République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Larbi Tébessi - Tébessa
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie
Département : Mathématiques et Informatique

Mémoire de fin d'étude
Pour l'obtention du diplôme de MASTER
Domaine : Mathématiques et Informatique
Filière : Informatique
Option : Systèmes et Multimedia

Thème

Identification de Script de textes à partir d'images vidéo

Présenté Par :
Chabou Khaled

Devant le jury :

Mr. Djeddi Chawki	*MCA*	*Univérsité Larbi Tébessi*	*Président*
Mr. Rouabhia Djaber	*MAA*	*Univérsité Larbi Tébessi*	*Examinateur*
Mr. Gattal Abdeljalil	*MCA*	*Univérsité Larbi Tébessi*	*Encadreur*

Date de soutenance : 23/Juin/2019

Résumé

Les méthodes de réseaux de neurones artificiels, en particulier l'apprentissage en profondeur, ont réalisé d'importants succès dans le domaine de la vision par ordinateur.

Ce projet de fin d'étude propose un système d'identification de script vidéo basé sur une méthode d'apprentissage en profondeur appelée réseau de neurones convolutifs qui réussissent dans diverses applications d'identification d'images. Les évaluations ont été effectuées sur la base de données appelées CSVI 2015 soumise lors de la compétition ICDAR 2015.

Le réseau de neurones convolutifs peut être présenté avec différentes architectures pour atteindre l'objectif d'apprentissage. Dans notre travail, nous avons proposé une architecture composée d'un ensemble de couches convolutives, de couches ReLu, de couches de max pooling et de couches entièrement connectées.

Nous démontrons de manière expérimentale que la précision peuvent être augmentés lors de l'utilisation optimale des paramètres dans la même architecture du réseau de neurones convolutifs et par la modification de l'architecture du CNN.

Mot clés : apprentissage en profondeur, réseaux de neurones artificiels, vision par ordinateur, identification de script vidéo, réseaux de neurones convolutifs, CSVI 2015.

Abstract

Artificial neural network methods, especially deep learning have achieved significant successes in the field of vision computer.

This end of study project work proposes a system for the Video Script Identification based on a deep learning method called convolutional neural network which are successful in various image identification applications, the assessments performed on the database called CSVI 2015 introduced during the ICDAR 2015 competition.

the convolutional neural network can be presented with deferent architectures to achieve the learning goal,in our work we proposed a architecture that consist of a sets of convolution layers, ReLu layers, max pooling layers and fully connected layers.

We experimentally demonstrate that the accuracy can be increased when using optimal parameters in the same architecture of the convolutional neural network, and that the modification of CNN architecture.

Keywords: Deep learning, artificial neural networks, vision computer, Video Script Identification, convolutional neural network, CSVI 2015

ملخص

حققت أساليب الشبكات العصبية الاصطناعية ، وخاصة التعلم عمقا ، نجاحات كبيرة في مجال الرؤية بالكمبيوتر.

يقترح مشروع نهاية الدراسة هذا نظامًا للتعرف على صور منتقاة من الفيديو يستند على طريقة تعلم عميقة تسمى CNN الناجحة في تطبيقات التعرف على الصور المختلفة ، تتعلم هذه الشبكة من مجموعة بيانات تسمى CSVI 2015 تم عرضها في مسابقة ICDAR 2015.

يمكن العمل بشبكة CNN باستخدام عدة هندسيات مختلفة لتحقيق هدف التعلم ، في عملنا اقترحنا هندسة تتكون من مجموعة من طبقات الالتفاف ، طبقات ReLu ، طبقات التجميع والطبقات المتصلة بالكامل.

لقد أثبتنا بشكل تجريبي أن دقة الشبكة يمكن زيادتها عند اختبار إعدادات مختلفة في نفس هندسة الشبكة، وتغيير البنية يمكن أن يحسن دقة التعلم.

الكلمات المفتاحية : التعلم عمق ، الشبكات العصبية الاصطناعية ، رؤية الكمبيوتر ، CNN، CVSI 2015

Dédicace

A mes parents pour leur indéfectible soutien.

A mon frère et mes sœurs.

A mon oncle Khamès et son épouse

A ma tante Sayda.

A tous mes amis.

Remerciements

Je tiens tout d'abord à remercier mon encadreur, le Docteur Abdeldjalil Gattal pour m'avoir fait confiance non seulement en acceptant de m'encadrer mais aussi pour m'avoir proposé ce sujet. Malgré ses multiples préoccupations (enseignement, recherche et charges administratives), il a su guider ma recherche en me conseillant et en m'inspirant tout en me laissant une grande liberté. Je dois également le remercier pour sa disponibilité et sa patience. Qu'il trouve ici le témoignage de toute ma gratitude et de ma reconnaissance.

Je dois également remercier le Dr Djeddi Chawki et le Docteur Rouabhia Djaber non seulement pour m'avoir honoré en acceptant d'être les membres de mon jury mais aussi pour le savoir qu'ils nous ont transmis pendant le cursus de formation.
Mes remerciements vont également à tous mes autres enseignants de 1ière et 2ième années du master pour leurs efforts en matière de transmission de connaissances , d'une part et aussi ,pour certains d'entre eux, pour leur ouverture d'esprit, du moins avec moi, afin de satisfaire ma curiosité en acceptant de répondre à mes nombreuses questions, parfois en dehors de la salle de cours.

Je ne peux également oublier les responsables de la société des ciments de Tébessa pour avoir facilité mon stage.

Enfin ces remerciements ne seraient pas complets sans mentionner ma mère et mon père ainsi que mon frère et mes sœurs et mon oncle Khamès et son épouse ainsi que ma tante Sayda.

Liste des figures

Liste des tableaux

Table des matières

Introduction générale

Introduction générale

Le fonctionnement de notre cerveau facilite la vision et l'identification des objets, des textes et des images. Les êtres humains ne rencontrent aucune difficulté dans leur vie quotidienne quand cela se produit.

Au cours des dernières années, la recherche scientifique a essayé de reproduire le système de vision humain dans des ordinateurs. Cette branche de l'informatique est connue sous le nom de vision par ordinateur. Néanmoins, les ordinateurs voient le monde différemment des êtres humains.

De nombreux systèmes ont été implémentés pour résoudre des problèmes individuels qui ont fait d'énormes progrès dans le domaine de la vision par ordinateur.

L'identification de script vidéo est l'un de ces systèmes. L'identification comporte généralement quatre phases: le prétraitement, la segmentation, l'extraction de caractéristiques et la classification.

Problématique

L'identification du script vidéo est l'une des tâches les plus difficiles dans le domaine d'analyse de documents.

Son objectif principal est d'extraire des informations à partir d'images vidéo ou d'images réalisées par des caméras. Cependant, il existe de nombreux problèmes pour construire un système universel capable de traiter et d'identifier le contenu d'images vidéo déférentes. En effet, en réalité, il est difficile d'obtenir une précision acceptable dans un système réel d'identification d'images quel que soit les problèmes qui peuvent affectée ces performances du système développé.

Ces problèmes complexes liés à la source des images proviennent de causes différentes et de paramètres différents, ce qui rend la situation difficile et rend le système sensible à tout paramètre inattendu. Ces problèmes divers nécessitent de nombreuses ressources en termes de temps et de capacités de calcul. Par conséquent, ce type de problème reste encore ouvert à la recherche. De ce fait « beaucoup de problème de traitement d'image ne sont que partiellement résolus. Tel est le cas de la reconnaissance des formes qui reste un sujet récurrent et difficile » [01].

Néanmoins, une évolution importante dans le traitement de l'image a été réalisée depuis l'apparition de l'apprentissage en profondeur.

Notre travail s'inscrit dans cette perspective de l'identification du script vidéo.

Structure du mémoire

Ce mémoire se présente sous la forme de quatre chapitres :

- Le **chapitre 1** présente un aperçu sur les notions de base du système d'identification de scripts.

- Le chapitre suivant (**chapitre 2**) est consacré à l'étude de l'apprentissage en profondeur et en particulier la méthode du réseau de neurones convolutifs et son l'intérêt dans le domaine de la classification et de l'identification de scripts.
- Le **chapitre 3** présente un état de l'art en présentant les méthodes les plus récentes de classification de scripts vidéo ainsi que les compétitions organisées dans ce domaine.
- Dans le **chapitre 4,** nous procéderons à l'expérimentation de notre travail et nous discuterons les différents résultats obtenus.

Notre travail sera achevé par une conclusion générale.

Chapitre 1

Le système d'identification de script

Chapitre 1: Le système d'identification de script

Introduction

Les systèmes d'identification de scripts [02][03] est un domaine qui identifie le texte dans une image numérique. Il est couramment utilisé pour reconnaître les textes imprimés ou manuscrits dans les documents numérisés.

Ce chapitre a pour objet de décrire les différentes étapes suivies dans les systèmes d'identification de scripts à savoir le prétraitement, la segmentation, l'extraction des caractéristiques et la classification.

En effet, après l'acquisition de l'image à l'aide d'un scanner ou une caméra, l'identification du script se déroule selon quatre étapes.

1. Le prétraitement

Le prétraitement est un processus d'amélioration de l'image pour faciliter le bon déroulement des étapes suivantes.

Notons qu'il existe de nombreuses techniques disponibles pour effectuer ce prétraitement sur les images. La sélection des techniques à adopter dépend du type d'application désirée.

Cependant le prétraitement peut être contourné dans des cas où on a besoin de préserver l'image originale et ces caractéristiques (couleurs, forme, ...) ou dans d'autres situations similaires. Il convient de signaler que les techniques de prétraitement les plus répandues sont :

1.1. La mise en niveau de gris

Cette transformation permet, dans le cas où l'image source est en couleur, de la transformer en une image en niveaux de gris [04][05] représentée uniquement sur une seule dimension et la valeur d'un pixel est comprise entre 0 et 255 (codée sur 8 bits), ce qui fait des niveaux de gris un élément essentiel, car les informations RVB ou en couleurs ont une propriété tridimensionnelle, ce qui rend le traitement du signal trop volumineux et lourd.

Figure 1-1: Transformation d'une image couleur vers une image en niveaux de gris.

1.2. La binarisation

La binarisation [06][07] d'images consiste à convertir une image de niveaux de gris en une image en noir et blanc composée de 2 valeurs 0 et 1.

La binarisation des images est un processus important pour l'analyse de l'image. Elle est utilisée, souvent, comme une étape préliminaire avant d'identifier le script dans l'image

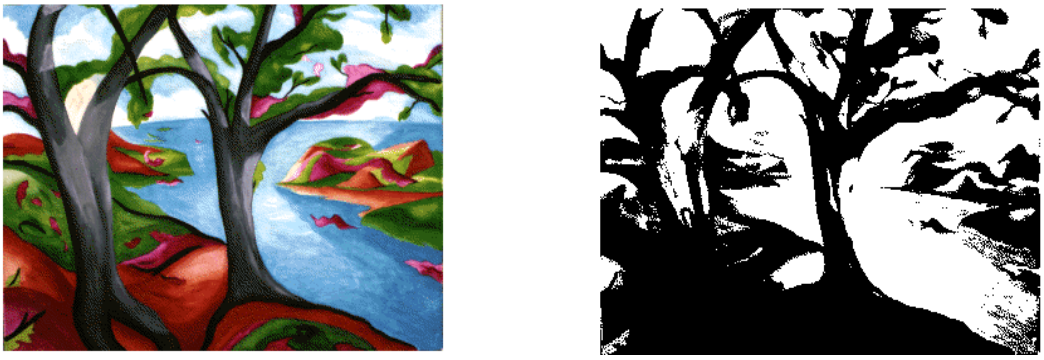

Figure 1-2: Transformation d'une image couleur en image binaire.

1.3. La réduction de bruit

Les images numériques sont sujettes à divers types de bruit. Qui est le résultat d'erreurs dans le processus d'acquisition d'images qui entraîne des valeurs de pixels qui ne reflètent pas les intensités réelles [08].

Le bruit peut s'infiltrer de plusieurs manières dans une image. Il se manifeste par des variations aléatoires de la luminosité ou des informations de couleur dans les images et constitue généralement un aspect du bruit électronique [09].

La réduction de bruit est donc l'une des techniques essentielles de prétraitement pour résoudre ce type de problèmes.

1.4. Le zonage

Dans le domaine de système d'identification de scripts, le zonage [06][10][11][12] d'images est une technique très utilisée pour l'extraction des caractéristiques.

Elle est principalement utilisée pour extraire des caractéristiques spéciales d'un motif.

Le zonage implique la division d'une image en plusieurs nombres prédéfinis de zone puis de chacune de ces zone (en tailles 2× 2, 3 × 3, 4 × 4 ...etc.) on peut extraire des caractéristiques.

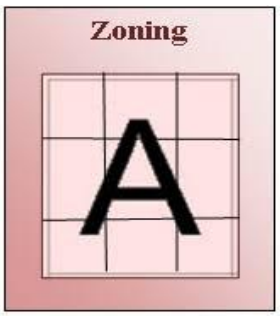

Figure 1-3: Zonage de caractère A [10]

2. La segmentation

La segmentation [13] est le processus de partitionnement de l'image en plusieurs régions homogènes et significatives pour simplifier la représentation et la rendre plus utile pour l'analyse et les interprétations. Le résultat de cette division est la constitution d'un nombre de régions qui partagent des attributs similaires (comme la couleur,), pour minimiser les erreurs de classification et réduire l'incertitude statistique.

3. L'extraction des caractéristiques

Dans la construction d'un système d'identification de script, une étape d'extraction de caractéristiques [14][15] est essentielle pour réduire le volume d'informations, afin que le système puisse différencier un objet d'un autre, et de ne nous fournir que des informations pertinentes.

On peut approcher cette étape d'extraction par plusieurs techniques qui, en majorité, partagent un inconvénient commun « la perte d'informations » ce qui nécessite le choix d' une approche économique et équilibrée des informations et des caractéristiques à extraire. Cette approche présente plusieurs avantages comme :

-la réduction du temps d'apprentissage.

-la réduction de la taille des bases (bases d'apprentissage)

- l'amélioration des performances dans la phase de classification.

3.1. Les caractéristiques globales

Les caractéristiques globales signifient toutes les caractéristiques au niveau superficiel de l'image, par exemple la hauteur, la largeur, la couleur et la texture etc.

Ils sont moins sensibles au bruit. Néanmoins, leur utilisation en exclusivité cause une perte des caractéristiques locales. Par conséquent, elles sont utilisées en combinaison avec d'autres caractéristiques afin de régler le problème.

3.2. Les caractéristiques locales

Ces caractéristiques concernent la recherche précise sur une partie de l'image. Ils s'intéressent aux détails comme les points, les contours etc. ……

Il est d'une importance toute particulière de souligner que l'analyse exclusive de ces caractéristiques risque de faire perdre le sens global de l'image.

4. La classification

La classification [16] est le processus qui suit l'extraction des caractéristiques. La classification est une méthode de prédiction des classes de points de données. Les techniques de classification fonctionnent selon un protocole de décision qui permet de catégoriser des objets selon certains critères d'optimisation.

4.1. L'apprentissage non supervisé

La tâche de classification non-supervisée [17] a pour but d'organiser un ensemble d'objets sans une classification fournie préalablement afin de structurer la base de reconnaissance des formes récurrentes.

L'apprentissage non supervisée consiste à apprendre à classer sans superviser le processus qui débute sans nombre ou sans définition de classes.

L'algorithme de classification doit tenter de structurer et former des groupes de caractéristiques sur des communalités aperçues par l'algorithme.

4.2. L'apprentissage supervisé

L'apprentissage supervisé [17] consiste à sculpter d'un complexe de données fournies et dotées de pré classification les caractéristiques pertinentes et nécessaires pour permettre de classer et organiser une nouvelle donnée.

Contrairement à l'apprentissage non supervisé, dans l'apprentissage supervisé, le processus commence avec des classes identifiées et définies préalablement.

Les modèles de classification les plus communs sont **SVM, ANN, KNN et CNN.**

Conclusion

Ce chapitre nous a donnés une vue globale et claire sur le système de reconnaissance en mettant en lumière les informations pertinentes comme les prétraitements, la segmentation, l'extraction des caractéristiques et la classification.

Chapitre 2

Le Deep Learning : L'apprentissage en profondeur

Chapitre 2: Le Deep Learning : L'apprentissage en profondeur

Introduction

L'apprentissage en profondeur est un sous-domaine de l'apprentissage automatique en intelligence artificielle. « La vision par ordinateur est une discipline relevant de l'intelligence artificielle et s'appuyant sur des tâches d'analyse et de traitement d'images. Elles permettent à un ordinateur de décrire ce qu'il voit en imitant le fonctionnement de la vision humaine. Les tâches comprennent par exemple, la classification pour associer un objet à une classe prédéfinie ; la reconstruction pour compléter, voir recréer une partie d'image manquante ; ou la détection pour identifier des objets spécifiques dans une image » [18]

Ce chapitre a pour objet de présenter les concepts ainsi que l'architecture de l'apprentissage en profondeur.

1. Réseau de neurones artificiels

Un réseau de neurones artificiel [19][22] est un ensemble de nœuds de traitement interconnectés dont certains nœuds ont pour missions de recevoir les données (input unit) par contre d'autres se chargent de les livrer (output unit).

La fondation d'un réseau ANN est formée d'une interconnexion de neurones avec des liens dirigés et pondérés de sorte que les nœuds représentent les neurones et les liens pondérés renforcent la connexion et gluent les neurones.

Lorsque le nœud reçoit des données, il calcule la somme pondérée des entrées ensuite il applique à ces résultats une fonction d'activation (sigmoïde, ReLu, tanh)

Selon la connexion, on discerne deux types de réseaux : les réseaux acycliques et les réseaux récurrents (au moins un cycle)

- Les réseaux perceptron et les réseaux convolutifs comptent parmi les réseaux acycliques
- Le réseau a mémoire court et long terme (LSTM) est un réseau récurrent.

2. Perceptron

Perceptron [23][25] est un réseau de neurones artificiels qui peut être considéré comme l'une des formes les plus simples de réseaux neuronaux à anticipation et constitue un classifieur linéaire.

Le perceptron a été proposé par *Frank Rosenblatt en 1957* [26] et mis en œuvre pour la première fois en tant que logiciel pour l'IBM 704.

Dans le domaine des réseaux de neurones artificiels, les perceptrons sont également appelés réseaux de neurones artificiels à couche unique afin de les distinguer du perceptron multicouche plus complexe. Le principal défaut inhérent au perceptron est qu'il ne peut pas traiter des problèmes linéaires indivisibles.

Progressivement l 'utilisation du perception multicouche a permis de transcender cette limitation et permettre de classifier des groupes qui ne sont pas séparables de façon linéaire et à résoudre des problèmes hors de portée du perceptron monocouche.

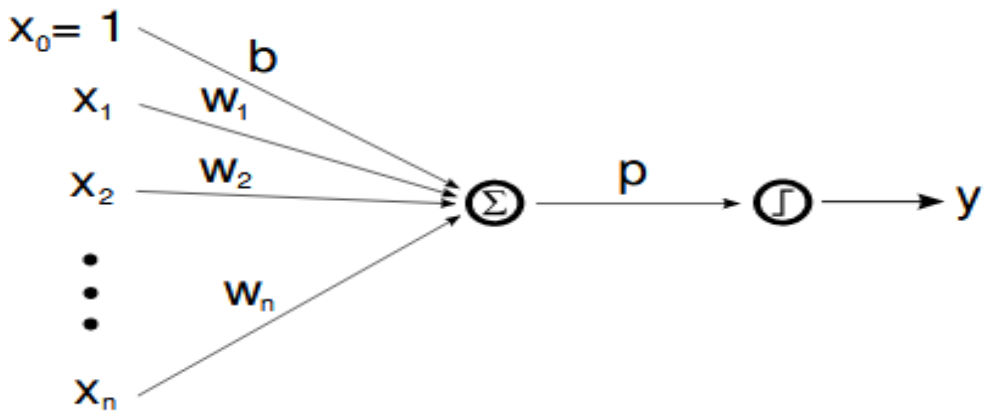

Figure 2-1: Modèle du perceptron [24].

2.1. Perceptron Multi Couches (MLP)

Un perceptron multicouche (MLP) [24] est un réseau neuronal artificiel. Il est composé de multiples perceptrons dans sa forme la plus pertinente. Il se compose de trois types de couches :

- Une couche d'entrée pour recevoir le signal,
- Une couche de sortie qui prend une prédiction concernant l'entrée,
- Et entre ces deux couches, un nombre arbitraire de couches cachées qui constituent le véritable générateur de calcul du MLP.

Potentiellement, il est possible d'augmenter le nombre des neurones de la couche cachée selon les besoins pour l'approximation de n'importe quelle fonction non linéaire.

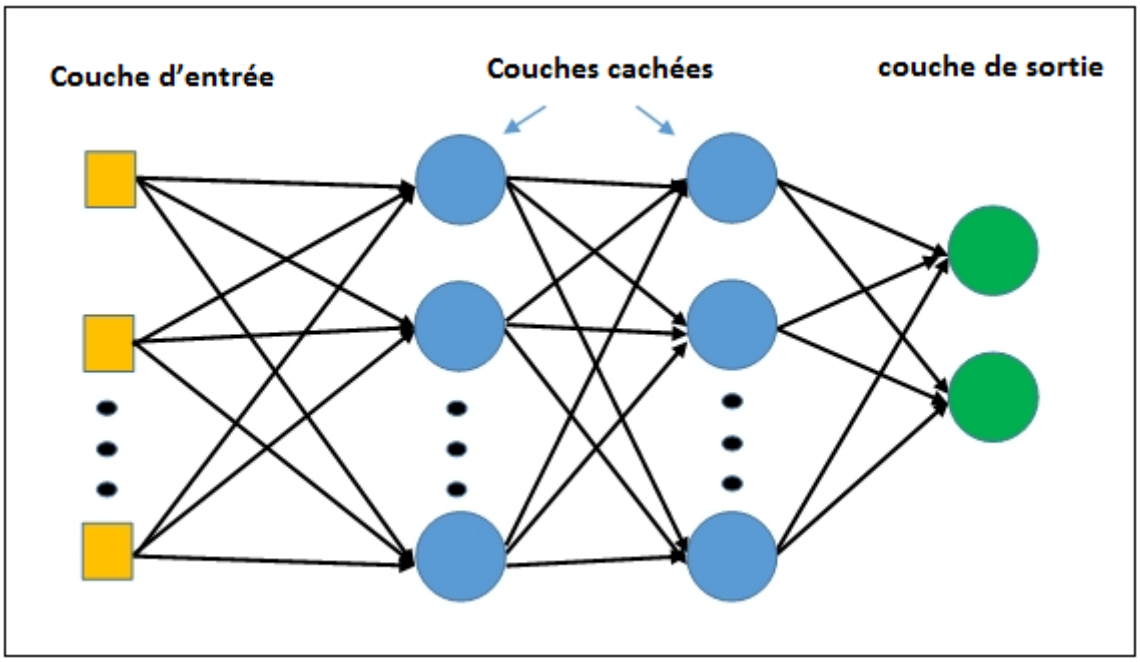

Figure 2-2: Architecture de MLP

3. L'architecture de l'apprentissage en profondeur

L'apprentissage en profondeur est un réseau de neurones constitué de plusieurs couches cachées utilisées pour apprendre des fonctionnalités plus complexes. Il s'agit de la principale différence entre l'apprentissage en profondeur et un réseau de neurones.

L'apprentissage de ces couches multiples rend le processus coûteux et nécessite une base de données volumineuse pour apprendre les caractéristiques complexes.

Un réseau de neurones artificiels ne permet que d'apprendre les poids d'un réseau avec une couche cachée, mais ne contient pas plusieurs nombres de ces couches et ne peut donc pas apprendre des caractéristiques complexes.

3.1. Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs sont un type particulier de réseaux de neurones multicouches.

Ils sont largement utilisés dans les problèmes de reconnaissance de formes.

Comparé à d'autres techniques, les réseaux de neurones convolutifs présentent notamment l'avantage de la reconnaissance de modèles visuels avec un traitement minimal et une variabilité extrême des modèles. Il est à noter aussi l'architecture exceptionnelle que nous traiterons en détail dans la sous-section suivante.

3.1.1. Architecture de réseaux de neurones convolutifs

Une architecture CNN est formée de couches distinctes qui transforment le volume d'entrée en un volume de sortie. Nous présentons quelques types de ces couches qui sont le plus couramment utilisés :

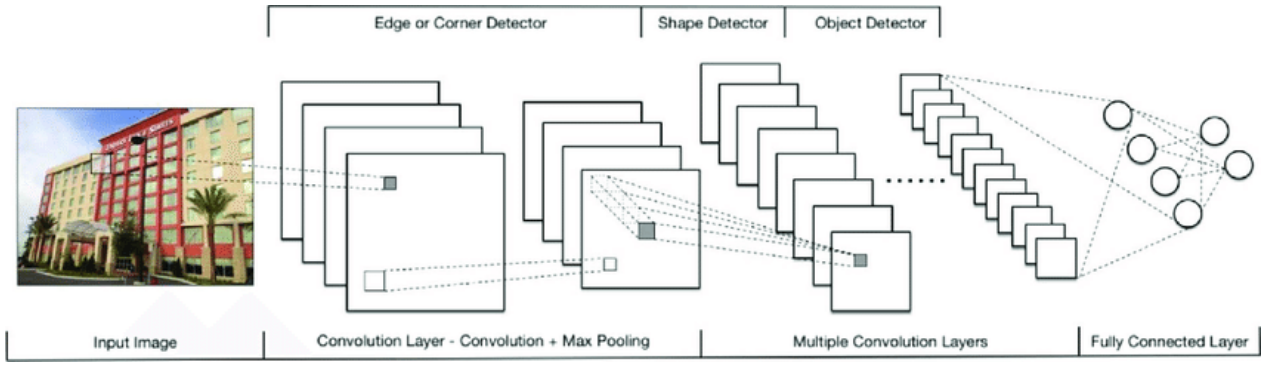

Figure 2-3: Une architecture de réseaux de neurones convolutifs

3.1.1.1. La couche convolutive

La couche Convolutive [32] applique des filtres sur ces entrés (exemple une image couleur), chaque filtre est petit en termes de dimensions de hauteur, de largeur et notamment de profondeur qui doit être la même que la profondeur d'entrée (les images couleurs ayant de profondeur de 3).

Le filtre se positionne initialement au coin supérieur gauche sur la matrice d'image. Il exécute une multiplication entre ces valeurs et celle de la section sélectionnée de la matrice, ce qui génère une nouvelle valeur.

Ce processus se répète sur chaque nouvelle position (le déplacement est d'un pas vers la droite). La collection de ces valeurs générées construit une matrice de résultats appelée la carte de caractéristiques (feature map).

Avant que l'étape de convolution soit effectuée, on doit définir trois paramètres qui contrôlent et affectent directement la taille de la carte de caractéristiques. Ces paramètres sont les suivants :

Premier paramètre : Profondeur de la couche (Depth)

La profondeur de la carte de caractéristiques est identique au nombre de filtres que nous utilisons pour l'opération de convolution.

La convolution effectuée sur l'image à l'aide de trois filtres distincts produit trois cartes de caractéristiques différentes (Figure 2-4).

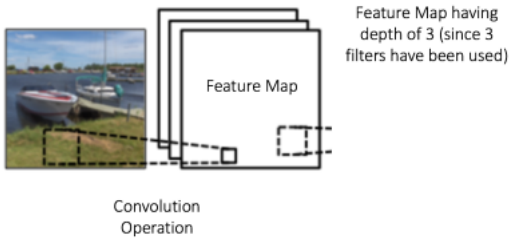

Figure 2-4: Le Profondeur d'une carte de caractéristiques.

Deuxième paramètre : Le pas (Stride)

Le pas (stride) est le nombre de pixels par lequel on glisse notre filtre sur la matrice d'image. Le pas est par défaut 1.

Notons que l'utilisation d'un pas plus grand produira des cartes de caractéristiques plus petites.

Troisième paramètre : La marge à zéro

Le remplissage (padding) [30] est une marge de zéro qui est placée autour de l'image.

Dans de nombreux cas, le filtre ne se glisse pas parfaitement sur la matrice d'image. Cela nous pousse à choisir entre deux options :

- Remplir (padding) l'image avec des zéros (zéro-padding)
- Conserver uniquement une partie valide de l'image en supprimant la partie où le filtre ne la couvre pas.

On note que l'utilisation du remplissage permet de résoudre un autre problème pertinent puisque si nous continuons à appliquer des couches convolutives, la taille du volume diminuera plus rapidement.

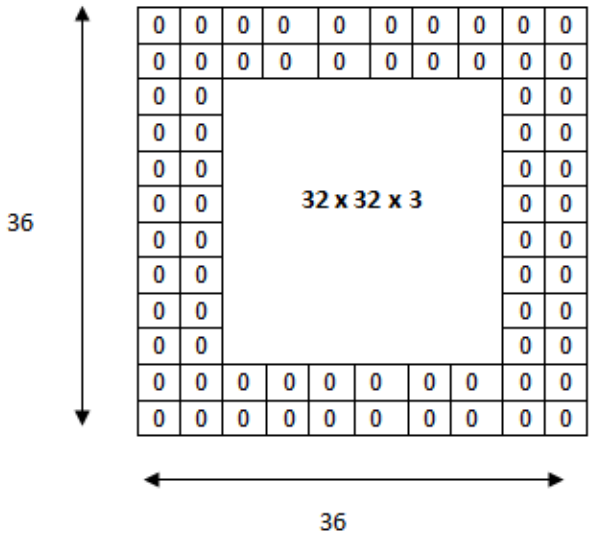

Figure 2-5: Remplissage de l'image avec des zéros

3.1.1.2. La couche de pooling

La couche de pooling [32][33][34][35] prend un filtre (normalement de taille 2x2) et un pas de même longueur. Il s'applique ensuite au volume d'entrée et génère le nombre maximal dans chaque région autour de laquelle le filtre se convole.

L'idée derrière l'utilisation de pooling dans les réseaux de neurones convolutifs est que l'emplacement exact d'une entité n'est pas aussi important que son emplacement relatif par rapport aux autres entités. Cette couche réduit, également, considérablement la dimension spatiale (la longueur et la largeur mais pas la profondeur) du volume d'entrée.

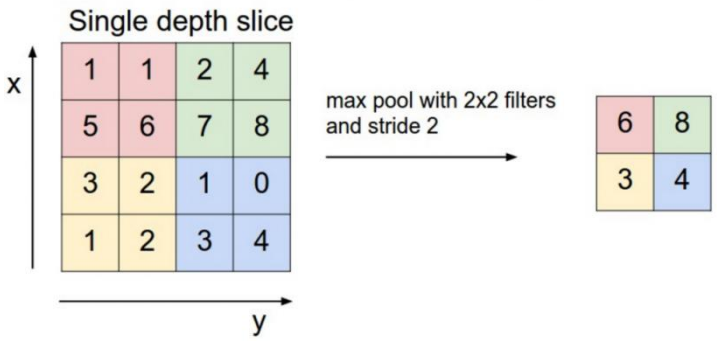

Figure 2-6: La fonction Max pooling

3.1.1.3. Les couches de correction

Le ReLu (Rectified Linear Unit) est l'une des fonctions d'activation les plus utilisée dans l'apprentissage en profondeur. Elle introduit la non-linéarité dans le réseau convolutif et elle est généralement appliquée après la couche convolutive.

La couche ReLu modifie toutes les valeurs négatives des entrées à zéro en appliquant la fonction f(x) [36] à toutes les valeurs du volume d'entrée.

$$f(x) = \begin{cases} 0 \; si \; x < 0 \\ x \; si \; x \geq 0 \end{cases}$$

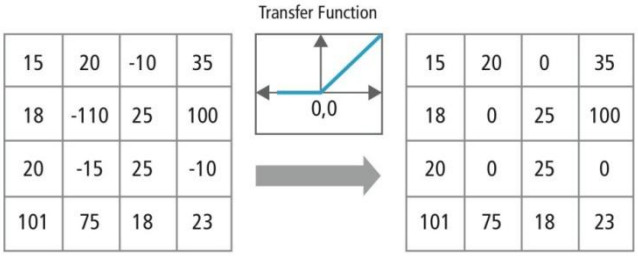

Figure 2-7: La fonction ReLu

3.1.1.4. Les couches entièrement connectées

Après plusieurs couches de pooling et de convolution, l'apprentissage des caractéristiques sur le réseau est terminé. Notre réseau passe maintenant à la classification via des couches entièrement connectées (fully connected layers).

Chaque neurone d'une couche se connectant à l'autre, ce qui signifie que cette couche est connectée à toutes les activations précédentes.

Cette couche prend un volume d'entrée et génère un vecteur à N dimensions où N est le nombre de classes connues par le réseau.

3.2. Réseaux de neurones récurrents

Les réseaux de neurones récurrents RNN [22][27][28] ont été créés dans la décennie 1980 mais l'exploitation de leur potentiel réel n'a ressurgit que récemment en raison de l'augmentation de la puissance de calcul disponible. Dans un réseau neuronal traditionnel, on suppose que toutes les entrées (et les sorties) sont indépendantes les unes des autres, par contre les réseaux de neurones récurrents relient l'information contextuelle des données par une connexion en boucle ce qui permet de prendre en compte la prédiction d'informations séquentielles.

Ce type d'architecture donne un contrôle flexible sur les séquences d'entrées/sorties, ce qui explique leur prévalence et leur valeur dans des domaines comme la traduction automatique ou la reconnaissance vocale, ...

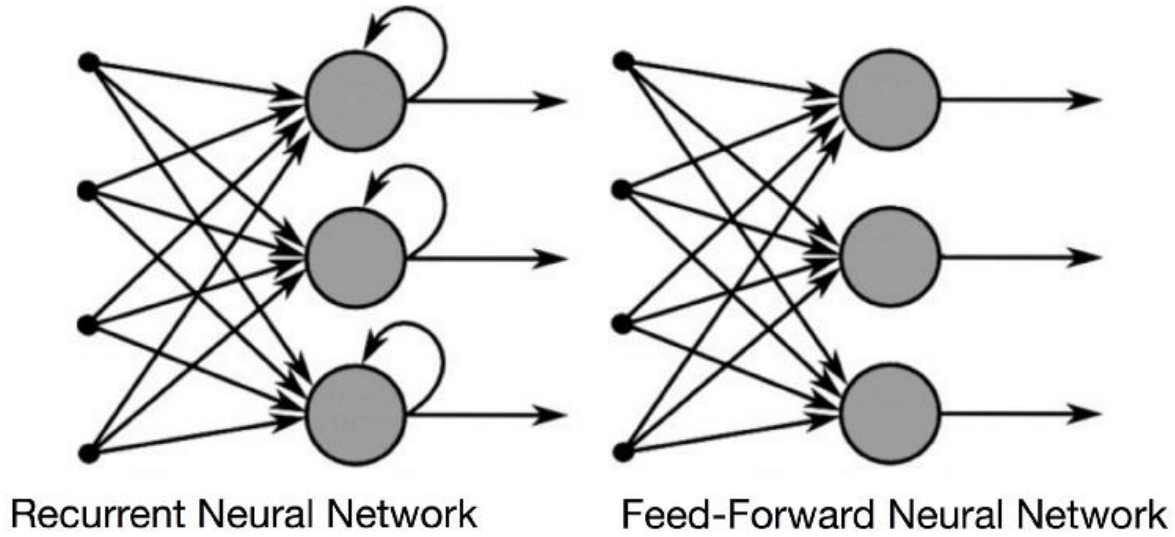

Figure 2-8: Réseaux de neurones récurrents

3.3. Réseau Hopfield

Un réseau Hopfield [37] est un réseau récurrent monocouche, c'est-à-dire que chaque neurone est connecté à un autre neurone mais pas à lui-même. Il a été introduit En 1982 par John Hopfield démontrant un réseau de neurones artificiels capable de stocker et de récupérer la mémoire en s'inspirant du cerveau humain.

Le concept est dérivé de la neurobiologie et de la psychologie et connu sous le nom de mémoire associative.

Il s'agit d'un réseau constitué de neurones à deux états (-1 et 1, ou 0 et 1).

Chaque unité agit à la fois comme une entrée et une sortie du réseau.

On note qu'après plusieurs mises à jour, l'état de chaque nœud se stabilise. (Convergence des nœuds)

3.4. La Machine de Boltzmann

La machine de Boltzmann [29] a été inventée en 1985 par le professeur Geoffrey Hinton, un pionnier dans le domaine de l'apprentissage en profondeur

La machine de Boltzmann est un modèle génératif non supervisé disposant d'une couche d'entrée (partie visible) et d'une ou plusieurs couches masquées (la partie masquée).

Son objectif principal est l'optimisation.

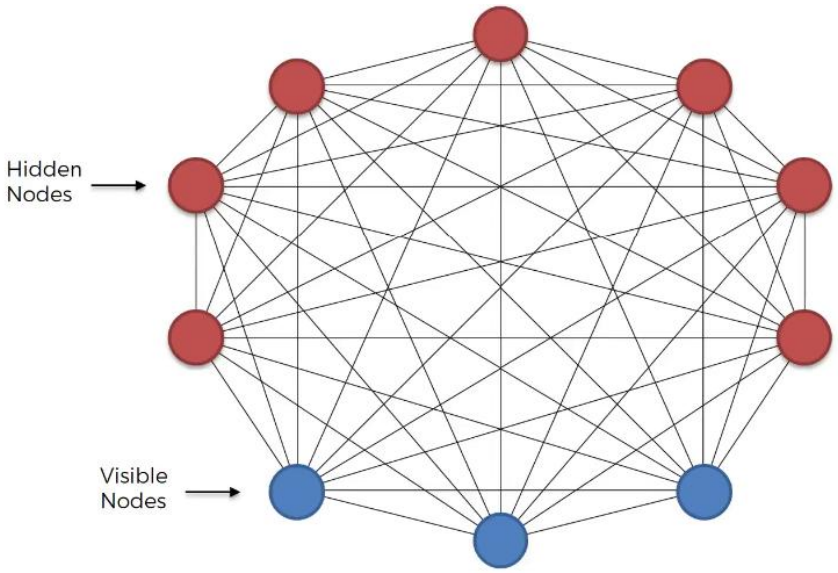

Figure 2-9: La machine de Boltzmann

La machine de Boltzmann se distingue par son architecture qui utilise de réseaux des neurones connectés non seulement à d'autres neurones situés dans d'autres couches, mais également à des neurones appartenant à la même couche, c'est-à-dire que tout est connecté à tout, chaque neurone partage une connexion bidirectionnelle avec l'autre.

En plus, le système traite les neurones avec égalité totale en utilisant à la fois les neurones cachées ou visibles afin de générer des informations.

3.5. LSTM (Long Short-Term Memory networks)

Les réseaux de neurones LSTM, qui correspondent à la mémoire à court et long terme, constituent un type particulier de réseaux de neurones récurrents, proposés en 1997 par Sepp Hochreiter et Jürgen Schmidhuber [31].

Une unité LSTM commune est composée d'une cellule, d'une porte d'entrée, d'une porte de sortie et d'une porte d'oubli. La cellule se souvient des valeurs sur des intervalles de temps arbitraires et les trois portes régulent le flux d'informations entrant et sortant de la cellule.

L'état de cette cellule module la sortie du réseau LSTM qui a une propriété très importante lorsque on veut que la prédiction du réseau de neurones dépende du contexte historique des entrées, plutôt que de la dernière entrée.

Les réseaux LSTM sont bien adaptés à la classification, au traitement et à la prédiction basée sur des données chronologiques faisant de LSTM le plus grand succès en matière d'Intelligence Artificielle (I.A) et ils sont applicables à des tâches telles que la reconnaissance d'écriture manuscrite connectée non segmentée ou la reconnaissance vocale.

Conclusion

Dans ce chapitre, nous avons axé notre travail surtout sur l'architecture CNN.

Enfin, à travers cette présentation de l'apprentissage en profondeur, nous pouvons dire que l'apprentissage en profondeur constitue une véritable révolution. En effet, l'ère numérique a entraîné une explosion de données sous toutes leurs formes, ces données sont cependant appelées « big data »

L'apprentissage en profondeur en tant qu'ensemble d'algorithmes utilisés dans l'apprentissage automatique permet de modéliser des abstractions de haut niveau des données à l'aide d'architectures de modèles.

L'utilisation de l'apprentissage en profondeur a permis également un essor fulgurant dans des domaines aussi variés tels que la médecine, les voitures autonomes, la reconnaissance des images et des écritures et la reconnaissance vocale. Malgré ces utilisations multiples et variées, l'utilisation de l'apprentissage en profondeur est sensée avoir une généralisation de son application dans tous les domaines. De ce fait, nous pensons qu'elle aura un avenir très prometteur.

Chapitre 3
Etat de l'Art

Chapitre 3 : Etat de l'Art

1. Introduction

Durant ces dernières années l'identification de script vidéo a progressé d'une manière importante. Plusieurs méthodes ont été proposées et cela a était possible, en partie, grâce aux compétitions organisées dans ce domaine.

Nous avons a identifié les travaux les plus pertinents dans ce chapitre et nous les avons les a présentés par ordre chronologique Ascendant. Nous avons aussi fourni des détails sur les systèmes utilisés et leurs taux de reconnaissance. Nous procéderons à la fin de ce chapitre à la présentation d'un aperçu sur les compétitions organisées en soulignant les aspects qui les différencient.

2. Travaux récents

Dans cette section, nous présentons différentes méthodes sur l'identification de script

2.1. Anguelos Nicolaou et al (2016)

Anguelos Nicolaou et al [42] ont introduit une méthode d'identification de script basée sur des caractéristiques de texture et un réseau de neurones artificiels.

La méthode proposée constitue en une étape de prétraitement, suivie de l'extraction de caractéristiques LBP (motifs binaires locaux) et de l'apprentissage d'un réseau de neurones artificiels (ANN) sur ces caractéristiques et avec KNN (pour la couche 1 ou 2).

Les couches intermédiaires de l'ANN sont ensuite utilisées comme modèle génératif pour effectuer la classification.

Les principales contributions sont l'introduction d'une méthode qui utilise un réseau de neurones profonds en plus de caractéristiques de texture pour l'identification de script.
C'est une méthode utilisée pour effectuer une identification purement visuelle de la langue, même pour les langues partageant le même script, et l'utilisation des activations du réseau de neurones à l'effet de générer des classificateurs plus adaptables.

Toutes les expériences de la méthode introduite ont été menées sur deux bases de données, la base CVSI-2015 (98,18%) et la base MLe2e (84,6%).

2.2. Louis Gomez et Dimosthenis Karatzas (2016)

Louis Gomez et Dimosthenis Karatzas [41] présentent une nouvelle méthode basée sur la combinaison de caractéristiques convolutives avec le classifieur NBNN (Naive-Bayes Nearest Neighbor). Cette méthode est basée sur la puissance de la représentation de caractéristiques locales et les utilise dans le schéma de classification capable de conserver le côté descriptif de petites parties d'image.

Les traits d'image sont extraits de l'image par une fenêtre glissante et introduits dans un réseau de neurones convolutifs simple (CNN). De cette façon, chaque ligne de texte est représentée par un nombre variable de descripteurs de traits d'image, utilisés pour calculer les distances entre les classes (Image to Class) et pour classer la ligne de texte d'entrée à l'aide du classifieur NBNN (Naive Bayes Nearest Neighbor).

Louis Gomez et Dimosthenis Karatzas ont également introduit une nouvelle base de données, connue sous l'appellation de «MLe2e».
Cette base de données contient un total de 711 images de scène couvrant quatre scripts différents (latin, chinois, kannada et hangul) et une grande variabilité d'échantillons de texte de scène.
Toutes les expériences rapportées ont été menées sur deux bases de données, en particulier la base CVSI-2015 (97,91%) et la base MLe2e (89,87%).

2.3. Jieru Mei et al (2016)

Jieru Mei et al [40] proposent une nouvelle approche pour l'identification de script de texte, combinant un réseau de neurones convolutifs (CNN) et un réseau de neurones récurrents (RNN) étant donné que le CNN génère des représentations d'image riches et le RNN analyse efficacement les dépendances spatiales à long terme.

Cette architecture consiste en une structure de couches convolutives et forme la première partie du modèle, ensuite dans une seconde partie les couches RNN qui prennent des cartes des caractéristiques de longueurs arbitraires produites par les couches CNN comme entrées pour exploiter les dépendances spatiales dans les images de script.

Après les couches RNN, les auteures ont ajouté une couche de pooling pour rassembler la sortie des couches RNN, la dernière partie étant le processus de classification utilisant une couche entièrement connectée.

Ce modèle est évalué sur des bases de données largement utilisée

SIW-13 (pour toutes les tâches) et CVSI2015 (Tache 4), avec un taux de succès de 92.75% et 94.2% respectivement.

2.4. Luis Gomez et al (2017)

En approfondissant leurs travaux précédents indiqués si- dessous (2.2), et afin de procéder à l'apprentissage des représentations plus discriminantes pour les patchs d'image individuels, les auteurs [39] proposent une nouvelle méthodologie d'apprentissage pour apprendre conjointement les représentations de patch et leur importance dans une image globale avec une mesure probabiliste.

Pour cela, ils ont procédé à l'apprentissage de CNN en utilisant un ensemble de réseaux conjoints et une fonction de perte qui prend en compte l'erreur de classification globale pour un groupe de N patch au lieu de ne rechercher qu'un seul patch d'image.

Ainsi, au moment de l'apprentissage, le réseau se voit présenter un groupe de N patchs partageant la même étiquette de classe et produit une distribution de probabilité unique sur toutes les classes. De cette façon, le réseau apprend mieux les représentations de patchs locaux et leur importance relative dans la tâche de classification globale des images.

Les expériences effectuées sur les trois bases de données pour la classification de texte de scène ont donné les résultats suivants : SIW-13 (94,8%), MLe2e (94,4%), CVSI2015 (97,2%).

2.5. Ankan Kumar Bhunia et al (2018)

Ankan Kumar Bhunia et al [38] ont proposé un Schéma global en trois étapes :

Étape 1:
Elle consiste à utiliser une structure de couches convolutives pour extraire les caractéristiques de l'image. Ces couches génèrent des vecteurs de caractéristiques qui seront utilisés lors de la prochaine étape.

Étape 2:

Après le CNN, un réseau d'attention (attention network) est utilisé suivi d'une couche softmax pour obtenir les poids de patch. Par la suite, les caractéristiques locales sont extraites à l'aide d'un vecteur de caractéristiques CNN et d'un vecteur d'attention (attention weights vector). Ces caractéristiques globales sont extraites du dernier état de cellule de l'unité LSTM.

Étape 3:

L'intégration des caractéristiques locales et des caractéristiques globales est une étape importante. Ankan Kumar Bhunia et al ont eu recours à un poids dynamique basé sur l'attention (attention based dynamic weighting) pour intégrer les caractéristiques locales et globales obtenues lors de la deuxième étape.

Une fois L'intégration est terminée, les scores de classification de chaque patch sont évalués à la fin en utilisant une couche entièrement connectée.

Dans ce travail, ils ont évalué un modèle proposé sur quatre bases de données de mots vidéo multilingues.

Nous présentons ci-après les bases de données utilisées et leurs taux de succès :

La CVSI-2015 avec un taux de (97,75%)
La SIW-13 avec un taux de (96,50%)
L'ICDAR-2017 avec un taux de (90,23%)
Et dernièrement la MLe2e avec un taux de (96,70%).

Le tableau ci-dessus montre le taux de chaque approche avec les bases de données utilisé.

Travaux	Base de donnée	Caractéristique	Classifieur	Taux
Ankan Kumar Bhunia et al (2018)	CVSI-2015	CNN	CNN	97,75%
	SIW-13			96,50%
	ICDAR-2017			90,23%
	MLe2e			96,70%
Louis Gomez et al (2017)	SIW-13	CNN	CNN	94,8%
	MLe2e			94,4%
	CVSI-2015			97,2%
Jieru Mei et al (2016)	SIW-13	CNN	CNN	92.75%
	CVSI2015			94.2%
Louis Gomez et Dimosthenis Karatzas (2016)	CVSI-2015	CNN	NBNN	97,91%
	MLe2e			89,87%
Anguelos Nicolaou et al (2016)	CVSI-2015	LBP	ANN+knn	98.18%
	SIW-10			84.6%

Tableau 1: Les résultats des travaux.

3. Compétitions Organisées

Les compétitions sur l'identification de script vidéo attirent de plus en plus d'intérêt ses dernières années. L'objectif général de la compétition est d'évaluer les méthodes récemment proposées.

Ces compétitions offrent une plateforme unique permettant aux chercheurs de partager de nouvelles méthodes et de coopérer pour faire face aux défis et problèmes persistants dans le domaine d'identification de scripts.

3.1. ICDAR2017 (CVSI 2017)

La compétition [43][45] s'est concentrée sur le développement d'algorithmes d'identification des scripts vidéo, quels que soient les scripts considérés.
Différentes combinaisons de 15 scripts sont prises en considération par la compétition. Ces scripts ont été soumis à certains paramètres et combinaisons qui sont organisés en six tâches :

Tâche 1: Identifier les scripts de (combinaisons de trois scripts, en gardant l'anglais et le hindi dans toutes les combinaisons), en fonction de leur utilisation dans le sous-continent indien.

Tâche 2: Identifier les scripts (combinaisons de trois scripts, conservant l'anglais dans toutes les combinaisons avec le chinois, le coréen, le japonais et le thaï), en fonction de leur utilisation en Asie du Sud.

Tâche 3: Combinaison de scripts utilisés dans le nord de l'Inde.
Tâche 4: Combinaison de scripts utilisés dans le sud de l'Inde.
Tâche 5: Combinaison de scripts romains et orientaux.
Tâche 6: Combinaison des quinze scripts.

3.2. ICDAR2015 (CVSI 2015)

Cette compétition [44][46] s'est concentrée sur différentes combinaisons de dix scripts indiens pris en considération.
Etant donné que l'Etat indien utilise généralement trois langues officielles, un document peut contenir un ou plusieurs de ces trois scripts.
Les participants ont été invités à accomplir quatre tâches différentes :

Tâche 1: Identifier les scripts de huit triplets de scripts différents (combinaisons de trois scripts, gardant l'anglais et l'hindi dans toutes les combinaisons), en fonction de leur utilisation dans le sous-continent indien.

Tâche 2: Identifier la combinaison de scripts utilisée dans le nord d'Inde. Cette tâche implique l'identification de sept scripts à savoir l'anglais, l'hindi, le bengali, l'oriya, le gujrathi, le punjabi et l'Arabe.

Tâche 3: Identifier la combinaison de scripts utilisée dans le sud de l'Inde. Cette tâche implique l'identification de cinq scripts, à savoir : Anglais (romain), hindi, kannada, tamoul et telegu.

Tâche 4: Identifier le script à partir de la combinaison des dix scripts est le défi de la tâche 4. Trois écritures utilisées dans le sud de l'Inde (c'est-à-dire Kannada, Tamil et Telugu) et six scripts utilisés dans le nord de l'Inde (c'est-à-dire hindi, bengali, oriya, gujrathi, punjabi et arabe) avec les scripts en anglais ont été pris en compte pour la tâche 4.

Cinq participants ont été soumis à la compétition, une brève description de leurs systèmes est présentée dans les lignes suivantes :

C-DAC, India [44]: Swapnil Belhe a participé aux quatre tâches du concours.
Le système proposé par Swapnil Belhe consiste à convertir les images originales en niveaux de gris lors du prétraitement. Deux caractéristiques différentes, les motifs binaires locaux (LBP) et l'histogramme des gradients orientés (HoG) sont calculées à partir des images.

Ces deux caractéristiques sont finalement combinées pour obtenir une caractéristique de dimension 292 (36 de HoG et 256 de LBP) utilisée pour l'apprentissage et le test.

HUST, Chine [44]: le système présenté par Baoguang Shi et al est principalement basé sur un réseau de neurones profonds. Le système prend des images d'entrée de format d'image arbitraire et peut prédire avec précision les types de scripts à partir d'images de texte.

CVC-1, Espagne [44]: Louis Gomez a soumis deux systèmes d'identification de script vidéo. Dans le premier système, il a procédé à l'apprentissage à l'aide d'un réseau de neurones convolutifs à couche unique pour l'identification de scripts et utilise une technique de classification basée sur Naive Bayes Nearest Neighbor (NBNN).

CVC-2, Espagne [44]: Il est presque identique au CVC-1 mais des distances de Mahalanobis en utilisant l'algorithme d'apprentissage de métrique de la marge étendue (Large Margin Metric Learning algorithm) ont été utilisées à la place de la distance euclidienne dans le classifieur NN (Nearest Neighbor)

Google, Inc [44]: Yuanpeng Li de Google a utilisé une image normalisée comme entrée à une hauteur fixe et binarisée, puis a été transmise à un réseau convolutif profond pour la prédiction de classe. Dans le cas où une image est suffisamment large, une fenêtre glissante est utilisée et la classe ayant la plus grande confiance est choisie durant l'étape d'apprentissage.

Il utilise une descente de gradient stochastique (Stochastic Gradient Descent) et une régularisation de la L2 pendant l'entraînement, puis augmente la base d'apprentissage en introduisant un réplicateur de l'images à différentes résolutions, largeurs et degrés de densité.

Le tableau suivant résume les résultats obtenus dans la tâche 1 par les systèmes proposés :

		Accuracy (%)				
Script Triplets		**C-DAC**	**HUST**	**CVC-1**	**CVC-2**	**Google**
Com1 Samples:970	Arabic	98.68	100	99.34	99.67	100
	English	92.67	99.41	93.55	91.50	100
	Hindi	95.4	100	99.39	99.69	100
	Overall	95.46	99.79	97.32	96.8	100
Com2 Samples:977	Bengali	98.71	97.1	96.13	93.54	99.68
	English	89.44	99.12	90.62	90.62	99.70
	Hindi	86.50	98.77	97.55	98.77	99.08
	Overall	91.40	98.36	94.68	94.27	99.49
Com3 Samples:994	Gujrathi	92.05	99.39	98.47	98.78	99.69
	English	78.00	97.95	92.96	90.62	98.53
	Hindi	95.40	100	97.85	98.77	100
	Overall	88.33	99.09	96.38	95.98	99.4
Com4 Samples:981	Kannada	88.54	99.36	99.04	97.77	99.68
	English	89.74	100	90.91	90.91	99.41
	Hindi	96.01	100	96.93	99.39	99.69
	Overall	91.44	99.80	95.51	95.92	99.59
Com5 Samples:993	Oriya	98.47	98.77	99.39	98.47	98.47
	English	92.38	99.71	92.08	91.20	99.12
	Hindi	96.93	100	99.39	99.69	100
	Overall	95.87	99.50	96.88	96.37	99.19
Com6 Samples:983	Punjabi	90.82	98.10	98.10	97.15	99.68
	English	93.25	99.41	92.67	91.49	99.12
	Hindi	70.55	98.47	92.02	97.54	99.69
	Overall	84.94	98.68	94.2	95.32	99.49
Com7 Samples:988	Tamil	98.44	100	100	100	99.69
	English	83.87	98.24	90.91	90.91	99.41
	Hindi	96.32	100	97.23	99.39	100
	Overall	92.71	99.39	95.95	96.66	99.70
Com8 Samples:990	Telugu	98.76	99.07	99.69	98.14	99.38
	English	87.98	95.01	90.62	90.32	98.83
	Hindi	95.09	100	99.38	99.69	99.39
	Overall	93.84	97.98	96.46	95.96	99.19

Tableau 2 : Les résultats obtenus dans la compétition ICDAR 2015 (Tâche 1)

Conclusion

Dans ce chapitre nous avons cité les travaux récents et les différentes méthodes proposées dans le domaine d'identification de scripts vidéo ainsi que les déférentes bases de données et leur taux de reconnaissances.

Nous avons présenté également les importantes compétitions organisées en détaillant les tâches préposées ainsi que les systèmes concernés.

Chapitre 4
Résultats expérimentaux

Chapitre 4: Résultats expérimentaux

1. Introduction

Dans ce chapitre, nous allons présenter l'architecture du système que nous proposons. Ce système est basé sur le CNN et les outils de développement tels que **Python**,**Tensorflow** et **keras**. Nous discuterons de l'environnement de travail à savoir **Google Colaboratory** qui nous offre la puissance de calcul nécessaire pour l'apprentissage en profondeur.

Ensuite, nous procéderons à la description de la base de données utilisée et de l'ensemble des expérimentations effectuées pour l'atteinte de l'objectif qui constitue l'objet de notre mémoire.

2. Présentation des outils de développement

Cette section est consacrée à la description des outils de développement employés dans la conception et l'implémentation de système proposé.

2.1. Python

Python [47] est un langage de programmation interprète de haut niveau et orienté objet. La syntaxe simple et facile à apprendre de Python met l'accent sur la lisibilité et réduit donc le coût de la maintenance du programme. Python prend en charge les modules et les packages, raison pour laquelle python est utilisé par une vaste communauté de développeurs et de programmeurs.

2.2. Tensorflow

TensorFlow [48][49] est une bibliothèque de logiciels open source qui a été développée à l'origine par l'équipe Google Brain. TensorFlow a atteint la version 1.0 en février 2017 et son développement a progressé rapidement.
TensorFlow est une multi-plateforme qui fonctionne sur presque tous les processeurs standards et les processeurs graphiques y compris les plates-formes mobiles et intégrées.

TensorFlow regroupe une multitude de modèles et d'algorithmes d'apprentissage automatique et d'apprentissage en profondeur afin de pouvoir exploiter des réseaux neuronaux profonds pour la reconnaissance d'images, des réseaux neurones récurrents, le traitement du langage naturel, etc.

2.3.Keras

Keras [51] est une application de réseaux de neurones de haut niveau, écrite en Python et capable de s'exécuter sur TensorFlow. Keras a été développé pour permettre une expérimentation rapide. Il est utilisé par les chercheurs pour pouvoir faire des recherches de qualité et passer de l'idée au résultat le plus rapidement possible.

Les keras sont utilisés pour construire un prototypage rapide et prendre en charge les réseaux convolutifs et les réseaux récurrents et s'exécute sur CPU et GPU.

2.4. Google Colaboratory

Google Colaboratory [50] est une application en nuage basée sur l'environnement Jupyter, qui ne nécessite aucune installation. L'application Google Colaboratory permet aux utilisateurs d'écrire, d'exécuter, de sauvegarder et de partager leurs codes avec un accès à de puissantes ressources de calcul gratuitement et à partir du navigateur et prend en charge python et bien d'autres langues.

2.5. Matlab

MATLAB [52] est une plateforme de programmation spécialement conçue pour les ingénieurs et les scientifiques. MATLAB est un langage matriciel permettant l'expression la plus naturelle des mathématiques informatiques.

MATLAB peut être utilisé pour analyser des données, développer des algorithmes, créer des modèles et des applications.

MATLAB peut traiter de l'apprentissage en profondeur et de l'apprentissage automatique, du traitement du signal et des communications, du traitement d'images et de vidéos, etc.

3. La base des données

La base de données utilisée dans notre travail est une base de données de scripts vidéo multilingues nommé CVSI-2015, qui contient des images de texte de scène de dix scripts différents: Arabe, Hindi, Bengali, Oriya, Gujrati, Punjabi, Kannada, Tamoul, Telegu et Anglais [44].

La base de données comprend 10688 échantillons de mots issus des dix scripts. Les statistiques de la base de données sont présentées dans le tableau 3**.** La base de données a été divisée de manière aléatoire en apprentissage (60%), en validation (10%) et en test (30%).

Script	Total Word	Train Set	Validation Set	Test Set
Arabic	1011	607	101	303
Bengali	1032	619	103	310
English (Roman)	1135	681	113	341
Gujrathi	1086	651	108	327
Hindi (Devnagari)	1088	653	109	326
Kannada	1047	628	105	314
Oriya	1087	652	109	326
Punjabi (Gurumukhi)	1055	633	106	316
Tamil	1070	642	107	321
Telegu	1077	646	108	323
Total	10688	6412	1069	3207

Tableau 3: Description de la base de données.

La figure 4-1 montre quelques échantillons de mots vidéo extraits de l'ensemble des données pour chaque script.

Figure 4-1: Quelques échantillons de mots vidéo [44]

4. Le système proposé

Dans cette section, nous présentons la démarche du système que nous proposons en décrivant l'architecture utilisée.

4.1. Prétraitement

La base de données CVSI-2015 contient différentes tailles d'image et arrière-plans complexes. Pour éliminer la variation, nous utilisons certaines techniques de prétraitement. Ces techniques sont citées dans les lignes suivantes :

-Grayscale

Nous avons utilisé la technique grayscale pour éliminer les couleurs de toutes les images (voir chapitre 1)

-Ajuster le contraste de l'image

Le réglage du contraste de l'image aide à distinguer clairement les couleurs les plus sombres des images des couleurs vives. La fonction imadjust sature les 1% inférieur et les 1% supérieur de toutes les valeurs de pixels. Cette opération augmente le contraste de l'image de sortie.

-Binarisation

Après ajustement des images, une technique de binarisation globale utilisée pour convertir les images en noir et blanc en utilisant un seuillage global de l'image détecté automatiquement. Ce seuil permet de remplacer tous les pixels de l'image d'entrée par une luminance supérieure à la valeur du seuil par la valeur 1 (blanc) et de remplacer tous les autres pixels par la valeur 0 (noir).

-Détecter la valeur de fond

Dans cette étape, nous avons proposé une méthode statistique pour détecter la couleur d'arrière-plan.

Cette méthode compte le nombre de pixels de bordure des images binarisées. Le nombre de pixel de même type est considéré comme l'arrière-plan.

-Normalisation de l'image

Cette étape consiste à redimensionner toutes les images dans la même taille. Pour cela, si la taille est inférieure à 64x64, nous faisons un remplissage autour de l'image en utilisant la valeur d'arrière-plan détecté pour augmenter la dimension 64x64. Si la taille est supérieure à 64x64, nous réduisons simplement l'image.

La figure suivante montre quelques échantillons de la base de données avant et après le prétraitement :

Figure 4-2: Quelques échantillons de la base de données avant et après le prétraitement

4.2. Architecture du CNN

Le CNN proposé est constitué de deux blocs de convolution, bloc dense et couche de sortie.

Chaque bloc de convolution contient deux couches de convolution, une couche de Maxpooling et une couche de perte (dropout), le bloc dense contient une couche aplatie, deux couches de perte et deux couches denses.

CNN a tout d'abord été alimenté avec un vecteur de taille 64x64x1, puis le vecteur est passé dans les couches du réseau et applique une activation ReLu à chaque fois. Après avoir passé toutes les couches, la couche finale est la couche de sortie avec l'activation softmax.

La figure 4-3 présente l'architecture du réseau CNN:

Layer (type)	Output Shape	Param #
input (InputLayer)	(None, 64, 64, 1)	0
input_dropout (Dropout)	(None, 64, 64, 1)	0
block1_conv1 (Conv2D)	(None, 60, 60, 10)	260
block1_conv2 (Conv2D)	(None, 56, 56, 20)	5020
block1_pool (MaxPooling2D)	(None, 28, 28, 20)	0
block1_dropout1 (Dropout)	(None, 28, 28, 20)	0
block2_conv1 (Conv2D)	(None, 24, 24, 20)	10000
block2_conv2 (Conv2D)	(None, 20, 20, 20)	10000
block2_pool (MaxPooling2D)	(None, 10, 10, 20)	0
block2_dropout1 (Dropout)	(None, 10, 10, 20)	0
block3_flatten (Flatten)	(None, 2000)	0
block3_dense1 (Dense)	(None, 128)	256128
block3_dropout1 (Dropout)	(None, 128)	0
block3_dense2 (Dense)	(None, 128)	16512
block3_dropout2 (Dropout)	(None, 128)	0
output (Dense)	(None, 10)	1290

Figure 4-3: Modèle de CNN avec KERAS

4.3. Paramètres du CNN

Dans l'étape de l'expérimentation, certains paramètres sont fixés au départ dans tous nos tests, comme décrit ci-dessous :

-Taille du filtre de couche convolutive: 5x5

-Le pas de filtre (stride) : 1

-Taille de noyaux (pooling): 2x2

-Dense (Fully Connected layer): 128 neurone sauf la couche de sortie 10

-Dropout: L'un des meilleurs moyens de «régulariser» un modèle consiste à utiliser une technique de régularisation du décrochage (dropout), qui permet de réduire le sur-apprentissage (overfitting) du réseau en préformant la moyenne du modèle avec le réseau de neurones. Dans notre réseau, nous avons appliqué une couche dropout de Taux de **20%** sur les entré **et 50%** sur les autres couches.

-Epochs: 75

-Batch size : 14 (optimal pour la base de données utilisé)

4.4. Algorithme et Implémentation

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

1) importation de la bibliothèque KERAS et ses composants

2) chargement de la base de données au format "**.mat**"

3) création et compilation de modèle

4) démarrage de l'apprentissage et le test

L'algorithme a été écrit en python avec l'utilisation de la bibliothèque KERAS et exécuté sur la plateforme Google Colaboratory (voir figure 4-4)

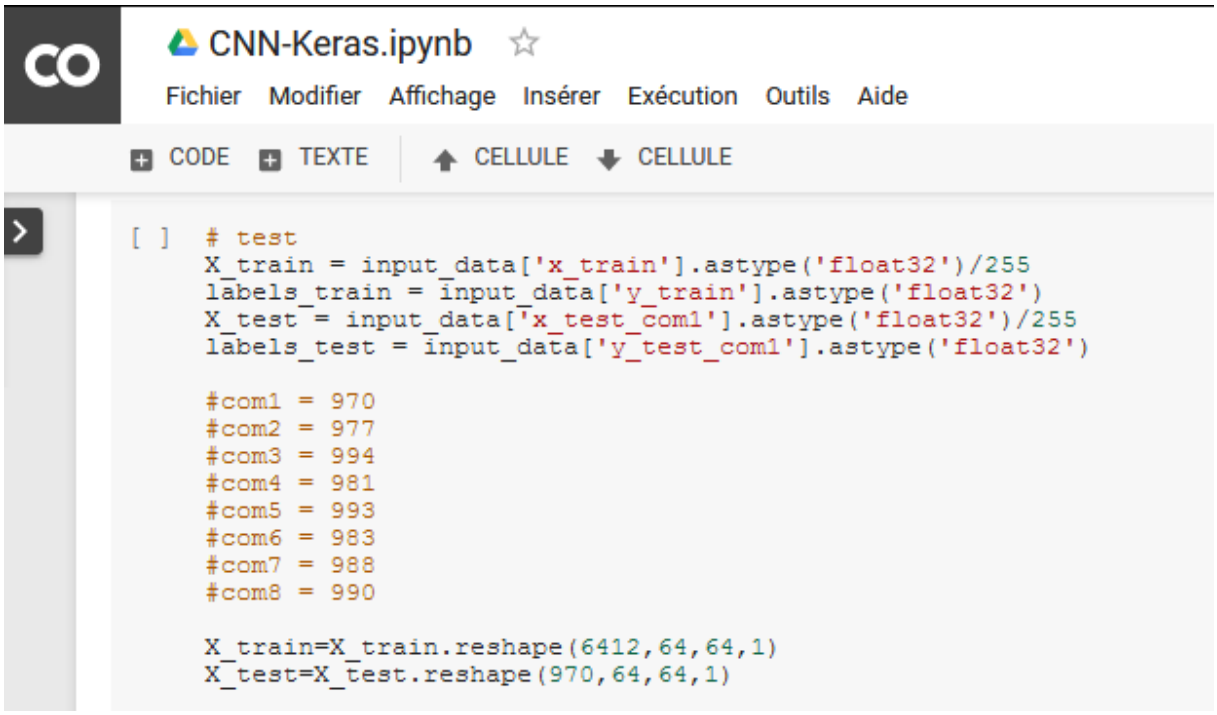

Figure 4-4: Partie du code dans Google Colaboratory

5. Résultats obtenus et discussion

Après la construction de CNN et la fixation de certains paramètres, nous effectuons d'abord un test sur une base de données d'images en niveaux de gris de dimension 28x28.

Le tableau ci-dessous montre les résultats obtenus.

Combinaison	Taille	Conv1	Conv2	Conv3	Conv4	Taux (%]	Nb Epoch
1	28x28	6	12	6	12	35,67	6
2						28,76	68
3						08,25	71
4						19,06	30
5						19,03	50
6						14,85	30
7						08,81	35
8						41,92	32
1		10	20	10	20	61,96	12
2						42,68	52
3						14,59	37
4						31,91	38
5						38,67	51
6						24,31	66
7						25,30	10
8						42,02	19

Table 4: Résultat du test 1

Discussion 1

Les résultats obtenus sont très faibles, même lorsque nous ajoutons des filtres de convolution, nous concluons que la taille des images 28x28 n'est peut-être pas optimisée, la fonction de redimensionnement atténue la résolution des images, la technique niveau de gris est également insuffisant dans la phase de prétraitement. C'est pourquoi nous évitons la fonction de redimensionnement en ajoutant des zones de remplissage autour des images. Nous appliquons également d'autres techniques de prétraitement (voir section 4-1).

Test 2

Ce test est réalisé avec la taille du images 64x64 avec la même variation de paramètres que lors du premier test.

Le tableau ci-dessous montre les résultats obtenus.

Combinaison	Taille	Conv1	Conv2	Conv3	Conv4	Taux (%)	Nb Epoch
1	64x64	6	12	6	12	80,72	70
2						71,95	75
3						78,57	56
4						75,43	56
5						82,78	40
6						80,98	55
7						80,97	29
8						77,37	72
1		10	20	10	20	87,11	66
2						71,55	68
3						78,37	67
4						74,62	75
5						83,38	74
6						82,71	45
7						85,02	69
8						78,69	70

Tableau 5: Résultat du test 2

Discussion 2

Comme nous nous y attendions, la taille de l'image affecte directement la précision du modèle. Cependant, l'ajout de couches Convolutives a un impact limité sur l'augmentation de la précision**.** Nous supposons que la fixation du paramètre de taille et le nombre optimal de couches convolutives avec modification de l'architecture pourrait nous conduire à une meilleure précision.

Le tableau ci-dessous montre une comparaison globale de notre système avec les systèmes soumis à la compétition CVSI-2015 (tâche 1)

	C-DAC	HUST	CVC-1	CVC-2	Google	**Notre Système**
Com1	95.46	99.79	97.32	96.8	100	**87,11**
Com2	91.40	98.36	94.68	94.27	99.49	**71,55**
Com3	88.33	99.09	96.38	95.98	99.4	**78,37**
Com4	91.44	99.80	95.51	95.92	99.59	**74,62**
Com5	95.87	99.50	96.88	96.37	99.19	**83,38**
Com6	84.94	98.68	94.2	95.32	99.49	**82,71**
Com7	92.71	99.39	95.95	96.66	99.70	**85,02**
Com8	93.84	97.98	96.46	95.96	99.19	**78,69**

Tableau 6: Comparaison des résultats

Conclusion

Nous avons présenté dans ce chapitre une approche de classification basée sur les réseaux de neurones convolutifs. Nous nous sommes concentrés sur la phase de prétraitement car l'architecture et ses paramètres sont fixés pour éviter les énormes variations.

Différents résultats sont obtenus en termes de précision. La comparaison des résultats obtenus a montré qu'une bonne technique de prétraitement peut conduire à de meilleurs résultats.

Même si les résultats obtenus par notre expérimentation ne sont pas très éloignés de ceux obtenus par les chercheurs professionnels, notamment ceux de CVC-1 et CVC-2, nous pensons qu'un peu plus de temps aurait pu nous permettre de nous rapprocher davantage de leurs résultats sans pour autant les égaler.

Conclusion Générale

Dans ce projet, nous avons discuté des notions fondamentales des réseaux de neurones en général, de l'apprentissage en profondeur et des réseaux de neurones convolutifs en particulier. Nous avons défini et utiliser les principes de la méthode CNN, de leurs couches et de leurs paramètres obligatoires ou facultatifs pour obtenir le meilleur modèle possible en termes de précision.

CNN a divers paramètres à manipuler dans une architecture unique, ce qui crée de nombreuses probabilités et complique le choix, c'est pourquoi nous avons fixé la plupart des paramètres.

Nous avons rencontré des problèmes dans l'étape d'implémentation, le temps d'exécution est trop coûteux, pour résoudre le problème, nous lançons notre code dans Google Colaboratory et choisissons l'exécution sur GPU.

Bibliographie

[01] Debayle Johan ; Thèse de doctorat ; Traitement d'image à voisinages adaptatifs généraux ; Université Jean Monnet de Saint –Etienne ; France ;Soutenue le 30 Novembre 2005 ; p 3

[02] M. Luc : "Reconnaissance de l'écriture manuscrite avec des réseaux récurrents", Université de Rouen, 2015.

[03] M.C. Padma, P.A. Vijaya, Script identification from trilingual documents using profile based features, Int. J. Comput. Sci.Appl. (IJCSA) 7 (4) (2010).

[04] NZIWOUE CHIADJEU Wilfried: "IMPLEMENTATION D'UN SYSTEME DE RECONNAISSANCE FACIALE PAR LA TECHNIQUE DES EIGENFACES SOUS JAVA ET ANDROID",2016, Université de Douala

[05] Pramod Kaler, —Study of Grayscale image in Image processing International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8169, Volume: 4 Issue: 11

[06] A. Belaid, Reconnaissance automatique de l'écriture et du document, Pour la science, 2001.

[07] Smith, E.H.B., Likforman-Sulem, L. and Darbon, J. (2010) 'Effect of pre-processing on binarization', DRR

[08] Noise Removal, *https://www.mathworks.com/help/images/noise-removal.html*

[09] S. Kaur, "Noise Types and Various Removal Techniques," International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), vol. 4, no. 2, 2015.

[10] P. Vithlani, and C.K.Kumbharana, "Structural and Statistical Feature Extraction Methods for Character and Digit Recognition," International Journal of Computer Applications, vol. 120, no. 24, 2015.

[11] Borse, S.B., Bhalekar, P.M., & Kharat, D.M. (2014). CHARACTER RECOGNITION WITH OPTIMAL ZONING USING GA.

[12] Herekar, R. and Dhotre, S. R., 2014. Handwritten Character Recognition Based on Zoning Using Euler Number for English Alphabets and Numerals, IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 16(4)

[13] Acharya T, Ray A (2005) Image processing: principles and applications. Wiley-Interscience. p154-155,p177

[14] Ryszard S. Chora´s , "Image Feature Extraction Techniques and Their Applications for CBIR and Biometrics Systems", 2007.

[15] Frédéric. Grandidier, un nouvel algorithme de sélection de caractéristiques: application à la lecture automatique de l'écriture manuscrite, thèse doctorat en génie P. H. D,université du Quebec, Montréal, Canada, 2003

[16] N. Benahmed, Optimisation de réseaux de neurones pour la reconnaissance de chiffres manuscrits isolés : sélection et pondération des primitives par algorithmes génétiques, Thèse de doctorat, Université de Québec, 2002

[17] Cleuziou, G.: Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information. Thèse de doctorat. LIFO, Université d'Orléans (2004)

[18] Romain Huet ; Thèse de doctorat ; Codage neuronal parcimonieux pour un système de vision ; Université de Bretagne Sud ; Institut de recherche en informatique et systèmes aléatoires (IRISA). Thèse soutenue le 19- 6- 2017

[19] TSOPZE Norbert,Treillis de Galois et r´eseaux de neurones : une approche constructive d'architecture des réseaux de neurones, Thèse de doctorat, l'Universit´e d'Artois et de l'Universit´e de Yaound´e I,2010,p14

[20] E. DAVALO et P. NAIM. Des réseaux de Neurones. Eyreolles, 1993.

[21] F. W. SCHMIDT. Neural Pattern Classiying Systems, Theory and experiments with trainable classifiers. Thesis, Techniche Universiteit Deif, 1994.

[22] Gelly, G. (2017). Réseaux de neurones récurrents pour le traitement automatique de la parole. (Speech processing using recurrent neural networks).

[23] Freund, Y.; Schapire, R. E. (1999). "Large margin classification using the perceptron algorithm". Machine Learning. 37 (3): 277–296.

[24] Pierre Buyssens, Fusion de différents modes de capture pour la reconnaissance du visage

appliquée aux e_ transactions DOCTORAT de l'UNIVERSIT´E de CAEN Le 4 Janvier 2011

[25] Bishop, Christopher M. (2006). Pattern Recognition and Machine Learning. Springer.

[26] F. ROSENBLATT Psychological Review Vol. 65, No. 6, 19S8 "THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN"

[27] Mohamed Bouaziz:Réseaux de neurones récurrents pour la classification de séquences dans des flux audiovisuels parallèles. (Recurrent neural networks for sequence classification in parallel TV streams). University of Avignon, France 2017

[28] Dinarelli, M., Tellier, I.: Etude des reseaux de neurones recurrents pour etiquetage de sequences. In: Actes de la 23eme conférence sur le Traitement Automatique des Langues Naturelles, Paris, France, Association pour le Traitement Automatique des Langues (2016)

[29] Geoffrey E. Hinton, Boltzmann Machines,2007,

[30] J. Murphy, An Overview of Convolutional Neural Network Architectures for Deep Learning, 2016, Microway, Inc,p7

[31] Hochreiter, S. and Schmidhuber, J. Long short-term memory.Neural Computation, 9(8):1735–1780, 1997.

[32] Ciresan, Dan; Ueli Meier; Jonathan Masci; Luca M. Gambardella; Jurgen Schmidhuber (2011). "Flexible, High Performance Convolutional Neural Networks for Image Classification". Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume Volume Two. 2: 1237–1242.

[33] Krizhevsky, Alex. "ImageNet Classification with Deep Convolutional Neural Networks"

[34] Ciresan, Dan; Meier, Ueli; Schmidhuber, Jürgen (June 2012). Multi-column deep neural networks for image classification. 2012 IEEE Conference on Computer Vision and Pattern Recognition. New York, NY: Institute of Electrical and Electronics Engineers (IEEE). pp. 3642–3649. arXiv:1202.2745. CiteSeerX 10.1.1.300.3283. doi:10.1109/CVPR.2012.6248110. ISBN 978-1-4673-1226-4. OCLC 812295155.

[35] "A Survey of FPGA-based Accelerators for Convolutional Neural Networks", NCAA, 2018

[36] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. (2017-05-24). "ImageNet classification with deep convolutional neural networks" . Communications of the ACM. 60 (6): 84–90. doi:10.1145/3065386. ISSN 0001-0782.

[37] J. J. Hopfield, « Neural networks and physical systems with emergent collective computational abilities », Proceedings of the National Academy of Sciences, vol. 79, no 8, 1er

avril 1982, p. 2554–2558 (ISSN 0027-8424 et 1091-6490, PMID 6953413, DOI 10.1073/pnas.79.8.2554

[38] Ankan, K.B., Aishik, K., Ayan, K.B., Abir, B., Partha, P. R., Umapada, P., : Script identification in natural scene image and video frames using an attention based Convolutional-LSTM network. Pattern Recognition 85.172–184. (2019)

[39] L. Gomez, A. Nicolaou, D. Karatzas,Improving patch-based scene text script identification with ensembles of conjoined networks Pattern Recognit, 67 (2017)

[40] Mei J., Dai L., Shi B., Bai X.Scene text script identification with convolutional recurrent neural networks,Proceedings of the Twenty Third International Conference on Pattern Recognition (ICPR), IEEE (2016), pp. 4053-4058

[41] L. Gomez, D. Karatzas,A fine-grained approach to scene text script identification,Proceedings of the Twelfth IAPR Workshop on Document Analysis Systems (DAS), IEEE (2016)

[42] A. Nicolaou, A.D. Bagdanov, L. Gómez, D. KaratzasVisual script and language identification,Proceedings of the Twelfth IAPR Workshop on Document Analysis Systems (DAS), IEEE (2016)

[43] The 14th IAPR International Conference on Document Analysis and Recognition,*http://u-pat.org/ICDAR2017/index.php*

[44] Sharma, Nabin & Mandal, Ranju & Sharma, Rabi & Pal, Umapada & Blumenstein, Michael. (2015). ICDAR2015 Competition on Video Script Identification (CVSI 2015). 1196-1200. 10.1109/ICDAR.2015.7333950.

[45] ICDAR2017 Competition on Video Script Identification, *http://www.ict.griffith.edu.au/cvsi2017/competition.html*

[46] ICDAR 2015 Competition on Video Script Identification (CVSI-2015) , *http://www.ict.griffith.edu.au/cvsi2015/*

[47] What is Python? Executive Summary, *https://www.python.org/doc/essays/blurb/*

[48] What is the TensorFlow machine intelligence platform?, *https://opensource.com/article/17/11/intro-tensorflow*

[49] What is TensorFlow? The machine learning library explained , *https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html*

[50] Welcome to Colaboratory, *https://colab.research.google.com/notebooks/welcome.ipynb*

[51] https://keras.io/

[52] https://www.mathworks.com/discovery/what-is-matlab.html

[53] Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever,Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. JMLR, 15:1929–1958, 2014.