutional neural network. *International Journal of Communication Networks and Information Security*, 12(3):411–416, 2020.

[3] R. S. Alkhawaldeh. Arabic (indian) digit handwritten recognition using recurrent transfer deep architecture. *Soft Computing*, 25(4):3131–3141, 2021.

[4] N. Altwaijry and I. Al-Turaiki. Arabic handwriting recognition system using convolutional neural network. *Neural Computing and Applications*, 33(7): 2249–2261, 2021.

[5] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74, 2021.

[6] A. Biswas and M. S. Islam. An efficient cnn model for automated digital handwritten digit classification. *Journal of Information Systems Engineering and Business Intelligence*, 7(1):42–55, 2021.

[7] P. Cavalin and L. Oliveira. Confusion matrix-based building of hierarchical classification. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 23rd Iberoamerican Congress, CIARP 2018, Madrid, Spain, November 19-22, 2018, Proceedings 23*, pages 271–278. Springer, 2019.

[8] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[9] E. Dufourq and B. A. Bassett. Eden: Evolutionary deep networks for efficient machine learning. In *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, pages 110–115. IEEE, 2017.

[10] A. El-Sawy, H. El-Bakry, and M. Loey. Cnn for handwritten arabic digits recognition based on lenet-5. In *International conference on advanced intelligent systems and informatics*, pages 566–575. Springer, 2016.

[11] A. El-Sawy, M. Loey, and H. El-Bakry. Arabic handwritten characters recognition using convolutional neural network. *WSEAS Transactions on Computer Research*, 5(1):11–19, 2017.

[12] P. Ghadekar, S. Ingole, and D. Sonone. Handwritten digit and letter recognition using hybrid dwt-dct with knn and svm classifier. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pages 1–6. IEEE, 2018.

[13] F. Lauer, C. Y. Suen, and G. Bloch. A trainable feature extractor for handwritten digit recognition. *Pattern Recognition*, 40(6):1816–1824, 2007.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] M. Loey, A. El-Sawy, and H. El-Bakry. Deep learning autoencoder approach for handwritten arabic digits recognition. *arXiv preprint arXiv:1706.06720*, 2017.

[16] X.-X. Niu and C. Y. Suen. A novel hybrid cnn–svm classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4):1318–1325, 2012.

[17] W. Pan, T. D. Bui, and C. Y. Suen. Isolated handwritten farsi numerals recognition using sparse and over-complete representations. In *2009 10th international conference on document analysis and recognition*, pages 586–590. IEEE, 2009.

[18] S. S. Samuel, N. N. B. Abdullah, and A. Raj. Interpretation of svm to build an explainable ai via granular computing. *Interpretable Artificial Intelligence: A Perspective of Granular Computing*, pages 119–152, 2021.

[19] G. K. Sarkon, B. Safaei, M. S. Kenevisi, S. Arman, and Q. Zeeshan. State-of-the-art review of machine learning applications in additive manufacturing; from design to manufacturing and property control. *Archives of Computational Methods in Engineering*, 29(7):5663–5721, 2022.

[20] M. Shams, A. Elsonbaty, W. ElSawy, et al. Arabic handwritten character recognition based on convolution neural networks and support vector machine. *arXiv preprint arXiv:2009.13450*, 2020.

[21] A. Sherif and E.-S. Ezat. The arabic handwritten digits databases: Adbase & madbase. URL https://datacenter.aucegypt.edu/shazeem/.

[22] M. Szarvas, A. Yoshizawa, M. Yamamoto, and J. Ogata. Pedestrian detection with convolutional neural networks. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pages 224–229. IEEE, 2005.

[23] Q. Zhang, Y. Yang, H. Ma, and Y. N. Wu. Interpreting cnns via decision trees. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6261–6270, 2019.