

MINISTERE DE LA RECHERCHE SCIENTIFIQUE ET L'ENSEIGNEMENT SUPERIEUR

UNIVERSITÉ DE KASDI MERBAH OUARGLA

Faculté des nouvelles technologies de l'information et de la
télécommunication

Département d'Electronique et de la Télécommunication



Mémoire

MASTER PROFESSIONNEL

Domaine : Science et Technologie

Filière : Electronique

Spécialité : Instrumentation pétrolière

Présenté par : Gasmi youcef

Belmasmar youcef

**Reconnaissance de caractères arabes
manuscrits**

Devant le jury :

M ^{me} Aourar salima	MA(A)	Président	UKM Ouargla
M ^r Benacer imad	MA(A)	Examineur	UKM Ouargla
M ^r Benatmane khaled	MA(A)	Encadreur	UKM Ouargla

Année Universitaire : 2016 /2017

Remerciements

En premier lieu, nous remercions le Tout Puissant ALLAH, notre créateur qui nous a donné la force d'accomplir ce travail.

Nous tenons à remercier vivement, mon encadreur :

Mr: benatemane kaled pour ses encouragements, ses conseils précieux et sa disponibilité.

Nous présentons nos chaleureux remerciements aux enseignants pour leurs aides et orientations durant notre étude.

Et à tous ceux qui ont participé de près ou de loin à la réalisation de ce travail.





Dédicace

Quoi que de plus que de pouvoir partager les meilleurs moments de sa vie avec les êtres qu'on aime.

Arrivé au terme de mes études, j'ai le grand plaisir de dédier ce modeste travail :

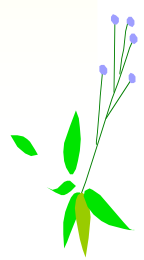
A ma très chère mère, qui me donne toujours l'espoir de vivre et qui n'a jamais cessé de prier pour moi.

A mon très cher père, pour ses encouragements, son soutien, surtout pour son amour et son sacrifice afin que rien n'entrave le déroulement de mes études.

A mes sœurs et mes frères.

A toute ma grande famille, belmesmar et gasmi

A tous mes amis et particulièrement mes collègues de la promotion pour leurs fraternités et esprit de groupe.



ملخص

العمل المقدم في هذه المذكرة هو عبارة عن كيفية التعرف على صورة معينة (الحروف العربية) من خلال برنامج في الماتلاب وذلك من خلال معالجة الصورة و استخراج بعض الخصائص (moment de zernik) و ادخالها في الشبكة (réseaux de neurones)
الكلمات الدلالية: معرفة الاشكال, غزوم zernik , خوارزمية معالجة الصور

Résumé

Le travail présenté dans ce mémoire est de savoir comment reconnaître un caractère arabe à travers un programme matlab basé sur le traitement d'image et l'extraction de certaines propriétés nommées les moments des Zernik, introduit dans un réseau de neurones.

Mots clés: Reconnaissance de formes, Moment de Zernik , Algorithme de traitement d'image

Abstract

The work presented in this memory is how to recognize an Arabic character through a matlab program based on image processing and extraction of certain properties which are introduced into neuronal network.

Key word: shape recognition, Zernik momenst, Image processing algorithm

Figure 1 : Le model générique d'un SRE	3
Figure 2 : Architecture de notre Système de reconnaissance de caractères	5
Chapitre I : Traitement d'image	
Figure I.1 Chiffre(2) et son squelette	18
Figure I.2 : Illustration de la procédure de normalisation d'un caractère	19
Figure I.3 mise on évidence d'un défaut de la première méthode de la normalisation (beaucoup de variable n'important aucune information)	20
Figure I.4 seconde méthode de normalisation	20
Chapitre II : Les Réseaux de Neurones	
Figure II -1 : Neurone biologique	23
Figure II -2 : Neurone formel	24
Figure II -3 : Schéma d'un réseau à trois couches	27
Figure II -4 : Réseau de Rétro propagation	29
Chapitre III : moments de Zernike et Découpage de l'image du caractère	
Figure III.1 les parties réelles et imaginaires de polynômes on ordre 3	33
Figure III.3 images reconstruites	35
Chapitre IV : Résultats et histogrammes	
Figure IV.1 : l'entraînement de réseau de neurone (ZE_globale)	46
Figure IV.2 : l'entraînement de réseau de neurone (ZE_haut)	48
Figure IV.3 : l'entraînement de réseau de neurone (ZE_bas)	48
Figure IV.4 : l'entraînement de réseau de neurone (ZE_gauche)	49
Figure IV.5 : l'entraînement de réseau de neurone (ZE_droit)	49
Figure IV.6 : l'entraînement de réseau de neurone (ZE_haut droit)	49
Figure IV.7 : l'entraînement de réseau de neurone (ZE_bas droit)	50
Figure IV.8 : l'entraînement de réseau de neurone (ZE_bas gauche)	50
Figure IV.9 : l'entraînement de réseau de neurone (ZE_haute gauche)	50

	Remerciements	
	Dédicace	
	Liste des figures	
	Introduction générale	1
	Chapitre I : Traitement d'image	
I.1.	Introduction.....	7
I.2.	Définitions général	8
I.3.	Acquisition d'image.....	10
I.3.2.	Définition de scanner	10
I.3.3.	Numérisation de l'image	10
I.4.	Prétraitement de l'image.....	11
I.4.1	Suppression de bruit.....	11
I.4.1.1.	Filtrage.....	11
I.4.1.2.	filtres linéaires	12
I.4.1.3.	filtres non linéaire.....	13
I.5.	Binarisation de l'image	13
I.6.	Segmentation	13
I.6.1.	Définition	13
I.6.2.	Approches de segmentation	14
I.6.2.1	Segmentations par extraction de contour.....	14
I.6.2.1.1.	Définition de contour	14
I.6.2.1.2.	Approches de détection	14
I.6.3.	La squelettisation	17
I.6.4.	Normalisation des caractères	18
	Chapitre II : Les Réseaux de Neurones	
II.1.	Introduction	21
II.2.	Les réseaux de neurones artificiels_	21

II.2.1	Historique.....	21
II.2.2	Le neurone biologique.....	23
II.2.3	Neurone formel	24
II.2.4	Fonction d'activation	25
II.3.	Les réseaux de neurones formels	26
II.3.1	Les réseaux non bouclés	26
II.3.2	Les réseaux bouclés	27
II.4.	L'apprentissage	27
II.4.1	L'apprentissage supervisé	27
II.4.2	L'apprentissage semi supervisé	28
II.4.3.	L'apprentissage non supervisé (auto-organisation)	28
II.5.	Réseau de Rétro propagation	28
II.5.1.	Algorithme de rétro propagation	29
II.5.2.	le pas d'apprentissage	31
Chapitre III : moments de Zernike et Découpage de l'image du caractère		
III.1.	Introduction	32
III.2.	Moments de Zernike	32
III.2.1.	Introduction	32
III.2.2.	Moments complexes de zernike	32
III.2.3.	Application a l'analyse de forme	34
III.2.4.	Application de la méthode des moments de Zernike sur une image	35
III.3.	Découpage de l'image du caractère	37
Chapitre IV : Résultats et discussions		
IV.1.	Introduction	43
IV.2.	Les bases de travail	43

IV.3.	Architecture du réseau de neurone utilisé	44
IV.4.	Résultats de la reconnaissance avec les moments Zernike	46
	Conclusion générale.....	52

Annexe

Bibliographie

Introduction générale

Introduction générale

Avec la reconnaissance de l'écriture se produit une autre révolution, intellectuelle plus que technologique. L'expression de "reconnaissance de l'écriture" désigne le fait que les ordinateurs sont devenus progressivement capables d'identifier des caractères écrits, manuscrits et imprimés, de les "reconnaître" et de les transformer immédiatement en telle ou telle représentation numérisée selon les codes utilisés. Les tout premiers systèmes de reconnaissance de caractère par des procédés de lecture magnétique ou optique ont été mis au point entre 1950 et 1960. C'étaient des systèmes de saisie indirecte, qui exigeaient que les caractères à identifier ont été soit imprimés ou dactylographiés auparavant à l'aide d'encre magnétisable soit convertis au préalable en codes à chiffres ou à barres. Ces systèmes de lecture optique sont devenus directs entre 1960 et 1970, mais nécessitaient toujours que l'on eût recours à des normes de transcriptions dactylographiques ou typographiques particulières, spécialement conçues à l'époque pour faciliter ces modes de saisie nouveaux. À partir de 1980, l'apparition de langages informatiques inédits de description de caractères permit d'assouplir l'utilisation de ces systèmes d'identification. Dès 1985, la plupart des polices de caractères alphabétiques étaient susceptibles d'être ainsi décrites et identifiées et, dès 1987, les polices de caractères idéographiques japonaises ou chinoises l'étaient à leur tour. La reconnaissance de l'écriture manuscrite a accompli de même des progrès spectaculaires, en particulier depuis 1980, jusqu'à permettre sa saisie dans l'instant même de sa transcription par le biais de tablettes de téléécriture qui se confondent parfois avec l'écran même de visualisation. Ces systèmes ont commencé à se répandre depuis 1988. Ils permettent désormais d'effectuer à volonté la synthèse de l'écriture manuscrite de tout un chacun. L'apparition également, depuis 1988, des premiers systèmes de traitement de texte (et donc d'écriture) à entrée vocale a enfin achevé de bouleverser les frontières admises entre l'écrit et l'oral. Ces systèmes permettent de produire en effet directement des signes écrits, des mots et des phrases, à partir de leur énoncé oral. En ce domaine, les progrès ont été lents pourtant. Les premiers systèmes de reconnaissance des mots qui aient été réellement opérants remontent au début des années 1970. En 1982, ces systèmes étaient déjà devenus capables de reconnaître entre une centaine et un millier de mots prononcés en différentes langues, de l'anglais au japonais en passant par le russe, l'espagnol, l'allemand ou l'italien. Ils réussissaient à identifier entre 3 000 et 5 000 mots en 1985, plus de 20 000 mots en 1988. Ces chiffres et ces dates

donnent une idée de la rapidité des progrès récents de ces systèmes. L'ambition proclamée de ces recherches est immense puisqu'il s'agit de rendre les ordinateurs capables de "comprendre" le langage humain, de le parler et de l'écrire du moins, et cela dès la décennie 1990-2000. En cette fin du XXe siècle, ce sont ainsi la place, le statut et le rôle que tous les systèmes d'écriture ont pu acquérir lentement au cours de ces cinq ou six derniers millénaires qui se trouvent sur le point d'être radicalement bouleversés. Cette conquête de l'écriture par l'informatique aura été extraordinairement rapide à l'échelle de l'histoire de l'humanité. De 1950 à 1985, les ordinateurs se seront en effet substitués, en moins de trente cinq ans, à pratiquement tous les procédés antérieurs connus de transcription, de transmission, de communication, de conservation ou de transformation d'un message écrit ou imprimé. De surcroît, en ramenant par des biais successifs tous les systèmes d'écriture concevables à un système unique de représentation à l'aide de deux chiffres, de deux caractères numériques, le "zéro" et le "un" seulement, la numérisation de l'écriture a brusquement révélé l'existence d'une espèce d'unicité des modes de transcription des langages humains qu'on ne faisait que soupçonner auparavant. En multipliant également les techniques de restitution, de régénération et de télécommunication ou de télédiffusion, l'informatique en aura aussi accentué la multiplicité. En ce sens, l'informatisation de ces systèmes d'écriture recèle autant de promesses d'éclatement que de perspectives de fusion vers un système de transcription unique, universel. Dès l'instant où ils ont commencé à être informatisés et numérisés, ces systèmes d'écriture ont cessé d'être des réalités stables, permanentes et figées comme on le concevait jadis. Ils sont devenus des entités instables, éphémères et mouvantes, sujettes à d'innombrables métamorphoses par le fait même de leur médiatisation et de leur numérisation. Les frontières avec ce qui était purement oral auparavant se déplacent également. C'est le statut même de ce qui était écrit, imprimé, lu, vu, entendu ou dit jusqu'à présent qui se déplace.

Grace au système de reconnaissance optique de caractères (OCR) un ordinateur peut, depuis plusieurs années, reconnaître un caractère comme image et de le transformer en code ASCII. Alors que cette reconnaissance est de plus en plus performante pour l'écriture imprimées, il n'en est pas de même pour l'écriture manuscrite. La reconnaissance de l'écriture manuscrite est actuellement un domaine très actif de recherche, où l'on distingue deux grands secteurs : la reconnaissance *en ligne* (on-line) et *hors ligne* (off-line). Les logiciels de reconnaissance étant encore limités.

Ce mémoire traite la reconnaissance des caractères manuscrits. En utilisant un réseau de neurones comme testeur et classificateur. Les programmes pour calculer les moments de Zernike et des programmes pour le traitement d'image (découpage, changement d'échelle, ...). La reconnaissance des caractères arabes manuscrits ou imprimée reste encore aujourd'hui au niveau de la recherche et de l'expérimentation, les travaux sont généralement axés sur la méthodologie du développement plutôt que sur la réalisation d'un produit fini vendable. Une version commercialisable reste encore au stade du rêve, les efforts doivent se multiplier pour la réaliser.

La figure 1 représente l'architecture générale d'un SRE (Système de reconnaissance d'Écriture)

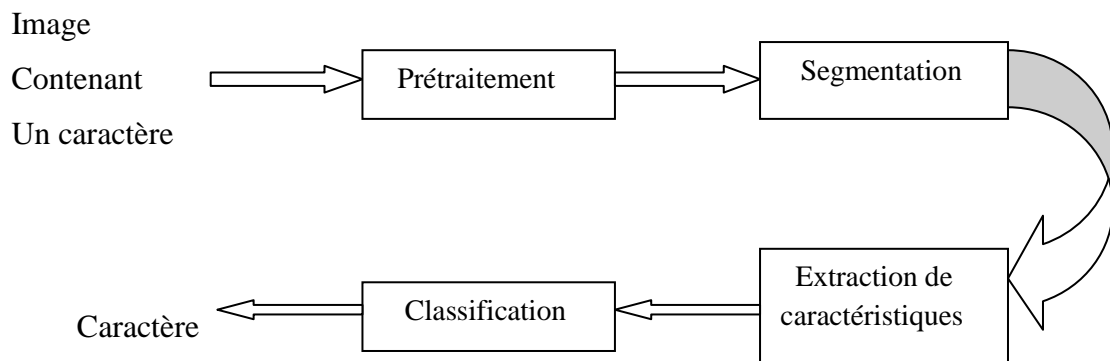


Figure 1 Le model générique d'un SRE

- **Le prétraitement** : dont le rôle est de préparer l'image à la phase de segmentation. Les opérations de prétraitement sont relatives au redressement de l'image, à la suppression de bruit, ...
- **La segmentation** : cette opération consiste à partager une image en ses constituants. Dans le cas des caractères arabe il peut s'agir de séparer le corps principal du caractère de ses corps secondaires (les points par exemple). Dans le cas des mots il s'agit de séparer ces mots en leurs caractères constitutifs. Dans le cas du texte, il s'agit à un premier niveau de partager ce texte en ses mots constitutifs, à un deuxième niveau de partager les mots en lettres et en fin, si nécessaire, partager le caractère en corps.

- **Extraction de caractéristiques :** consiste à extraire l'information distinctive à partir des matrices des caractères digitalisés. Lorsque l'extraction des caractéristiques est déterminée. Chaque caractère est représenté par l'ensemble des caractéristiques extraites.
- **La reconnaissance de caractères :** permet de comparer la caractéristique de caractère de reconnaissance avec la caractéristique des autres caractères qu'il ont fait l'apprentissage. L'architecture générale de notre système de reconnaissance des caractères arabe manuscrits est montrée dans la figure 2

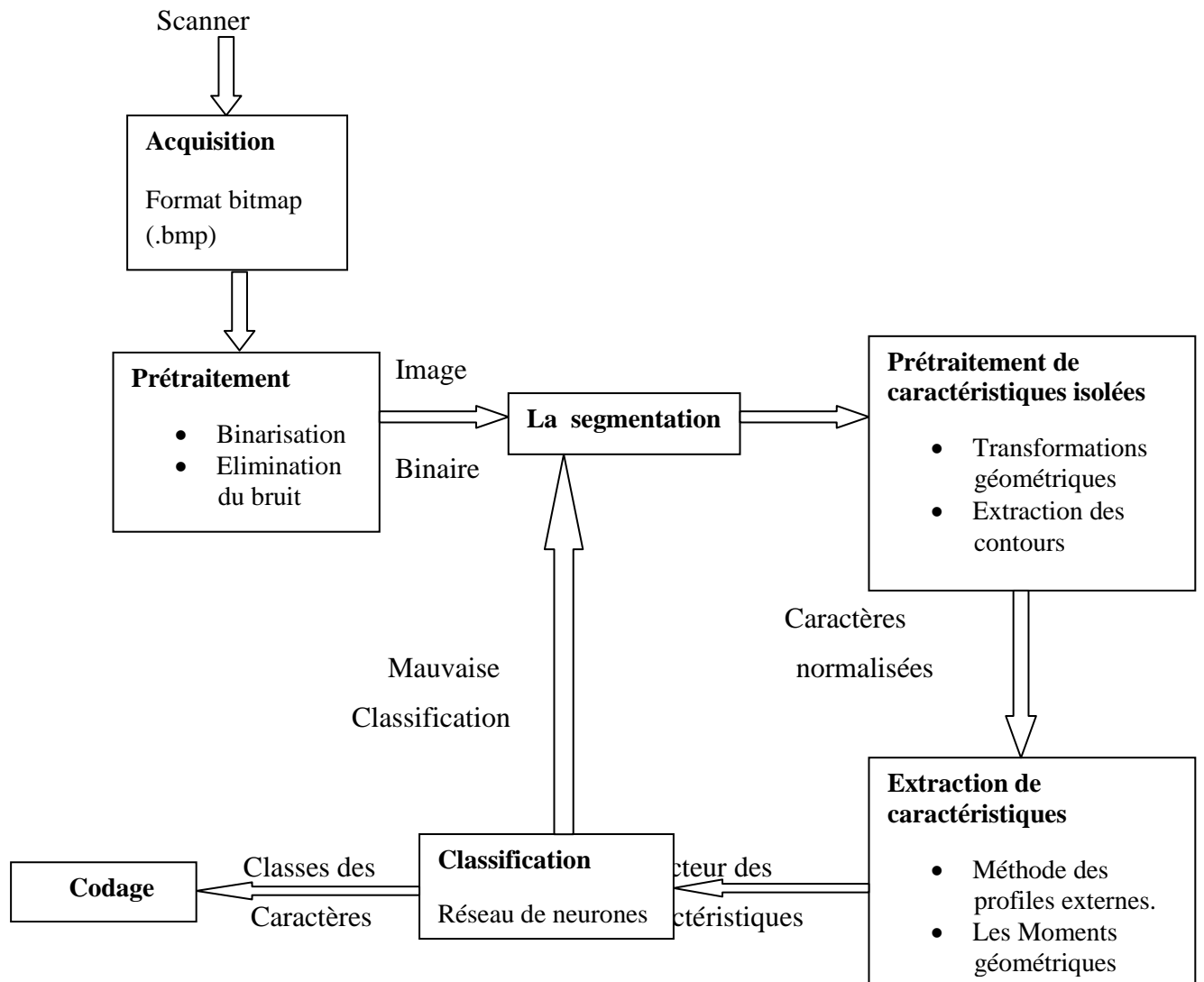


Figure 2 Architecture de notre Système de reconnaissance de caractères.

Notre objectif à travers ce mémoire est d'essayer de réaliser un système de reconnaissance des caractères arabe manuscrits.

Notre travail est concentré sur :

- Supporter le format d'image le plus utilisé (bitmap « .bmp »).
- Le choix, l'amélioration ou le développement des algorithmes selon les deux critères :

La rapidité d'exécution ou la performance des résultats (squelettisation, binarisation , normalisation de taille de l'image ,...).

- Deux méthodes
 - La méthode des moments de Zernike.
 - La méthode de découpage de l'image.
- de calculer le taux de reconnaissance et de comparer entre ces différentes méthodes.

Le reste de ce mémoire est organisé en 4 chapitres .Le chapitre 1 présente généralité sur le traitement d'image. Le chapitre 2 présente un résumé sur les réseaux de neurones,. Le chapitre 3 présente la méthode des moments de Zernike. Le chapitre 4 présente les résultats obtenus et une description sur les programmes. On achève notre mémoire par une conclusion générale.

Chapitre I :

Traitement d'image

I.1-introduction

Le xxi^e siècle est celui de l'image numérique, vecteur d'information pour la presse, les artistes, le monde médical mais aussi celui de l'industrie. Le traitement d'images est une science récente qui a pour but d'offrir aux spécialistes de différents domaines, comme au grand public, des outils de manipulation de ces données digitales issues du monde réel. L'histoire commence dans les années cinquante avec les rayons X. Puis vint l'étude des méthodes d'extraction automatique d'informations avec de nombreuses applications en reconnaissance de caractères et en contrôle qualité. Dans les années quatre-vingt, les efforts se poursuivront avec l'introduction des systèmes experts qui devaient remplacer l'homme dans ses différents diagnostics ! Enfin dans le courant des années quatre-vingt et quatre-vingt-dix, le traitement d'images passe du 2D au 3D grâce aux nouvelles possibilités technologiques. Les supports de stockage, les processeurs permettent désormais de traiter des informations multidimensionnelles de couleur et de mouvement, puis dans les années deux mille, l'intérêt se porte de plus en plus sur les modèles perceptifs afin d'optimiser les modèles usuels. Par ailleurs, plutôt que de développer des algorithmes utilisables dans toutes les situations, des méthodes dédiées sont développées, en intégrant de la connaissance a priori sur les contenus et surtout sur la finalité de l'exploitation des données. Le traitement d'images est donc une science en pleine évolution qu'il n'est pas possible de décrire pleinement dans ces quelques pages. Nous allons tenter de donner les bases de cette science qui a pour objectifs principaux

1. de saisir une scène réelle pour constituer un fichier informatique, c'est l'étape d'acquisition ;
2. de mettre en évidence les informations intéressantes : c'est l'étape de traitement dans laquelle on distinguera tout particulièrement les étapes de débruitage puis la segmentation ;
3. d'interpréter ces informations pour décider d'un diagnostic ou d'une action à engager : c'est l'étape d'analyse ;

I.2-Définitions général

Pixel

Le mot pixel provient d'une abréviation de l'expression Anglaise "Picture Elément". Le pixel représente ainsi le plus petit élément constitutif d'une image numérique. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image finalement obtenu. Etant donné que l'écran effectue un balayage de gauche à droite et de haut en bas, on désigne généralement par les coordonnées [0,0] le pixel situé en haut à gauche de l'image.

Bit

Un bit est la plus petite donnée qu'un ordinateur peut utiliser. De la même manière qu'une lampe, il ne peut être qu'allumé ou éteint. Si nous décidons d'utiliser un bit pour décrire notre image, nous pouvons utiliser cet état allumé ou éteint pour représenter le noir ou le blanc sans avoir d'état intermédiaire possible (pas de gris).

Image

Une image est constituée d'un ensemble de points appelés pixels. Il existe plusieurs types d'image : bitmap, gif, tiff, jpeg,...

Image Bitmap (.bmp) : Matrice 2D de points (pixel - Picture élément)

- Adapté à l'affichage sur écran.
- Adapté au traitement de l'image.

Image GIF (.gif) : Fichier fortement compressés (méthode LWZ)

- Conservation d'une qualité très correcte.
- Le nombre maximal de couleurs contenu dans une image au format GIF est de 256.

Image TIFF (.tif) : Tagged Image File Format, Fichier fortement compressés (méthode LWZ)

- Adapté aux impressions

Image JPEG (.jpg) : Joint Photographic Expert Group, Fichier fortement compressé, mauvaise conservation de la qualité de l'image (Hoffman).

Image en noir et blanc

L'image monochrome en noir et blanc se caractérise par des niveaux de gris. Le niveau de gris recherché va correspondre à la valeur de l'intensité lumineuse recherchée. Ici, chaque pixel est codé sur un octet (8 bits) ce qui donne une valeur comprise entre 0 et 255

pour ce pixel. On a donc 2^m niveaux de gris ou m est le nombre de bits du pixel. Si cette valeur du pixel est proche de 0 alors on aura une couleur noire. Si la valeur est proche de 2^{m-1} alors on aura du blanc.

Image en couleurs

Pour l'image en couleurs on va pouvoir représenter un pixel sur 3 octets. En effet en utilisant la synthèse des couleurs dite additive on peut recomposer toutes les couleurs en partant des 3 de base: le rouge, le vert et le bleu. Pour les représentations réelles en 24 bits on aura donc 8 bits, soit un octet par couleur par pixel. Cela nous donne une possibilité d'affichage de 16,7 millions de couleurs simultanément. Il est à noter qu'aucun moniteur d'ordinateur n'est en mesure actuellement d'afficher autant de points mais surtout que l'homme est strictement incapable de distinguer autant de couleurs. Il existe aussi la possibilité de faire des images en couleurs indexées, le but étant de déterminer le nombre précis de couleurs différentes utilisées dans une image et de créer une palette de couleurs appropriées (ensemble de couleurs choisies pour colorer une image).

Le poids d'une image

Pour connaître le poids (en octets) d'une image, il est nécessaire de compter le nombre de pixels que contient l'image, cela revient à calculer le nombre de cases du tableau, soit la hauteur de celui-ci que multiplie sa largeur. Le poids de l'image est alors égal à son nombre de pixels que multiplie le poids de chacun de ces éléments.

Voici le calcul pour une image 640x480 en True color :

➤ Nombre de pixels :

$$640 \times 480 = 307200$$

➤ Poids de chaque pixel

$$24 \text{ bits} / 8 = 3 \text{ octets}$$

➤ Le poids de l'image est ainsi égal à :

$$307200 \times 3 = 921600 \text{ octets}$$

$$921600 / 1024 = 900 \text{ Ko}$$

(Pour connaître la taille en Ko il suffit de diviser par 1024).

I.3 Acquisition d'image

I.3.1 Définition de scanner

Le scanner ou scanneur est certainement le périphérique d'acquisition d'images le plus répandu. La qualité toujours croissante de ces périphériques permet actuellement d'obtenir des Images numériques de bonne qualité pour un budget très modeste, On rencontre 3 types de scanners :

- Les scanners à plat sont les plus répandus: leur maniement est très aisé, les images obtenues sont de bonne qualité et ils permettent de numériser tous les documents, même épais (livre par exemple). Leur principal inconvénient est leur encombrement sur le poste de travail.
- Les scanners à défilement sont généralement intégrés dans des périphériques multifonctions, ce qui réduit l'encombrement total, mais ils ne permettent pas la numérisation de documents épais.
- Les scanners à main sont plus particulièrement destinés aux documents de petite taille. Leur transport est aisé mais leur maniement est plus délicat car le déplacement du dispositif d'acquisition est assuré par l'utilisateur.

La plupart des scanners usuels utilisent un pilote TWAIN qui devra être installé sur l'ordinateur.

I.3.2 Numérisation de l'image

La numérisation est un procédé omniprésent de nos jours, nous le rencontrons lorsque nous écoutons un disque, regardons un DVD, visualisons une image ou une page Web sur notre ordinateur. Avec l'évènement d'Internet et des nouvelles technologies, tout le monde parle d' « image numérique », de « son numérique » ou encore « d'ère du tout numérique » ; mais savons-nous réellement ce que signifie cet attribut si important ?

A travers l'exemple de la numérisation des images, nous allons passer en revue les différents aspects de la numérisation, aussi bien les bases théoriques que sont le système binaire ou les procédés de numérisation au sein de l'ordinateur, que leurs applications et notamment l'exemple du scanner, qui est devenu un outil de plus en plus utilisé même chez les particuliers.

I.4 Prétraitement de l'image

I.4.1 Suppression du bruit

I.4.1.1 Filtrage

On considère que l'image non dégradée $f(x,y)$ est affectée d'un bruit additif $n(x,y)$:

$$g(x,y) = f(x,y) + n(x,y)$$

Pour résumé, à partir de $g(x,y)$ on désire supprimer $n(x,y)$ pour obtenir $f(x,y)$.

La notion de filtre est empruntée à la physique et aux techniques de traitement du signal. Si un signal (électrique, radio, image, etc...) présente des composantes de fréquences très différentes, il peut être avantageux d'en éliminer certaines, on parle de filtrage. Par exemple un filtre passe-bas enlève toutes les composantes du spectre de valeurs supérieures à une valeur de fréquence donnée.

Filtrer une image c'est lui appliquer une transformation mathématique qui modifie la valeur de gris de tout ou partie des pixels. Si cette fonction est linéaire au sens algébrique du terme, on parle de filtrage linéaire. Si la fonction prend en compte les valeurs de gris au voisinage de chaque pixel transformé, on parle de fonction de convolution. Le terme le plus général est celui d'anamorphose, qui signifie "image grotesque donnée par un miroir courbe". Contrairement aux manipulations d'histogrammes, qui sont des opérations ponctuelles sur tous les pixels de l'image, les filtrages sont des produits de convolution qui mettent en jeu l'environnement (voisinage) de chaque pixel.

Un filtre est défini par:

- Une fenêtre carrée de dimension impaire (3 ou 5) qui est déplacée sur l'image
- Une matrice de coefficients de même dimension (3 ou 5)

$$\begin{matrix} a_{1.1} & a_{1.2} & a_{1.3} \\ a_{2.1} & a_{2.2} & a_{2.3} \\ a_{3.1} & a_{3.2} & a_{3.3} \end{matrix}$$

- Une transformation combinant les valeurs recouvertes par la fenêtre et les valeurs de la matrice. Le filtre est linéaire si l'opération est une combinaison linéaire. Si $N_e(x,y)$ est le niveau d'entrée du pixel de coordonnées x et y , le niveau de sortie de ce pixel sera donné par la combinaison linéaire :

$$\begin{aligned}
 Ns(x, y) = & a_{1,1} \cdot N_{\varepsilon}(x-1, y-1) + a_{1,2} \cdot N_{\varepsilon}(x-1, y) & + & a_{1,3} \cdot N_{\varepsilon}(x-1, y+1) \\
 & + a_{2,1} \cdot N_{\varepsilon}(x, y-1) & + & a_{2,2} \cdot N_{\varepsilon}(x, y) & + & a_{2,3} \cdot N_{\varepsilon}(x, y+1) \\
 & + a_{3,1} \cdot N_{\varepsilon}(x+1, y-1) + a_{3,2} \cdot N_{\varepsilon}(x+1, y) & + & a_{3,3} \cdot N_{\varepsilon}(x+1, y+1)
 \end{aligned}$$

L'application du filtre à l'image consiste à déplacer la fenêtre et à remplacer la valeur de chaque pixel par le résultat de l'opération portant sur ses voisins. Une nouvelle image est ainsi générée (un facteur de normalisation permet de conserver un intervalle de valeurs identique en entrée et en sortie).

I.4.1.2. filtres linéaires

Un filtre est linéaire s'il peut s'écrire de la manière suivante :

$$\text{Filtre}(c1.\text{image1}+c2.\text{image2}) = c1\text{Filtre}(\text{image1}) + c2\text{Filtre}(\text{image2})$$

Filtrer signifie convoluer une image $I(x,y)$ avec une fonction $f(x,y)$ appelée réponse Impulsionnelle (ou opérateur de convolution) du filtre. Dans le cas continu, l'image filtrée est donnée par :

$$If(x, y) = (f * I)(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x', y') I(x - x', y - y') dx' dy'$$

Dans le cas discret et pour prendre l'exemple simplifié d'une image carrée, les domaines de I et de f sont bornés. Le domaine de I est $[-N/2, +N/2]^2$ si N désigne la taille de l'image et le domaine de f est $[-K/2, +K/2]^2$. La convolution s'écrit alors :

$$\begin{aligned}
 If(x, y) &= (f * I)(x, y) \\
 &= \sum_{i'=-k/2}^{i'+k/2} \sum_{j'=-k/2}^{j'+k/2} f(i - i', j - j') I(i', j')
 \end{aligned}$$

On notera que le filtrage linéaire consiste simplement à remplacer chaque niveau de gris par une combinaison linéaire des niveaux de gris des points voisins.

Les filtres linéaires les plus connus sont les filtres passe-haut, passe-bas, Sobel, Prewitt et Laplacien .

I.4.1.3. filtres non linéaires

Comme pour la convolution, ces filtres travaillent avec le voisinage du pixel courant. On prend alors le minimal, le maximal ou le médian du pixel et de son voisinage. Les filtres les plus connus sont les filtres Moyenne, Médiane, Nagao.

I 5.1 Binarisation de l'image

a. Définition

La binarisation (le seuillage) est la technique de classification la plus simple où les pixels de l'image sont partagés par un seul seuil s en deux classes : ceux qui appartiennent au fond et ceux qui appartiennent à la scène (l'objet). L'image est alors séparée en deux classes de façon à ce que l'information comprise entre 0 et s est retenue et l'autre non, ou vice-versa.

Soit l'image $I (M \times N)$, supposons que $f(x, y)$ représente le niveau de gris du pixel aux coordonnées (x, y) , $0 \leq x \leq M$, $0 \leq y \leq N$ et s est le seuil choisi, les pixels de l'objet sont ceux ayant le niveau de gris inférieur à s et les autres ayant le niveau de gris supérieur à s sont des pixels du fond. Alors, l'image binarisée G est déterminée par les pixels (x, y) .

Selon Horaud il existe trois grandes techniques de sélection du seuil s : global, local et dynamique. Comme il y a des différentes façons de déterminer le seuil s , il peut être considéré comme une fonction sous forme de $s = t((x, y), p(x, y), f(x, y))$ où $p(x, y)$ représente des propriétés locales du point (x, y) . Si s ne dépend que de la valeur $f(x, y)$ du point, le seuil est global, s'il dépend en plus de $p(x, y)$, s est un seuil local. Et si s dépend à la fois de (x, y) , de $p(x, y)$ et de $f(x, y)$, on dit le seuil dynamique ou bien adaptatif.

I.6 Segmentation :

I.6.1 Définition :

Phase d'interprétation d'une image sous forme d'entités décrites en termes de régions et de contours. La segmentation porte sur l'élaboration de processus de détection, suivie d'une mise en correspondance. La détection inclut l'intégration d'un indice visuel couramment décrit en termes de similarité (région) ou de dissimilarité (frontière). La mise en correspondance concerne le regroupement des points appartenant à une même entité.

La segmentation est de loin la phase la plus délicate de l'analyse d'images car elle doit permettre d'interpréter la scène aussi bien (voir mieux) que le ferait un observateur. Elle conditionne la pertinence des mesures effectuées. Elle vise donc à définir les zones d'intérêt dans une image. Deux types d'approches sont habituellement utilisés: l'approche par régions et l'approche par frontières. Le but de ces approches est de créer des images binaires à partir d'images niveaux de gris. C'est-à-dire des images dans lesquelles il n'existe que deux classes de pixels: les pixels objets qui correspondent aux zones d'intérêts et les pixels fonds qui correspondent à tout ce qui n'est pas zone d'intérêt. Les pixels objets définissent des masques binaires qui seront mis à profit pour les mesures morphologiques et photométriques.

Il n'existe pas d'algorithme universel de segmentation, à chaque type d'images correspond une approche spécifique.

I.6.2 Approches de segmentation :

I.6.2.1 Segmentations par extraction de contour :

I.6.2.1.1 Définition de contour

Un point du contour est un pixel qui se situe sur la frontière séparant deux régions voisines ou adjacent, il résulte d'une variation brusque de l'intensité lumineuse en ce point.

Un contour est le lieu où l'on observe une discontinuité de luminosité.

I.6.2.1.2 Approches de détection :

Il y a deux approches

1. Approche gradient : détermination des extrema locaux dans la direction du gradient.
2. Approche Laplacien : détermination des passages par zéro du Laplacien.

Le gradient d'une image :

Le gradient d'une image est le vecteur $\nabla I(x, y)$ défini par :

$$\nabla I(x, y) = \left(\left(\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right)_t \right)$$

Il est donc caractérisé par un module m et une direction ϕ dans l'image :

$$m = \sqrt{\left(\frac{\partial I(x, y)}{\partial x}\right)^2 + \left(\frac{\partial I(x, y)}{\partial y}\right)^2}$$

$$\phi = \arctan\left(\frac{\frac{\partial I(x, y)}{\partial y}}{\frac{\partial I(x, y)}{\partial x}}\right)$$

- La direction du gradient maximise la dérivée directionnelle.
- La dérivée de $I(x, y)$ dans une direction donnée d s'écrit $\nabla I(x, y) \cdot d$
- Le gradient d'une image filtrée
 $\nabla I'(x, y) = \nabla(I(x, y) * h(x, y)) = \nabla I(x, y) * h(x, y) = I(x, y) * \nabla h(x, y)$.

Le Laplacien d'une image :

- Le Laplacien d'une image d'intensité $I(x, y)$ est défini par

$$\nabla^2 I(x, y) = \frac{\partial^2 I(x, y)}{\partial \cdot x^2} + \frac{\partial^2 I(x, y)}{\partial \cdot y^2}$$

Invariant aux rotations de l'image.

- Le Laplacien est souvent utilisé en amélioration d'images pour accentuer l'effet de contour :

$$I'(x, y) = I(x, y) - c \nabla^2 I(x, y)$$

Sensibilité au bruit accrue par rapport au gradient.

- Le Laplacien d'une image filtrée $\Delta I'(x, y) = \Delta I(x, y) * h(x, y) = I(x, y) * \Delta h(x, y)$

Les opérateurs de différentiations :

Opérateurs de Prewitt :

$$h_1 = \frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad h_2 = \frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Masques de calcul du gradient en x et y

➤ La convolution de l'image par les masques ci-dessus correspond aux calculs des dérivées de l'image pré-filtrée par un filtre h séparable :

$$h(x, y) = h(x) \cdot h(y)$$

Avec: $h = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ et $d = \nabla h = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$

$$h_1(x, y) = d(x) \cdot s(y)$$

En effet :

$$h_2(x, y) = s(x) \cdot d(y)$$

➤ Masques de Prewitt directionnels :

$$h_3 = \frac{1}{3} \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad h_2 = \frac{1}{3} \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

la direction du gradient est celle du masque donnant la réponse maximale.

$$\nabla^2 = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

Ou :

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Masques d'approximation du Laplacien.

Remarque : Le Laplacien est obtenu par l'application, après le lissage, du noyau 8-connexe.

➤ Le Laplacien nécessite une convolution, le gradient deux.

➤ Invariance par rotation.

Opérateurs de Sobel :

$$h_1 = \frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad h_2 = \frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- Très populaire.
- Correspond à la convolution de l'image par : $[1 \ 2 \ 1] * [-1 \ 0 \ -1]$.
- Masques directionnels également.

Opérateurs de Kirch :

$$h_1 = \frac{1}{15} \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad h_2 = \frac{1}{15} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$$

$$h_3 = \frac{1}{15} \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \quad h_4 = \frac{1}{15} \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

Opérateurs de Robinson

$$h_1 = \frac{1}{5} \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix} \quad h_2 = \frac{1}{5} \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$h_3 = \frac{1}{5} \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix} \quad h_4 = \frac{1}{5} \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

I.6.3 La squelettisation

La procédure de squelettisation s'effectue sur une image binaire, et a pour but de réduire l'épaisseur du tracé d'un caractère à un pixel seulement, tout en conservant la continuité de celui-ci. Le principe de cette procédure est d'effectuer une succession d'opérations d'érosion conditionnelle, jusqu'à ce que le but recherché soit atteint. Une opération d'érosion consiste à éliminer dans une image tous les pixels d'intensité non nulle qui sont adjacents aux pixels de l'arrière-plan, L'érosion conditionnelle signifie qu'un pixel d'intensité non nulle ne peut être éliminé que si, une fonction logique déterminée des valeurs des pixels qui l'entourent, est vérifiée. Cette fonction logique a pour rôle de vérifier la participation ou non du pixel central au maintien de la continuité du tracé.

La procédure de squelettisation consiste, sur base de ces principes, à balayer l'image, en passes successives, au moyen d'une fenêtre de dimension 3×3 . Le pixel du centre de cette fenêtre prend alors une valeur déterminée par une fonction logique de l'ensemble des 9 points de la fenêtre. Le balayage de l'image s'effectue de haut en bas et de gauche à droite, et le processus se répète autant de fois que nécessaire. L'arrêt se produit lorsque l'ensemble de la matrice qui contient le caractère a été balayée sans qu'aucun nouveau pixel n'ait pu être éliminé.

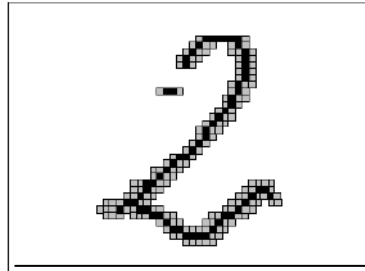


Figure I.1 Chiffre(2) et son squelette

I.6.4 Normalisation des caractères

En fonction de la taille initiale des caractères sur le document original d'une part, et de la résolution employée par le périphérique de digitalisation d'autre part, les dimensions de l'image qui représente un caractère dans l'ordinateur sont fortement variables. Afin d'éviter que le système de reconnaissance soit perturbé par cette dispersion de grandeurs, l'image d'un caractère peut être ramené à des dimensions normalisées.

Cette normalisation se fait en ré-échantillonnant l'image des caractères. Les points d'échantillonnage correspondent chacun à un pixel de l'image normalisée; leurs coordonnées dans l'image initiale du caractère sont régulièrement espacées, et calculées de manière à recouvrir celle-ci entièrement selon sa plus grande dimension (figure I.2). Le calcul de la valeur que prend un point d'échantillonnage diffère toutefois suivant que les dimensions initiales du caractère, soit sa largeur et sa hauteur, sont supérieures ou inférieures aux dimensions de normalisation.

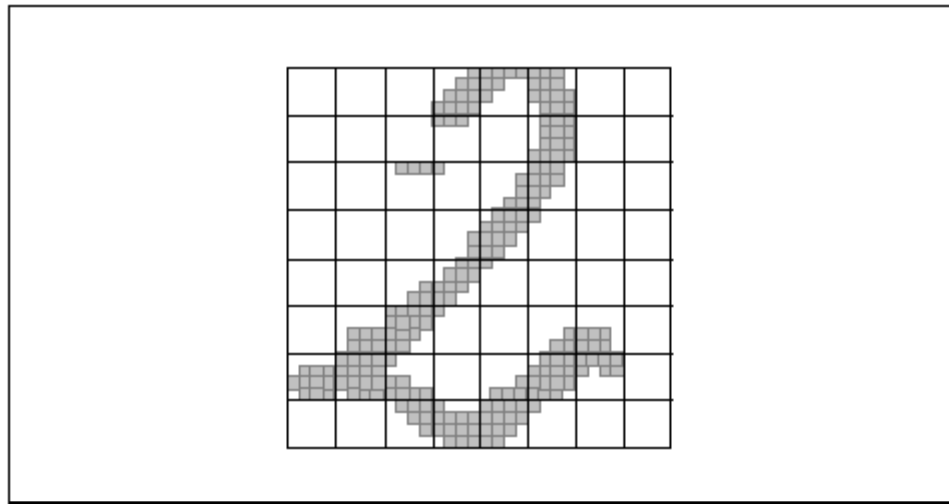


Figure I.2 : Illustration de la procédure de normalisation d'un caractère

Lorsqu'au moins une des deux dimensions de l'image qui représente un caractère, est supérieure à la dimension standard, celle pour laquelle l'écart est le plus élevé détermine l'amplitude de la contraction à effectuer. L'espace moyen entre deux points d'échantillonnage est calculé de manière à ce que, selon cette direction, la grille recouvre entièrement l'image du caractère. Les coordonnées calculées des points d'échantillonnage sont pratiquement toujours des valeurs réelles, et doivent donc être arrondies aux valeurs entières les plus proches figure I.2. Afin de conserver l'aspect initial du caractère, la même valeur d'intervalle d'échantillonnage est utilisée selon l'autre direction.

Lorsque les deux dimensions du caractère sont inférieures aux dimensions de normalisation, c'est celle pour laquelle l'écart est le plus faible qui détermine l'amplitude de la dilatation à effectuer. La largeur moyenne de l'intervalle d'échantillonnage est alors calculée de manière à recouvrir entièrement le caractère selon cette direction. Toujours de manière à conserver l'aspect initial du caractère, le même rapport de dilatation sera ensuite utilisé selon chacune des directions. Comme l'espace qui sépare deux points d'échantillonnage est ici, en moyenne, inférieur à un pixel, ces derniers prennent simplement la valeur du pixel dont les coordonnées entières sont les plus proches de ses coordonnées réelles calculées.

L'inconvénient de cette méthode de normalisation est que beaucoup de pixels de l'image normalisée n'apportent parfois aucune information (figure I.3).

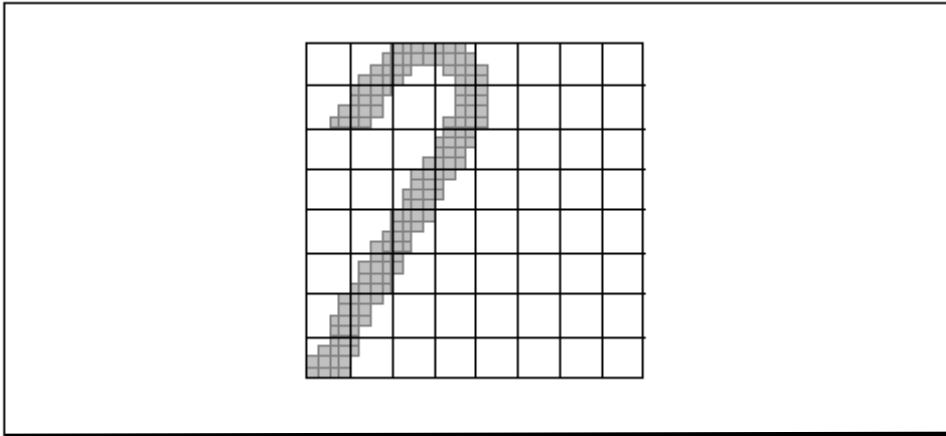


Figure I.3 mise on évidence d'un défaut de la première méthode de la normalisation (beaucoup de variable n'importent aucune information).

Une autre méthode de normalisation qui peut être plus efficace consiste à utiliser une valeur distincte de l'intervalle d'échantillonnage pour chaque direction, de manière à ce que le caractère soit entièrement recouvert par la grille d'échantillonnage selon chacune d'entre elles (figure I.4). Cette manière de procéder implique, toutefois, une perte d'information quant à l'aspect initial des caractères, ce qui pourrait compromettre la reconnaissance de certains d'entre eux. Pour éviter ce problème, le rapport d'aspect initial du caractère, c'est-à-dire le rapport entre sa largeur et sa hauteur, peut être calculé et inclus dans le vecteur de primitives.

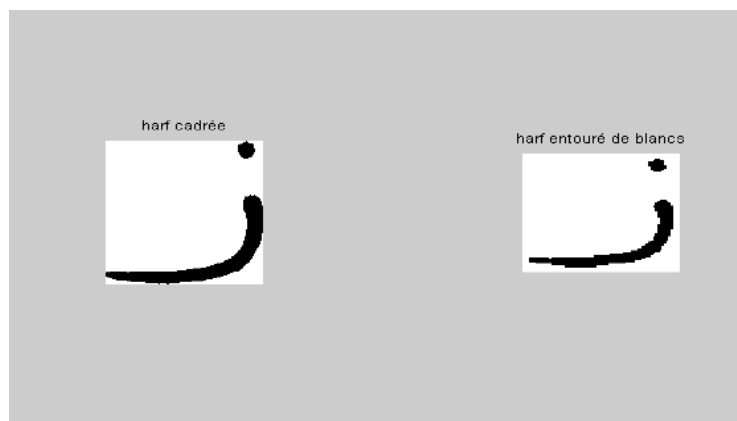


Figure I.4 seconde méthode de normalisation

NB.

Normalisation effectué en entrée pour le caractère entier. Cependant les parties du caractère, dans les traitements avec découpage ne doivent pas normalisée.

Chapitre II :

Les Réseaux de

Neurones

II.1. Introduction :

Les réseaux de neurones sont des structures (la plupart du temps simulées par des algorithmes exécutés sur des ordinateurs d'usage général, parfois sur des machines ou même des circuits spécialisés) qui prennent leur inspiration dans le fonctionnement élémentaire du système nerveux. Ils sont utilisés essentiellement à résoudre des problèmes de classification, de reconnaissance de formes, d'association, d'extraction de caractéristiques, d'identification, ...etc.

II.2. Les réseaux de neurones artificiels :

II.2.1. Historique :

Les neurologues Warren Sturgis McCulloch et Walter Pitts [1] menèrent les premiers travaux sur les réseaux de neurones à la suite de leur article fondateur : «What the frog's eye tells to the frog's brain ». Ils constituèrent un modèle simplifié de neurone biologique communément appelé . Ils montrèrent également théoriquement que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes [1].

Les travaux de McCulloch et Pitts n'ont pas donné d'indication sur une méthode pour adapter les coefficients synaptiques.

Cette question au cœur des réflexions sur l'apprentissage a connu un début de réponse grâce aux travaux du physiologiste américain [1] sur l'apprentissage en 1949 décrits dans son ouvrage "The Organization of Behaviour." Hebb a proposé une règle simple qui permet de modifier la valeur des coefficients synaptiques en fonction de l'activité des unités qu'ils relient. Cette règle aujourd'hui connue sous le nom de « règle de Hebb » est presque partout présente dans les modèles actuels, même les plus sophistiqués.

Franck Rosenblatt en 1957 introduit le modèle du Perceptron. C'est le premier système artificiel capable d'apprendre par expérience, y compris lorsque son instructeur commet quelques erreurs (ce en quoi il diffère nettement d'un système d'apprentissage logique formel).

En 1969, un coup grave fut porté à la communauté scientifique gravitant autour des réseaux de neurones : Marvin Lee Minsky et Seymour Papert publièrent un ouvrage mettant en exergue quelques limitations théoriques du Perceptron, notamment l'impossibilité de traiter

des problèmes non linéaires ou de connexité. Ils étendirent implicitement ces limitations à tous les modèles de réseaux de neurones artificiels. Paraissant alors une impasse, la recherche sur les réseaux de neurones perdit une grande partie de ses financements publics, et le secteur industriel s'en détourna aussi. Les fonds destinés à l'intelligence artificielle furent redirigés plutôt vers la logique formelle et la recherche piétina pendant dix ans.

En 1982, John Joseph Hopfield, physicien reconnu, donna un nouveau souffle au neuronal en publiant un article introduisant un nouveau modèle de réseau de neurones (complètement récurrent).

A la même date, les approches algorithmiques de l'intelligence artificielle furent l'objet de désillusion, leurs applications ne répondant pas aux attentes. Cette désillusion motiva une réorientation des recherches en intelligence artificielle vers les réseaux de neurones (bien que ces réseaux concernent la *perception* artificielle plus que l'intelligence artificielle à proprement parler). La recherche fut relancée et l'industrie reprit quelque intérêt au neuronal (en particulier pour des applications comme le guidage de missiles de croisière).

En 1984, c'est le système de rétro-propagation du gradient de l'erreur qui est le sujet le plus débattu dans le domaine.

Une révolution survient alors dans le domaine des réseaux de neurones artificiels : Une nouvelle génération de réseaux de neurones, capables de traiter avec succès des phénomènes non linéaires : le Perceptron multicouche ne possède pas les défauts mis en évidence par Minsky. Proposé pour la première fois par Werbos, le Perceptron Multi-Couche apparaît 1986 introduit par Rumelhart, et, simultanément, sous une appellation voisine, chez Yann le Cun. Ces systèmes reposent sur la rétro-propagation du gradient de l'erreur dans des systèmes à plusieurs couches, chacun de type Adaline de Bernard Widrow, proche du Perceptron de Rumelhart.

Les réseaux de neurones ont par la suite connu un essor considérable, On peut réguler un système d'apprentissage pour qu'il arbitre au mieux entre une modélisation pauvre (exemple : la moyenne) et une modélisation trop riche qui serait optimisée de façon illusoire sur un nombre d'exemples trop petits, et serait inopérant sur des exemples non encore appris, même proches des exemples appris.

Le sur-apprentissage est une difficulté à laquelle doivent faire face tous les systèmes d'apprentissage par l'exemple, que ceux-ci utilisent des méthodes d'optimisation directe (e.g. régression linéaire), itératives (e.g. descente de gradient), ou itératives semi-directes (gradient conjugué, estimation maximisation de Baum,...etc) et que ceux-ci soient appliqués aux modèles statistiques classiques, aux modèles de Markov ou aux réseaux de neurones formels.

II.2.2. Le neurone biologique :

Les neurones sont des cellules nerveuses au sein du cerveau (figure II-1). Elles sont composées de:

- **Corps (Soma) :** Où se déroule toutes les activités vitales de la cellule, par ailleurs c'est là où se trouve le noyau.
- **Axone et Dendrite:** Organes spécialisés dans la communication avec les autres cellules (neurones).
- **Synapse :** C'est une jonction entre les terminaisons axonales et les autres cellules.

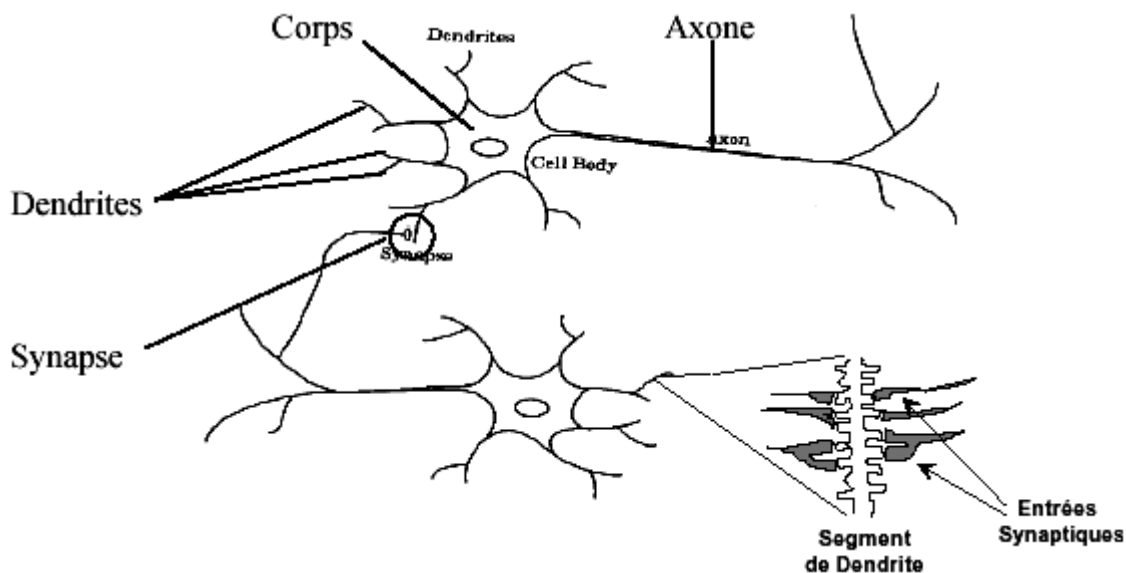


Figure II-1 : Neurone biologique [2].

L'impulsion nerveuse est une manifestation de la communication intercellulaire. C'est une décharge électrique prenant naissance à l'extrémité de l'axone (segment initial). Cette décharge se propage le long de l'axone pour arriver aux synapses où se déroulent des interactions fortement complexes qui transforment la décharge en un signal biochimique et cela par la libération de neurotransmetteurs (acétylcholine, adrénaline,...).

Ces neurotransmetteurs auront pour effet d'exciter les neurones qui les reçoivent ou au contraire d'inhiber l'influx nerveux.

II.2.3. Neurone formel :

Le modèle de neurone formel dû à Mac Culloch et Pitts [1] (figure II-2) est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amont. A chacune de ces entrées est associée un poids w , représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones aval. A chaque connexion est associée un poids.

Les paramètres plus importants de ce modèle sont les coefficients synaptiques (poids et seuil). Ce sont eux qui construisent le modèle de résolution en fonction des informations données au réseau. Il faut donc trouver un mécanisme qui permette de les calculer à partir des grandeurs que l'on peut acquérir du problème. C'est le principe fondamental de l'apprentissage. Dans un modèle de réseaux de neurones formels.

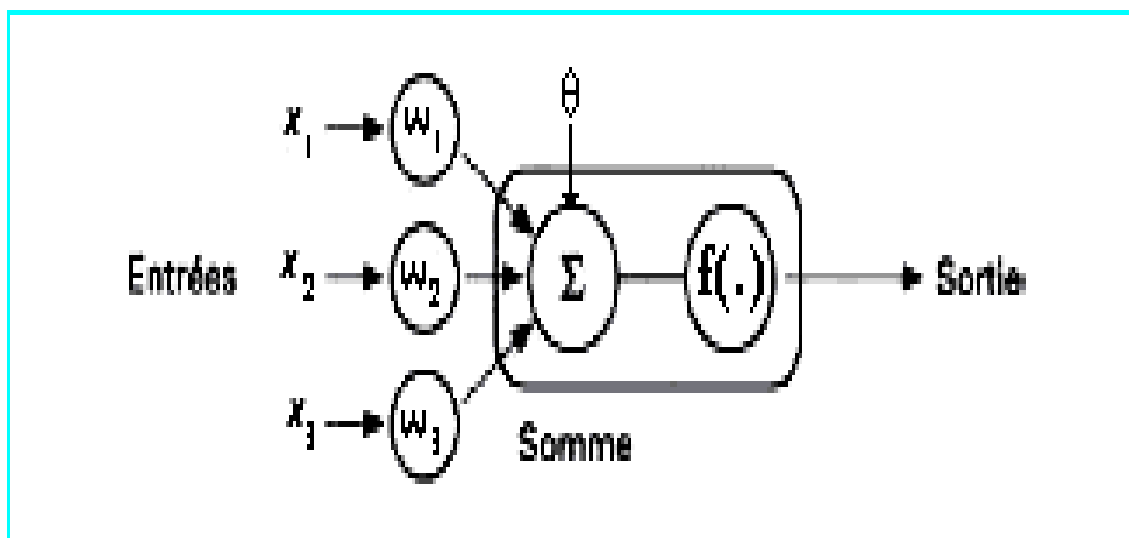


Figure II-2 : Neurone formel [1].

x_i : Les éléments du vecteur d'entrée.

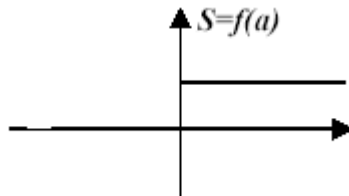
w_i : Les poids, θ est le seuil (biais).

$f(.)$: La fonction d'activation.

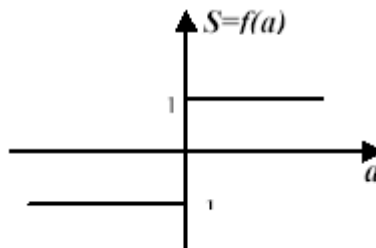
II.2.4. Fonction d'activation :

La fonction d'activation joue un rôle très important dans le comportement du neurone. Elle retourne une valeur représentative de l'activation du neurone, cette fonction a comme paramètre la somme pondérée des entrées ainsi que le seuil d'activation. La nature de cette fonction diffère selon le réseau. On en compte divers types, parmi elles :

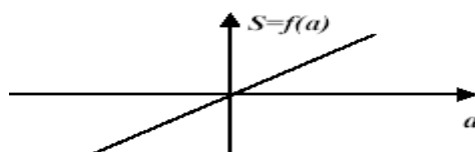
- **Fonction de Heaviside (seuil) :** $S = \begin{cases} 1 & \text{si } a \geq 0 \\ 0 & \text{si } a < 0 \end{cases}$



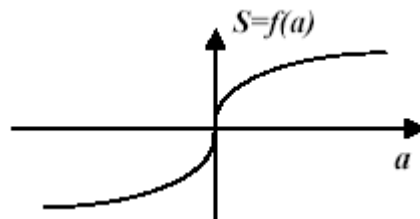
- **Fonction de signe :** $S = \begin{cases} 1 & \text{si } a \geq 0 \\ -1 & \text{si } a < 0 \end{cases}$



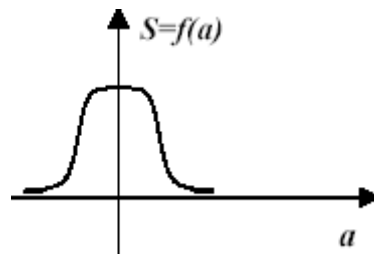
- **Fonction linéaire :** $S = \lambda a$



- **Fonction sigmoïde :** $S = \frac{1}{1 + \exp(-x)}$



- **Fonction gaussienne :** $S = \exp(-a^2/\sigma)$



II. 3. Les réseaux de neurones formels :

Nous pouvons dire qu'un réseau de neurones peut être défini comme un graphe orienté et pondéré. Les neurones étant les nœuds, qui seront connectés par des liens appelés liens synaptiques ou synapses.

Ces liens synaptiques sont pondérés par des poids judicieusement choisis. La propagation de l'activation à travers les liens synaptiques des neurones en aval va influencer les autres neurones en amont, cette activation sera pondérée par le lien qu'elle prendra ainsi on l'appellera poids synaptique [1] [2].

On distingue deux grandes architectures de réseaux de neurones :

II.3.1. Les réseaux non bouclés :

Un réseau de neurones non bouclé, si son graphe ne possède pas de cycle. Dans tel réseau, l'information circule de l'entrée vers la sortie sans aucun retour. Un réseau de neurones non bouclé est dit acyclique. Citons l'exemple des réseaux multicouches figure (II-3).

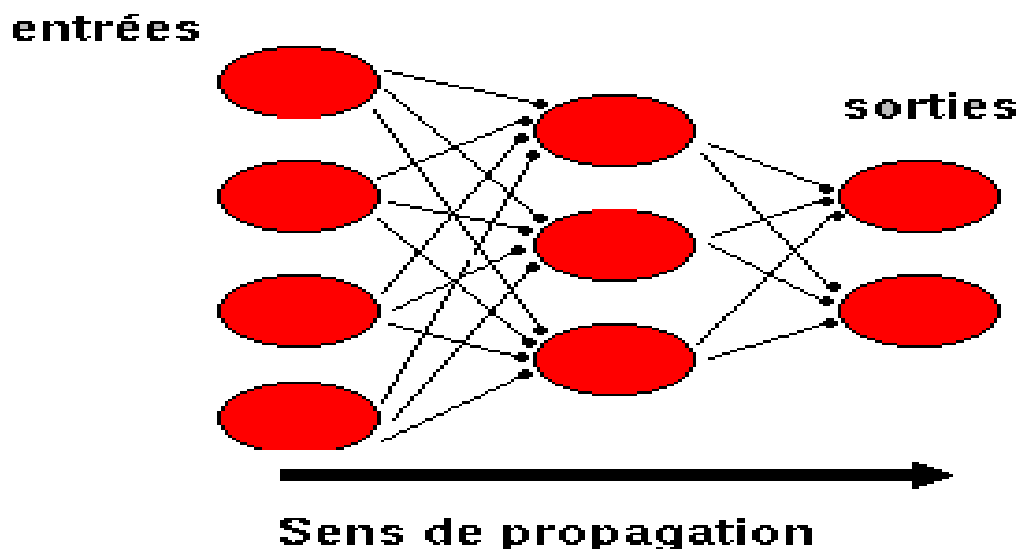


Figure II-3 : Schéma d'un réseau à trois couches.

II.3.2. Les réseaux bouclés :

Un réseau est bouclé, si son graphe possède au moins un cycle. Le réseau bouclé fait ramener une ou plusieurs valeurs à l'entrée. Un réseau de neurones bouclé est donc un système dynamique, régi par des équation différentielles. Comme l'immense majorité des application sont réalisées par des programmes d'ordinateurs, on se place dans le cadre des systèmes à temps discret, où les équations différentielles sont remplacées par des équations aux différences.

II.4. L'apprentissage :

L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. L'apprentissage neuronal fait appel à des exemples de comportement.

Dans les algorithmes actuels, les variables modifiées pendant l'apprentissage sont les poids des connexions. Il existe trois classes d'apprentissage : l'apprentissage supervisé, semi supervisé et non-supervisé [1], [2].

II.4.1. L'apprentissage supervisé:

Les apprentissages supervisés demandent que l'on donne au réseau neuronal des exemples résolus, c'est à dire des couples de vecteurs entrée/ sortie. Cet apprentissage exploite le plus souvent plusieurs idées simples. L'idée principale est la minimisation itérative d'un critère de l'erreur en sortie du réseau. On initialise les matrices de connexion au hasard, puis l'on fait évoluer ces matrices de manière à ce qu'elles autorisent

l'association souhaitée, c'est à dire jusqu'à ce qu'un critère de l'erreur (entre les sorties réellement obtenues et les sorties souhaitées) soit quasi nul.

II.4.2. L'apprentissage semi supervisé :

Dans ce type d'apprentissage, l'utilisateur possède seulement des indications imprécises sur le comportement final du réseau, mais en revanche, il est possible d'obtenir des indications qualitatives (correctes/ incorrectes) .

II.4.3. L'apprentissage non supervisé (auto-organisation) :

Dans ce type d'apprentissage (sans professeur) les poids synaptiques du réseau sont modifiés selon des critères internes comme la co-activation des neurones.

Le comportement de ce type d'apprentissage est comparable à des techniques d'analyse de données.

Enfin, certains réseaux associent les deux types d'apprentissage supervisé et non-supervisé.

II.5. Réseau de Rétro propagation :

Le réseau de rétro-propagation est un réseau à plusieurs couches constitué au minimum de trois couches (Figure 2-4), une couche d'entrée, une couche de sortie et une ou plusieurs couches cachées. Chaque couche contient un ou plusieurs neurones. Cela dépend du nombre de données qu'on désire faire apprendre au réseau, et du nombre de sortie qu'on désire avoir. Il n'existe pas une méthode bien précise pour déterminer le nombre de neurones dans la couche cachée, car il dépend de la complexité du problème.

Dans le réseau de la rétro-propagation, il n'existe pas d'interconnexions entre les neurones d'une même couche. Cependant chaque neurone de la couche courante fournit une entrée à chaque neurone dans la couche suivante.

Les neurones présents à la couche d'entrée auront chacun une seule entrée, et passeront tout simplement leurs valeurs d'entrée à leurs sorties sans passer par un limiteur ni une quelconque fonction. En ce qui concerne les valeurs initiales des poids entre couches, elles sont initialisées aléatoirement [2].

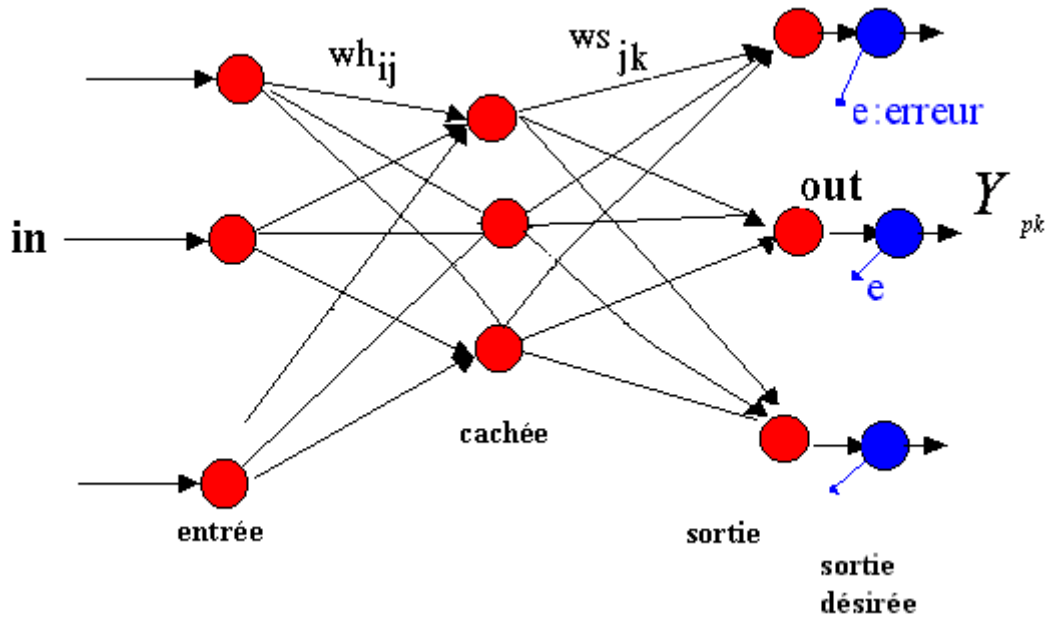


Figure II-4 : Réseau de Rétro propagation supervisé

II-5-1 Algorithme de rétro propagation :

Cet algorithme comporte les étapes suivantes :

- Étape 1 : Initialisation des poids synaptiques avec des petites valeurs aléatoires.
- Étape 2 : Présentation du vecteur d'entrée et propagation de l'activation des neurones.
- Étape 3 : Calcul de l'erreur, dans le cas d'un apprentissage supervisé cette erreur dépend de la différence entre l'activation des neurones et le vecteur de référence
- Étape 4 : Calcul du vecteur de correction à partir des valeurs des erreurs avec lequel on effectue la correction des poids synaptiques.

Les étapes 2-3-4 sont répétées tant que l'erreur est trop importante.

Dans ce qui suit nous allons présenter la notation utilisée dans cet algorithme. [2]:

1. Appliquer un vecteur d'entrée $in=[x_1, x_2, x_3, \dots, x_N]$ à la couche d'entrée.
2. Calculer l'entrée de la couche cachée.

$$Net_j^h = \sum_{i=1}^{N_i} W_{ij} * in_i + \theta_i^h \quad (1-1)$$

h : hidden (cachée).

j : Indice de neurone de la couche cachée.

θ : Le biais

Wh_{ij} : Le poids reliant le neurone i de la couche d'entrée au neurone j de la couche cachée.

N_i : Nombre de la couche d'entrée.

3. l'entrée de la couche cachée est

$$I_{pj} = f(Net_{pj}^h) \quad (1-2)$$

$f(.)$ Est la fonction d'activation.

4. l'entrée de la couche de sortie :

$$Net_k^o = \sum_{j=1}^{Nc} WS_{jk} * I_{pj} + \theta_k^o \quad (1-3)$$

WS_{jk} : Le poids reliant le neurone j de la couche cachée au neurone k de la couche de sortie,

Nc est le nombre de neurone de la couche cachée.

5. la sortie du neurone de la couche de sortie est :

$$Out_{pk} = f(Net_k^o) \quad (1-4)$$

6. calculer l'erreur pour les vecteurs de sortie

$$E_p = \frac{1}{2} \sum_{j=1}^{Nc} (Y_{pk} - Out_{pk})^2 \quad (1-5)$$

Où Y_{pk} est la sortie désirée, et Out_{pk} est la sortie calculée.

II.5.2. le pas d'apprentissage :

La rétro-propagation est une technique de descente du gradient, et doit donc en principe converger vers une solution. Cependant, cette constatation repose sur l'hypothèse que la décente sur la surface d'erreur soit continue. En pratique, nous avons vu que la rétropropagation procède par saut discret, dont la taille est réglée par un paramètre, le pas d'apprentissage.

Le choix d'une valeur de pas satisfaisante est difficile. Trop petit, il augmente inutilement le temps d'apprentissage en forçant le réseau à procéder par « petit pas » sur la surface d'erreur. Trop grand, il empêche le réseau de se stabiliser sur un choix de poids unique. Pour un choix de pas très grand, l'apprentissage n'obéit plus à une descente de gradient simple : L'erreur cesse de diminuer de façon monotone, et peut se comporter selon des dynamiques complexes, voir chaotiques. Cela n'est pas nécessairement un tort. En effet, l'apprentissage peut être plus efficace quand il n'est pas limité à une simple descente de gradient. Cependant, la convergence de l'apprentissage n'est pas garantie à cause de l'absence de fondement théorique.

Plusieurs méthodes heuristiques sont basées sur la constatation précédente, usant de ce fait d'un pas variable. Bien sur, ces méthodes dépendent d'intuitions sur la forme de la surface d'erreur et ne se justifient que par le succès qu'elles obtiennent en pratique. Outre ces méthodes, plusieurs auteurs ont proposé des techniques capables de modifier automatiquement la taille du pas d'apprentissage.

En particulier, il propose quatre règles aux quelles devrait adhérer ce type de techniques d'apprentissage. Il est possible de comprendre le fonctionnement de plusieurs variantes de la rétropropagation en fonction de ces règles, dont la première, avec laquelle on a travaillé est la suivante : une fonction d'apprentissage capable d'adapter automatiquement la taille de son pas d'apprentissage devrait le faire en fonction de la forme de la surface d'erreur

Locale à chaque correction. Cela est nécessaire, car la surface d'erreur peut changer considérablement de forme d'une région à l'autre ; un pas d'apprentissage donné peut donc ne pas convenir à toutes les régions de la surface.

Chapitre III :
Moment de Zernike et
Découpage de l'image
du caractère

III.1 Introduction

Pour réaliser et calculer le taux de la reconnaissance de notre mémoire on a utilisé deux méthodes:

1. Les moments de Zernike.
2. Découpage de l'image du caractère

III.2 Moments de Zernike [4]

III.2.1 Introduction

Comme pour les autres moments, les moments de Zernike sont également utilisés pour la reconnaissance et l'identification. La naissance de ces moments revient à Tague qui a proposé tout au début un ensemble de moments basés sur la base de polynômes orthogonaux appelées par la suite moments de Zernike. Il est bien connu qu'une fonction discrète d'image puisse être reconstruite par les moments de Zernike. En particulier les moments de Zernike se sont avérés intéressants quant à la robustesse par rapport au bruit et l'invariance par rapport à la rotation. Un ensemble relativement petit de moments de Zernike peut caractériser la forme globale d'un objet, Les moments d'ordre bas représentent la forme globale d'un objet et celles d'ordre plus supérieur le détail de cet objet. Plus on fait appel à des moments de Zernike d'ordre plus élevé, plus est la précision de reconstruction de l'image de l'objet en cours du traitement. La reconstruction dépend du nombre de polynômes de Zernike impliqués. Pour des images binaires la reconstruction peut être réalisée avec un ordre modeste. Cependant, pour les images à niveaux de gris l'ordre de reconstruction est en général plus important. En outre, les valeurs de la partie imaginaire des moments de Zernike pour des images de niveaux gris sont plus hautes que pour des images binaires.

III.2.2 Moments complexes de zernike [4] [6]

Le moment de Zernike d'ordre p est défini, sur une image à l'intérieur d'un cercle de rayon unitaire, par :

$$z_{pq} = \frac{(p+1)}{\pi} \int_0^{2\pi} \int_0^1 V_{pq}^*(r, \theta) \times im(r, \theta) \times r \times dr \times d\theta$$

Avec $p - |q|$ pair et $|q| \leq p$

$im(r, q)$ est l'image à analyser en coordonnées polaires.

V_{pq} est le conjugué du polynôme de Zernike d'ordre p et de répétition q .

Les polynômes de Zernike sont à valeurs complexes. Nous avons représenté jusqu'à l'ordre 3, sur la (Figure III.1), les parties réelles et imaginaires de ces polynômes.

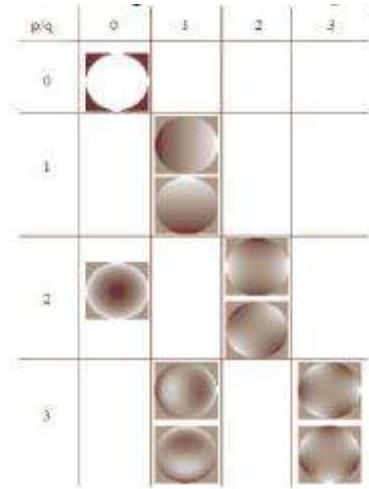


Figure III.1 Parties réelles et imaginaires de polynômes d'ordre 3

Les moments sont également complexes :

$$z_{pq} = R_e(z_{pq}) + j \times (z_{pq})$$

Ou

$$R_e(z_{pq}) = \sum \sum R_e(V_{pq}^*(x, y)) \times im(x, y)$$

$$im(z_{pq}) = \sum \sum im(V_{pq}^*(x, y)) \times im(x, y)$$

Les parties réelles et imaginaires peuvent donc être estimées séparément en suivant la même démarche que pour les moments géométriques. Cependant les fonctions $Re(V^*pq(x, y))$ et $Im(V^*pq(x, y))$ présentent des valeurs positives et négatives. Une normalisation de ces fonctions est donc nécessaire afin de pouvoir leur appliquer un algorithme de dithering et générer le masque binaire F_{pq} qui leur est associé. D'autre part, une fois obtenue la fonction f_{pq} il est nécessaire, comme nous le montrerons par la suite, de générer la fonction f_{pq} complémentaire de f_{pq} et permettant de tenir compte de la valeur minimale de $Re(V^*pq(x, y))$ et $Im(V^*pq(x, y))$ dans l'estimation de z_{pq} . La démarche est décrite sur la (Figure III.2) Nous avons représenté $Re(V^*11(x, y))$ en 3D. Une première étape de normalisation conduit à la fonction f_{pq} à valeurs comprises dans l'intervalle $[0,1]$. Le masque f_{pq} est alors généré par application du dithering et finalement son complémentaire est calculé.

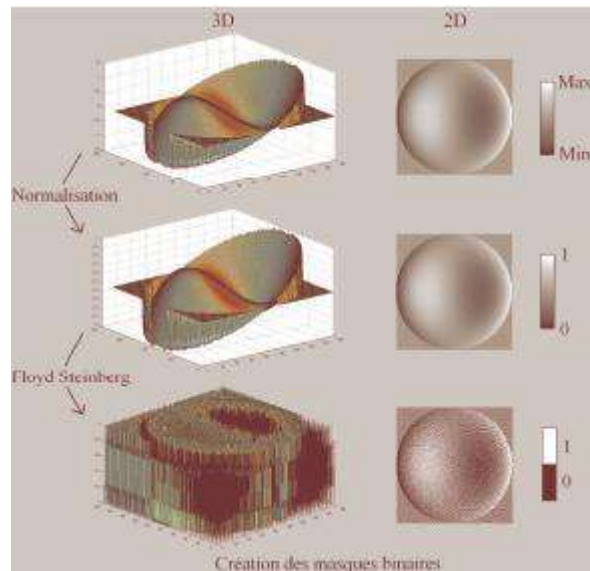


Figure III.2 Représentation 3D des parties réelles et imaginaires du polynôme

Finalement la valeur estimée de la partie réelle du moment de Zernike est définie par :

$$R_e(z_{pq}) = \text{Max} \times \sum \sum f_{pq}(x,y) \times \text{im}(x,y) + \text{Min} \sum \sum \overline{f_{pq}}(x,y) \times \text{im}(x,y)$$

Où Max et Min sont respectivement les valeurs maximales et minimales de $\text{Re}(V^{*pq}(x,y))$.

De même pour la partie imaginaire.

III.2.3 Application a l'analyse de forme :

Les moments de Zernike sont les moments les plus performants en matière de d'analyse de forme. La méthode que nous proposons dans ce mémoire, nous permet de déterminer une valeur approchée des parties réelles et imaginaires des moments de Zernike avec une erreur faible. En vue d'une application de reconnaissance de forme ou encore de compression d'image, l'influence de cette erreur doit être étudiée. Pour ce faire nous reprendrons la même démarche pour comparer les performances des descripteurs de Legendre et de Zernike : pour chacun on calcule une série de moments sur l'image analysée, et l'on reconstruit une image à partir de ces valeurs. Le descripteur le plus performant est celui qui conduit à une image reconstruite la plus proche de l'image analysée. Pour Zernike l'image reconstruite est obtenue à partir de la transformée inverse des moments de Zernike définie en coordonnées polaires par [4] :

$$\text{im}_r(r, \theta) \cong \sum \sum z_{pq} \times V_{pq}(r, \theta)$$

De même nous définissons l'image reconstruite à partir des valeurs estimées de moments de Zernike :

$$\text{IM}_r(r, \theta) \cong \sum \sum z_{pq} \times V_{pq}(r, \theta)$$

Sur la (Figure III.3), nous avons représenté deux exemples d'images reconstruites à partir des valeurs estimés et exactes de moments de Zernike. L'image a été reconstruite en utilisant les

valeurs des moments de Zernike allant successivement jusqu'à l'ordre 10, 15, 20, 25 et 30. La résolution est de 100x100 pixels.

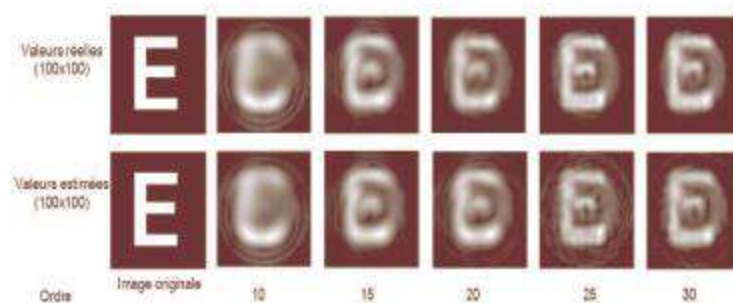


Figure III.3 Images reconstruites

Il apparaît que la reconstruction à partir des valeurs approchées est très proche de celle obtenue à partir des valeurs exactes.

Ceci confirme la pertinence des valeurs de moments obtenues par notre approche en matière de description d'image : même si les valeurs de moments déterminées à partir de masques binaires ne sont que des valeurs approchées de moments de Zernike, elles restent tout de même représentatives de la forme analysée.

III.2.4 Application de la méthode des moments de Zernike sur une image:

Voici un exemple des trente (30) moments de Zernike extrait de trois caractères arabes.

➤ **Pour un ordre égal 9 (30 moment)**



Sad cadrée (500×500)

1.222367e-01 1.154270e-02 1.665059e-01 2.796141e-02 3.359747e-02 1.806345e-02
 3.033127e-01 2.985401e-02 6.014355e-02 7.860131e-02 6.779703e-02 3.258886e-02
 2.830913e-01 3.101336e-02 2.176828e-02 6.077350e-02 4.799106e-02 5.773398e-02
 3.174221e-02 1.653974e-02 3.498997e-01 1.086640e-02 2.672630e-02 3.149362e-02

5.301007e-02 3.642991e-02 6.494741e-02 1.146036e-02 2.105548e-02 2.399382e-02



Ya cadrée (500×500)

5.976116e-02 2.389210e-02 4.289932e-03 1.903731e-02 1.527718e-02 2.385771e-02

5.896475e-02 1.993538e-02 4.558536e-02 1.311280e-02 2.693087e-02 2.137988e-02

9.244671e-02 2.594818e-02 4.955838e-02 1.667898e-02 1.820021e-02 1.221235e-02

4.170826e-02 3.050490e-02 1.072172e-01 1.232505e-02 2.527796e-02 3.640702e-02

1.943692e-02 4.435375e-02 3.597200e-02 2.559909e-02 2.598230e-02 2.038203e-02



Kha cadrée (500×500)

6.291854e-02 2.454080e-02 4.367285e-02 1.728114e-02 2.829308e-02 2.043296e-02

1.283819e-01 1.365691e-02 2.519519e-02 1.457083e-02 1.204311e-02 1.747815e-02

1.240730e-01 4.071374e-02 3.446912e-02 4.281106e-02 1.914685e-02 1.412067e-02

2.543022e-02 4.791616e-02 2.205901e-01 2.846158e-02 3.784106e-02 4.481104e-02

4.448166e-02 4.162600e-02 2.320351e-02 1.158395e-02 1.441239e-02 1.961061e-02

III.3. Découpage de l'image du caractère

Comme on l'a signalé auparavant une nouvelle technique de reconnaissance est mise en œuvre. Cette technique opère un découpage de l'image (caractère dans notre application) dans plusieurs directions. D'abord le centre de gravité de l'image est retenu comme centre de découpage à partir duquel la dite image est découpée :

- Soit horizontalement ce qui donne naissance à deux sous images : une sous image du haut et une sous image du bas. Chacune de ces sous image va faire l'objet d'une identification à part ;



Harf cad haut

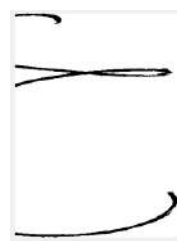


Harf cad bas

- Soit verticalement ce qui donne naissance à deux sous images : une sous image de gauche et une sous image de droite. Chacune de ces sous image va faire l'objet d'une identification à part ;



Harf cad gauche



Harf cad droite

- Soit dans les deux sens ce qui donne naissance à quatre sous images : une sous image haut gauche, une sous image haut droite, une sous image bas gauche et une sous image bas droite. Chacune de ces sous image va faire également l'objet d'une identification à part.

Harf cad haut gauche



Harf cad haut droite



Harf cad bas gauche

Harf cad bas droite

Le passage d'un découpage à l'autre est opéré selon les besoins de l'opération de reconnaissance. En effet si cette opération de reconnaissance de l'image complète a échoué, avec un certain pourcentage, on passe aux différents découpages dans un ordre bien défini.

Ci dessous on donne des exemples de découpages du caractère « Ghin » dans les trois directions citées ci-dessus.

Découpage horizontal

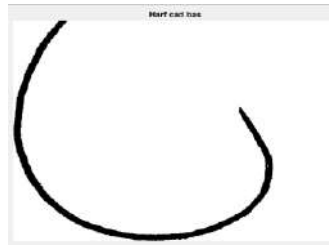
- Partie haute



Ghin HH cadrée

1.198483e+00 2.443065e-02 7.616934e-01 2.803205e-02 3.271513e-02 3.470779e-02
 1.303294e+00 3.195181e-02 5.982026e-02 2.770350e-02 3.685725e-02 9.205997e-03
 1.882194e+00 5.293346e-02 3.133280e-02 3.845300e-02 5.974894e-02 5.668014e-02
 4.879392e-02 6.183548e-02 2.415041e+00 3.002142e-02 2.307652e-02 1.607369e-02
 4.392763e-02 5.917516e-02 3.751835e-02 1.687371e-02 1.325464e-02 3.927006e-02

- partie basse



Ghin HB cadrée

1.188794e+00 6.170525e-02 8.515544e-01 4.943504e-02 9.362998e-02 1.463323e-02
 1.067300e+00 4.499350e-02 5.727572e-02 8.032028e-02 2.031287e-02 1.035444e-02
 1.766007e+00 1.865941e-02 4.730745e-02 1.349301e-02 3.096147e-02 4.336771e-02
 1.197861e-02 3.551558e-02 2.179124e+00 4.453134e-02 2.683436e-02 3.348381e-02
 2.836019e-02 4.544154e-02 5.595591e-02 9.539902e-03 1.093250e-02 4.379390e-02

Découpage vertical

- partie gauche



Ghin VG cadrée

1.160327e+00 2.337754e-02 7.450384e-01 7.753951e-02 4.081334e-02 5.373635e-02
 1.140017e+00 7.458388e-02 4.839689e-02 5.846900e-02 2.642857e-02 2.176106e-02
 1.652732e+00 1.180431e-02 6.461998e-02 5.228759e-02 5.230100e-03 2.075258e-02
 2.717802e-02 1.332125e-02 2.118261e+00 2.601918e-02 3.400624e-02 5.551260e-02
 3.526714e-02 3.416472e-02 1.884297e-02 1.717574e-02 4.299319e-02 5.423331e-02

- Partie droite



Ghin VD cadrée

1.164239e+00 1.836489e-02 7.452487e-01 3.510933e-02 1.407145e-02 7.065918e-02
 1.265694e+00 1.486814e-02 4.200973e-02 2.594755e-02 6.294610e-02 8.222729e-03
 1.745078e+00 2.022665e-02 7.271524e-02 9.841149e-03 2.369082e-02 4.553686e-02
 4.647449e-02 4.740774e-02 2.220159e+00 1.539186e-02 2.453982e-02 6.304239e-02
 2.579451e-02 5.873027e-02 7.076241e-02 6.860346e-02 7.282683e-02 5.181397e-02

Découpage horizontal et vertical

- partie bas gauche



Ghin GB cadrée

1.144930e+00 1.203482e-01 9.185388e-01 3.287499e-02 1.091893e-01 5.564878e-02
 1.150956e+00 9.543735e-02 1.678901e-02 8.599854e-02 7.380856e-02 5.519310e-02
 1.640987e+00 1.103889e-01 3.423801e-02 5.888029e-03 1.001244e-01 2.930109e-02
 1.617093e-02 4.001948e-02 2.296592e+00 4.210823e-02 1.524486e-02 4.400457e-02
 1.382911e-02 5.276088e-02 5.827354e-02 3.496103e-02 4.076215e-02 3.453609e-02

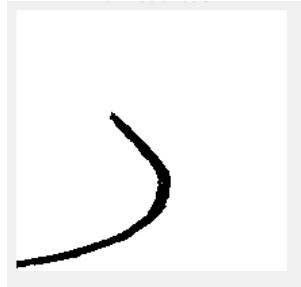
- partie haut gauche



Ghin GH cadrée

1.216361e+00 5.848869e-02 7.886713e-01 5.586474e-02 2.583968e-02 3.968620e-02
 1.342673e+00 5.102372e-02 1.667069e-02 3.470684e-02 5.551886e-02 1.306130e-02
 1.811221e+00 4.181168e-03 4.150740e-02 3.453060e-02 3.418966e-02 3.102885e-02
 1.610236e-02 5.040108e-02 2.297915e+00 3.449254e-02 4.920585e-02 1.353644e-02
 5.668623e-02 2.496559e-03 1.484848e-02 4.896156e-02 3.961761e-02 5.556969e-02

- partie bas droite



Ghin DB cadrée

1.198479e+00 4.376753e-02 6.938706e-01 3.934666e-02 3.852277e-02 3.805968e-02
 1.206979e+00 4.908879e-02 3.516783e-02 3.601644e-02 5.238160e-02 3.323138e-02
 1.731104e+00 5.252462e-02 4.252957e-02 3.227524e-02 3.936017e-02 5.947120e-02
 3.922863e-02 3.154816e-02 2.260974e+00 5.539142e-02 4.312814e-02 3.634784e-02
 3.315896e-02 5.200306e-02 5.964181e-02 4.441261e-02 3.285571e-02 3.004762e-02

- partie haut droite



Ghin DH cadrée

1.134239e+00 6.006318e-02 6.810572e-01 5.685628e-02 7.071087e-02 5.680386e-02
1.006499e+00 1.822710e-02 2.469562e-02 6.526119e-02 5.204030e-02 9.073630e-02
1.390578e+00 4.271860e-02 4.644786e-02 8.437769e-02 3.352029e-02 5.337540e-02
3.520024e-02 3.840099e-02 1.913937e+00 6.314975e-02 4.700617e-02 3.860325e-02
5.320959e-02 1.604409e-02 2.245798e-02 2.530674e-02 3.007335e-02 3.243894e-02

Chapitre IV :

Résultats et

discussions

IV.1 Introduction

Ce chapitre va mettre en œuvre plusieurs techniques de reconnaissance utilisées en traitement d'image. Bien sur ces techniques sont appliquées aux caractères arabes. Au préalable, il est important de signaler que ces techniques peuvent être appliquées aux images des caractères entiers, ou alors appliquées à des caractères découpés. C'est une nouvelle technique qu'on a mis en œuvre pour la première fois. Il s'agit, au lieu de prendre le caractère entier et lui appliquer un algorithme de calcul quelconque (par exemple calcul des moments), de prendre une partie de ce caractère et lui effectuer ce traitement. C'est une sorte de zoom sur le caractère pour essayer de voir plus finement les détails que contient celui-ci.

Ce découpage est opéré dans plusieurs directions, selon le choix de l'utilisateur, dans le but d'augmenter les chances de reconnaître ce caractère (voir chapitre trois pour les directions de découpage).

Différents scripteurs « teste » ont été utilisés pour faire la reconnaissance et on calcule le taux de reconnaissance général par différentes méthodes, et le taux final de la reconnaissance des caractères arabe manuscrit.

1^{ère} Méthode : Avec les moments de Zernike

Dans cette méthode on a utilisé des caractères non découpés (image complète) pour les scripteurs (scripteurs1, scripteurs2, scripteurs3, scripteurs4) [voir l'Annexe]

Aussi on a choisit d'après plusieurs ordres des moments de Zernike qu'on a calculés l'ordre 9 (ordre9= 30 moments de Zernike pour un caractère) .parce que cet ordre facilite les calculs de réseau de neurone et le calcul des banques de reconnaissance.

IV.2 Les bases de travail :

On a basé dans ce travail sur ce ci :

1^{ère} méthode :Les moments de Zernike

1) avec des caractères non découpés :

4scripteurs différentes et $4*28$ caractères = 112 caractères, $112* 30$ moments de Zernike = 3360 moments de Zernike.

2) avec des caractères découpés:

4scripteurs différentes et $4*28*4$ coupures = 448coupures, $448* 30$ moments de Zernike = 13440 moments de Zernike.

Pour quoi on a utilisé des réseaux de neurone ?

Dans la reconnaissance des caractères manuscrit on peut utiliser plusieurs méthodes. La méthode des algorithmes génétiques, La méthode de la logique floue ou La méthode des réseaux de neurones. Toutes ces méthodes sont nouvelles pour nous. Nous avons optés pour les réseaux de neurones tout simplement parce qu'elle est un peu lancée au sein de notre département et qu'on peut espérer trouver de l'aide si besoin. A noter cependant qu'après notre petit parcours avec cette méthode, on peut mentionner les points suivants :

- Elle est simple à mettre en œuvre. On peut même dire que ces réseaux sont bien adaptés à résoudre tout type de problème de reconnaissance tout l'enjeu est dans la phase d'apprentissage qui doit être menée le plus soigneusement possible;
- Les temps d'apprentissage du réseau peuvent être considérables. Ceci est dû au fait que certains paramètres de ce réseau (le coefficient d'apprentissage alpha et le momentum beta) sont déterminés par essais (tâtonnement) et que la structure même du réseau (le nombre de couches et le nombre d'éléments de chaque couche) est elle aussi déterminée par essai.

IV.3 Architecture du réseau de neurone utilisé

Le réseau utilisé est de type perceptron multicouche (MLP) à rétro propagation de l'erreur avec un apprentissage supervisé (chapitre des réseaux de neurones). On a utilisé un réseau à trois couches (une couche d'entrée, une couche cachée et une couche de sortie).

- a. La couche d'entrée :** le nombre de neurone de cette couche dépend du nombre de paramètres que peut générer chaque méthode; la méthode des moments de Zernike le nombre des neurones est fixé par essai.
- b. La couche cachée :** Le nombre de neurone de cette couche est choisit après plusieurs essais. Le meilleur nombre étant celui qui donne les meilleurs résultats. (par exemple : avec les moments de Zernike on a utilisé 32 neurones).
- c. La couche de sortie :** on a utilisé deux différents réseaux
 - Le réseau de neurone à une sortie codée en cinq bits donc le nombre des neurones de la couche de sortie est 5 neurones ($2^5=32 > 28$ nombre de caractères arabe)

Numéro de caractères de la couche de sortie	Le codage sur 5 bits
1	00000
2	00001
3	00010
4	00011
...	...
27	11011
28	11100

- Le réseau de neurone à une sortie non codée donc le nombre de neurones de la couche de sortie est égale au nombre caractères arabe donc 28.

Alif 1	Ba 2	Ta 3	Haa 26	Oua 27	Ya 28
-------------------------	-----------------------	-----------------------	-------	-------------------------	-------------------------	------------------------

Pour la suite de nos travaux on a travaillé avec la première structure.

La banque :

Une banque est la structure toute entière du réseau résultat d'une opération d'apprentissage. Elle comprend les éléments suivants :

- La structure proprement dite du réseau : Le nombre de couches et le nombre d'éléments de chaque couche ;
- Les poids entre les neurones des différentes couches. Ces poids sont évidemment le résultat d'apprentissage préalable.

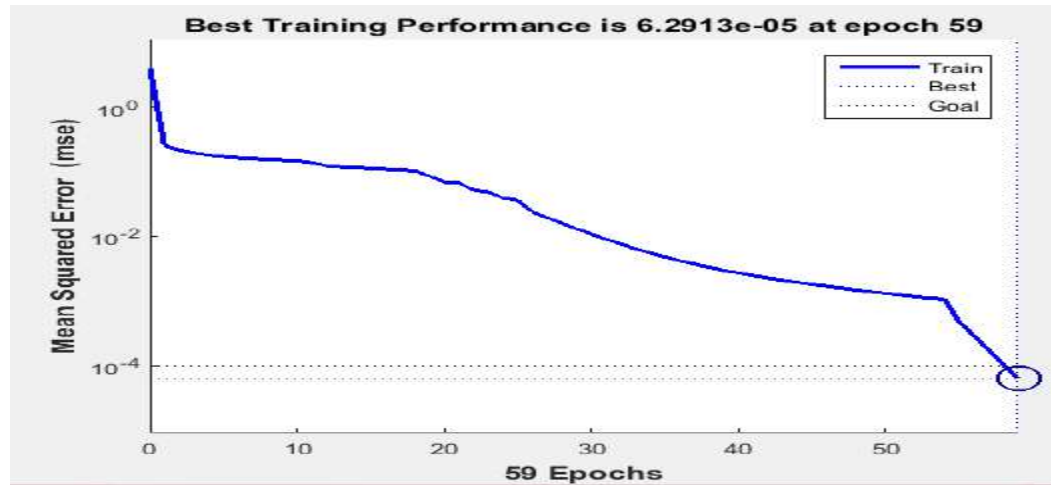
Cette banque est automatiquement créée par le programme une fois que la structure du réseau est définie par l'utilisateur, et que l'opération d'apprentissage est menée à terme.

Population :

On entend par population l'ensemble d'individus qui ont contribué à l'opération d'apprentissage. Une banque peut contenir plusieurs populations. Le programme est structuré de manière à ce qu'il permette d'enrichir une banque, initialement vide, avec une première population. D'ajouter à cette dernière une nouvelle population et ainsi de suite. Cette technique nous est dictée par le fait de donner au réseau plus de chance dans la phase de reconnaissance en lui donnant plus de catégories de gens qui ont contribué à sa création.

IV.4 Résultats de la reconnaissance avec les moments Zernike :

A. Caractères non coupés (complet)



Figure(IV.1) : l'entraînement de réseau de neurone (ZE_globale)

les caractères	Reconnaissance de script1 avec la banque 1	Reconnaissance de script2 avec la banque 2	Reconnaissance de script3 avec la banque3	Reconnaissance de script4 avec la banque 4
1 Ain	1	1	1	1
2 Alif	1	1	1	1
3 Ba	1	1	1	1
4 Del	1	1	1	1
5 Dha	1	1	1	1
6 Dhad	1	1	1	1
7 Dhel	1	1	1	1
8 Fa	1	1	1	1
9 Ghin	1	1	1	1
10 Ha	1	1	1	1
11 Haa	1	1	1	1
12 Gim	1	1	1	1
13 Kaaf	1	1	1	1
14 Kaf	1	1	1	1
15 Kha	1	1	1	1
16 Lem	1	1	1	1
17 Mim	1	1	1	1
18 Noun	1	1	1	1
19 Oua	1	1	1	1
20 Ra	1	1	1	1
21 Sad	1	1	1	1
22 Shin	1	1	1	1
23 Sin	1	1	1	1
24 Ta	1	1	1	1
25 Taa	1	1	1	1
26 Tha	1	1	1	1
27 Ya	1	1	1	1
28 Zi	1	1	1	1
	28/28	28/28	28/28	28/28

Le taux $\tau_{zer_1}=100\%$ $\tau_{zer_2}=100\%$ $\tau_{zer_3}=100\%$ $\tau_{zer_4}=100\%$

Tableau (IV.1). Les résultats de La reconnaissance de scripteur 1.2.3.4, l'un avec ça banque.

La reconnaissance de scripteur teste (moments Zernike)

les caractères	Reconnaissance de scripteur teste avec la banque 1.2.3.4
1 Ain	0
2 Alif	1
3 Ba	0
4 Del	1
5 Dha	0
6 Dhad	1
7 Dhel	0
8 Fa	0
9 Ghin	0
10 Ha	0
11 Haa	0
12 Gim	0
13 Kaaf	1
14 Kaf	1
15 Kha	0
16 Lem	0
17 Mim	1
18 Noun	0
19 Oua	0
20 Ra	0
21 Sad	1
22 Shin	0
23 Sin	1
24 Ta	0
25 Taa	0
26 Tha	0
27 Ya	0
28 Zi	0
Le Nombre des caractères connus	8/28

Le taux $\tau_{zer}=28.57\%$

Tableau (IV.2). Les résultats de La reconnaissance de scripteur de teste avec la banque général

B. La reconnaissance de scripteur teste (moments Zernike par la méthode de découpage)

B.1 Découpage horizontale :

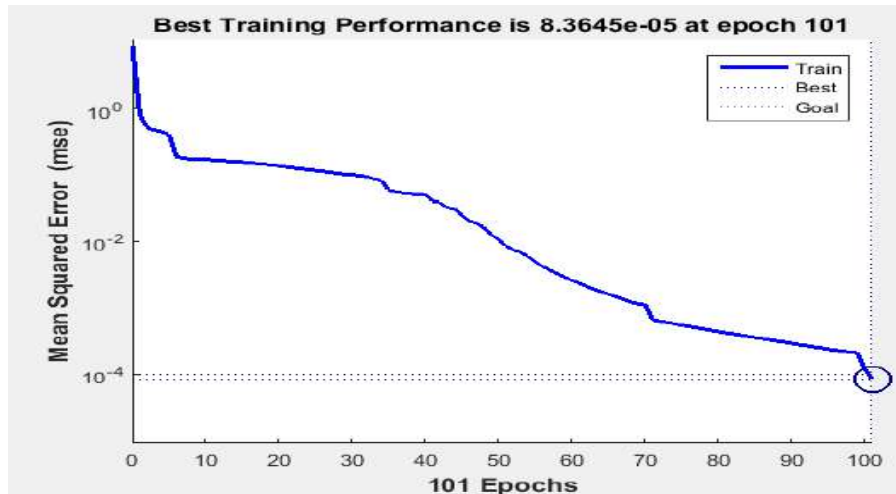


Figure(IV.2) :l'entraînement de réseau de neurone (ZE_haut)



Figure(IV.3) :l'entraînement de réseau de neurone (ZE_bas)

B.2 découpage vertical:

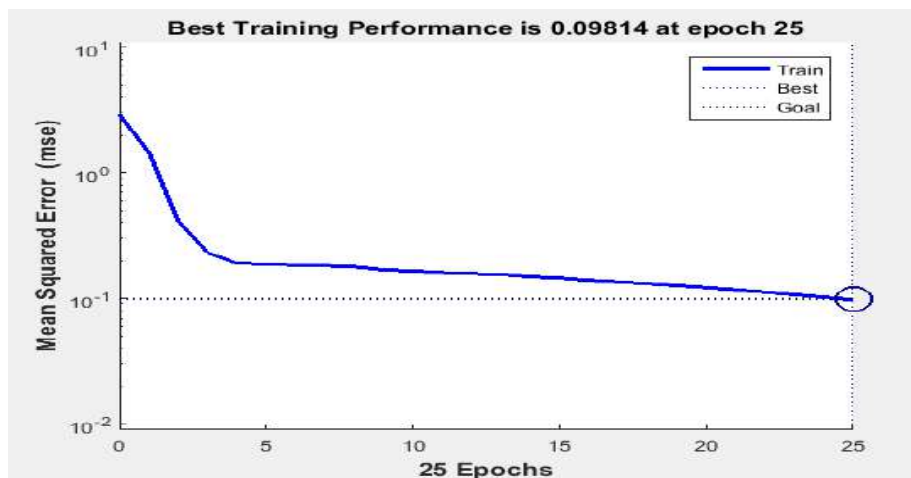


Figure(IV.4) : l'entraînement de réseau de neurone (ZE_gauche)

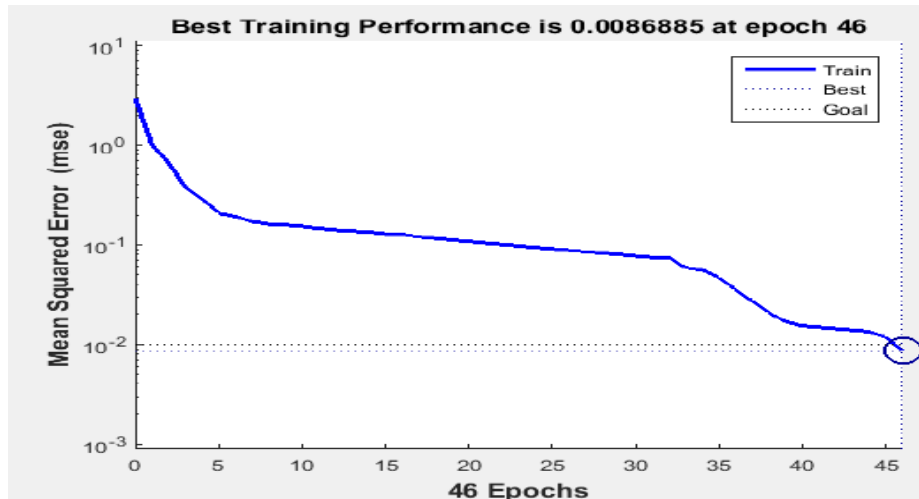


Figure(IV.5) : l'entraînement de réseau de neurone (ZE_droit)

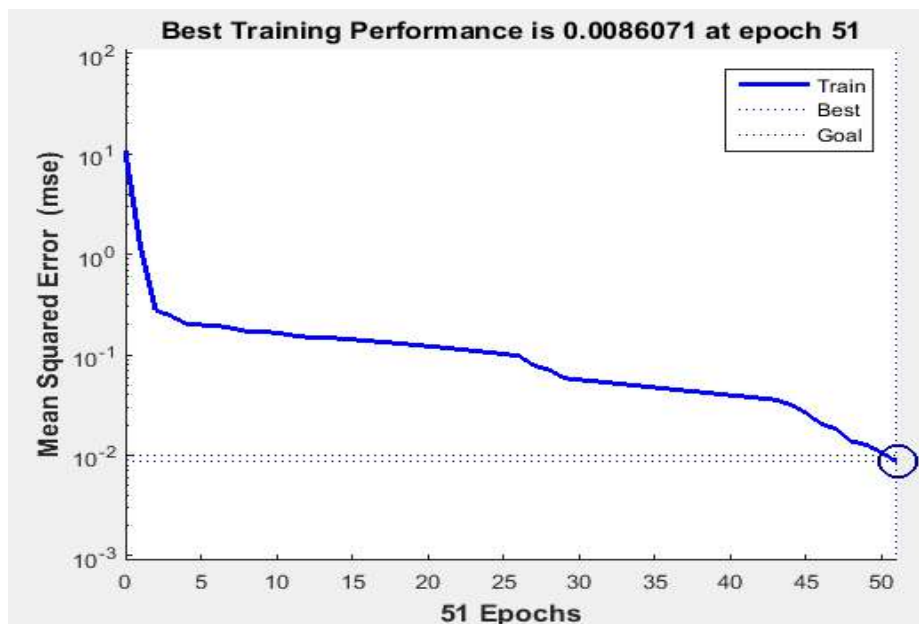
B.3 découpage horizontale verticale:(en quatre)



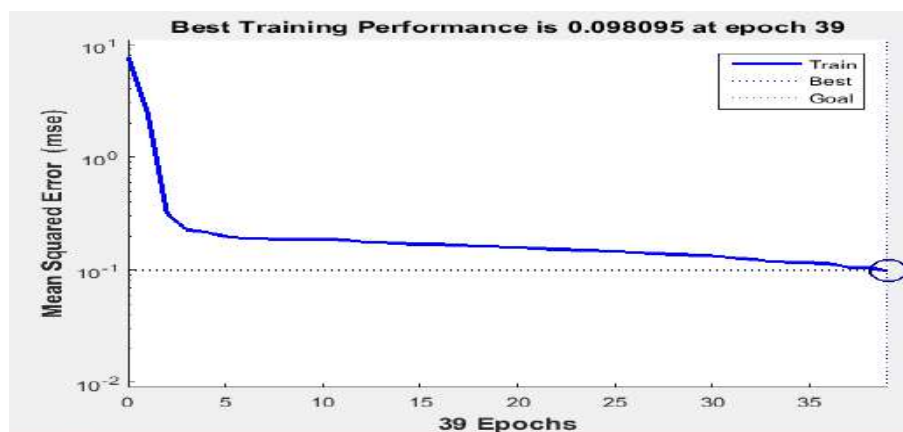
figure(IV.6) : l'entraînement de réseau de neurone (ZE_haut droit)



Figure(IV.7) : l'entraînement de réseau de neurone (ZE_bas droit)



Figure(IV.8) : l'entraînement de réseau de neurone (ZE_bas gauche)



Figure(IV.9) : l'entraînement de réseau de neurone (ZE_haute gauche)

		reconnaissance de scripteur teste par découpage							
	Les caractères	haut	bas	gouache	droit	Haut_droit	Haut_gouache	bas_droit	bas_gouache
01	Ain	0	0	0	0	0	0	0	0
02	Alif	1	1	1	0	1	1	1	0
03	Ba	0	0	0	0	0	0	0	0
04	Del	0	0	0	0	1	0	1	0
05	Dha	0	0	0	0	0	0	0	0
06	Dhad	0	0	0	0	1	1	1	0
07	Dhel	0	0	0	1	0	0	0	0
08	Fa	0	0	0	0	0	0	1	0
09	Ghin	0	0	0	0	0	0	0	0
10	Ha	1	0	0	0	0	0	1	0
11	Haa	0	0	0	0	0	0	0	0
12	Gim	0	0	0	0	0	0	0	0
13	Kaaf	1	1	1	0	1	0	1	0
14	Kaf	1	1	1	0	1	1	1	0
15	Kha	0	0	0	0	0	0	0	0
16	Lem	0	0	0	0	0	0	0	0
17	Mim	0	0	0	0	0	0	1	0
18	Noun	0	0	0	0	0	0	0	0
19	Oua	0	0	0	0	0	0	0	0
20	Ra	0	0	1	0	0	0	0	0
21	Sad	1	1	1	0	1	0	1	0
22	Shin	0	0	0	0	0	0	0	0
23	Sin	0	0	0	0	0	0	0	0
24	Ta	0	0	0	0	0	0	0	0
25	Taa	0	0	0	0	0	0	0	0
26	Tha	0	0	0	0	0	0	0	0
27	Ya	0	0	0	0	0	0	0	0
28	Zi	0	0	0	0	0	0	0	0
	Le Nombre des caractères connus	5/28	4/28	5/28	1/28	6/28	3/28	9/28	0/28
	Le taux	17.86%	14.29%	17.86%	3.57%	21.43%	10.71%	32.14%	0%

Conclusion générale

Conclusion générale

La reconnaissance des caractères manuscrits présente le sous-ensemble le plus difficile de la reconnaissance de formes. Cette difficulté provient du fait qu'on a plus de fluctuations présentées par chaque scripteur, le problème de chevauchement empêche de faire une bonne segmentation, car celle-ci est la base de tout système de reconnaissance.

C'est pour cela que beaucoup d'études sont basées sur la reconnaissance des mots entiers (pour connaître le caractère il faut le segmenter et pour le segmenter il faut le connaître).

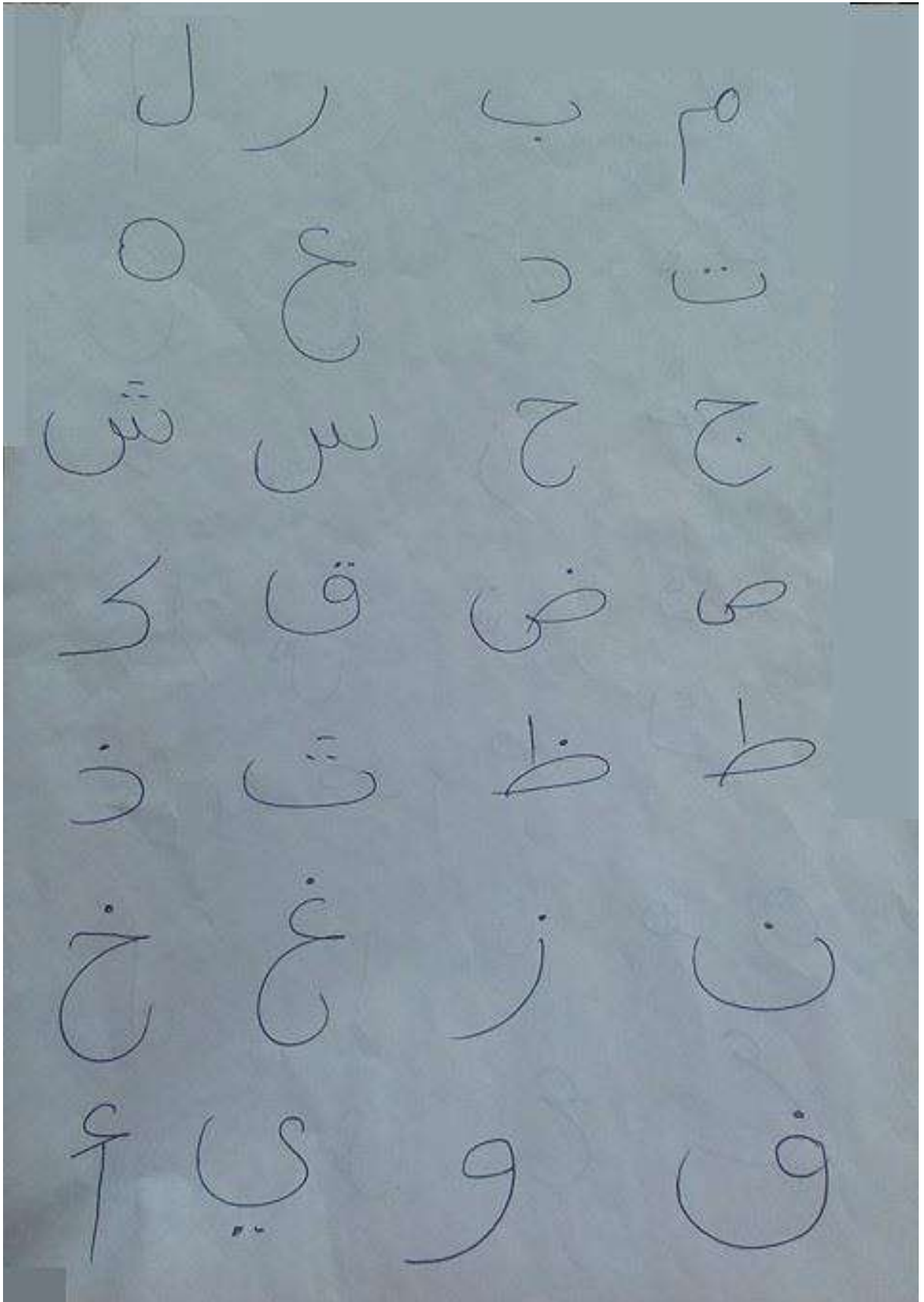
Dans notre travail les caractères à reconnaître sont limités en quatre scripteurs. Pour cela, nous avons utilisé les réseaux de neurones réputés pour leur performance dans ce domaine. Et vu l'utilisation limitée de notre système, le travail présenté se compose de deux parties (après la segmentation) :

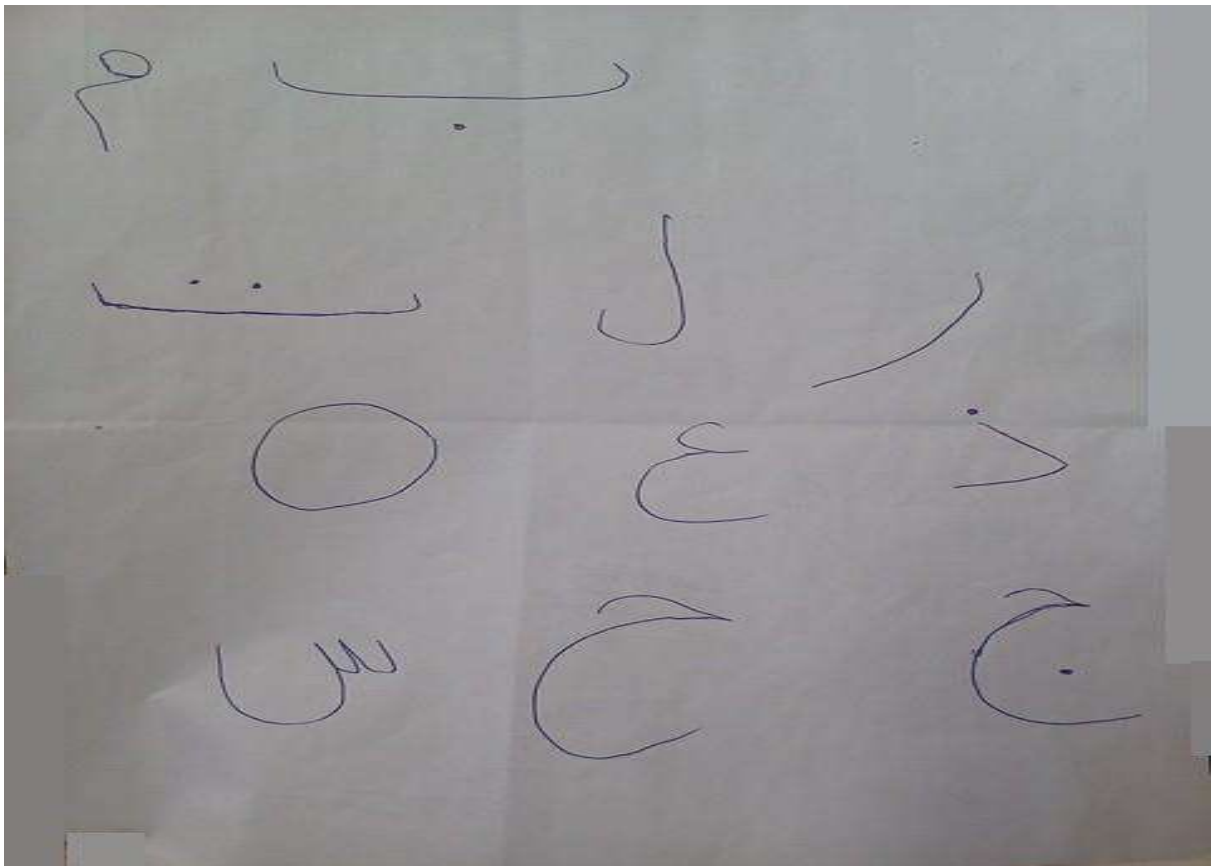
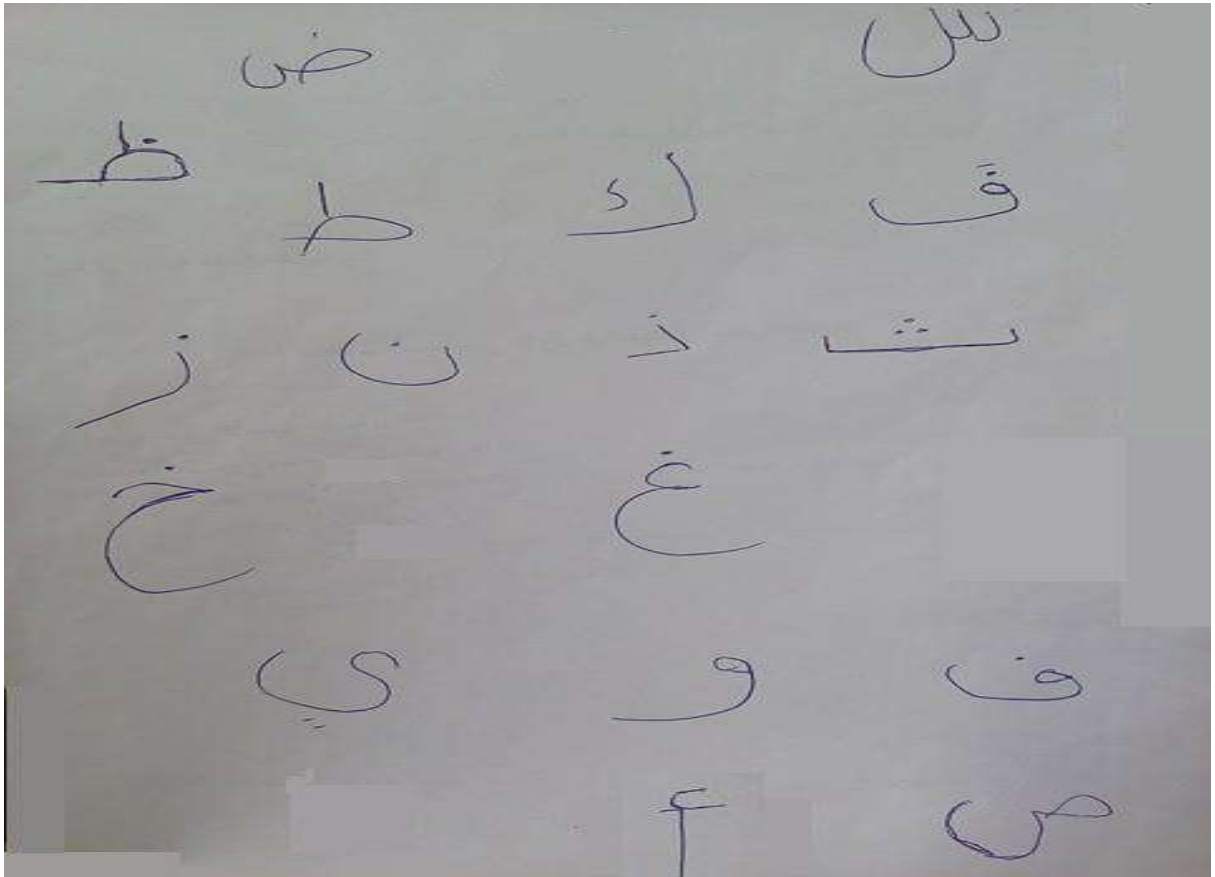
La première est l'extraction des caractéristiques (structurelles, géométriques et Zernike et statistiques).

La deuxième partie est le classificateur basé sur un réseau de neurone multicouche à une seule couche cachée.

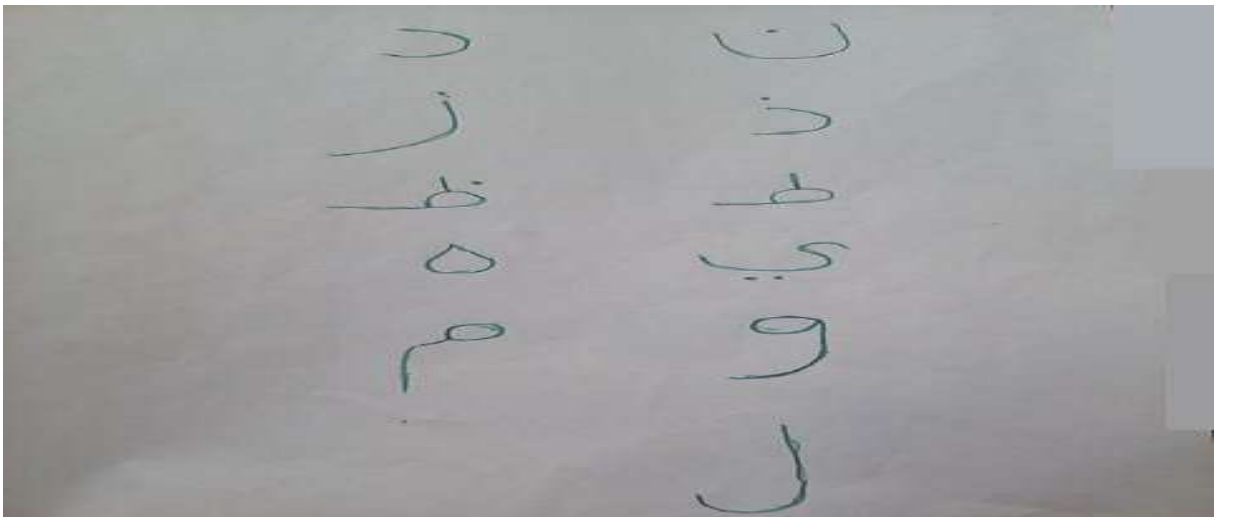
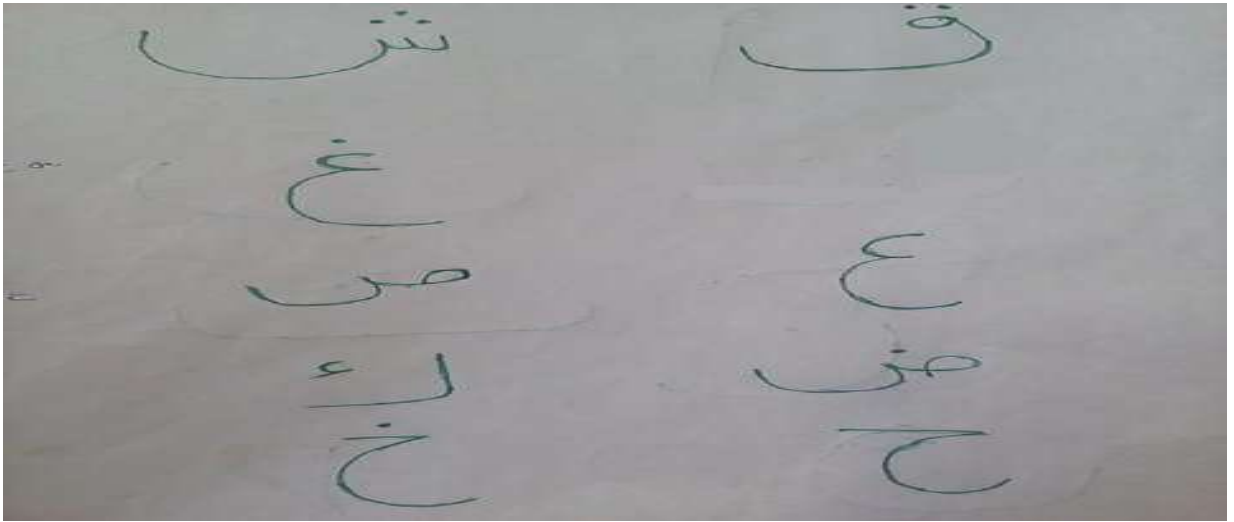
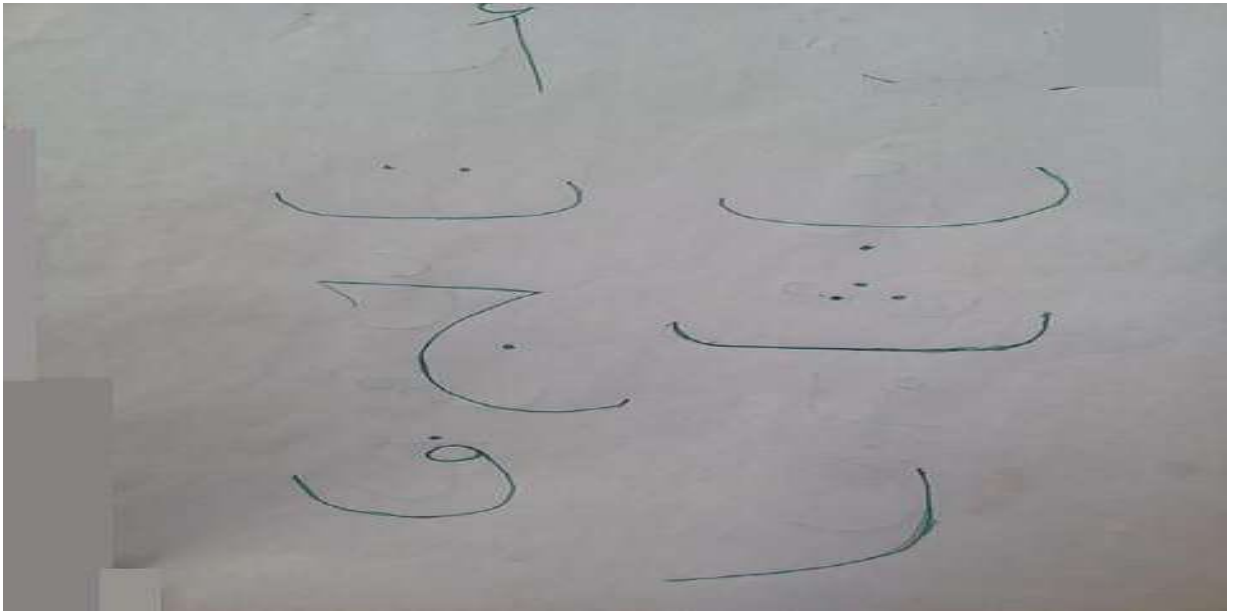
Malgré les efforts et les travaux réalisés dans ce domaine aucun système n'est fiable à 100%. Pour cela on considère que nous avons obtenu des résultats encourageants, et ce domaine reste ouvert aux propositions.

Scripteur :1

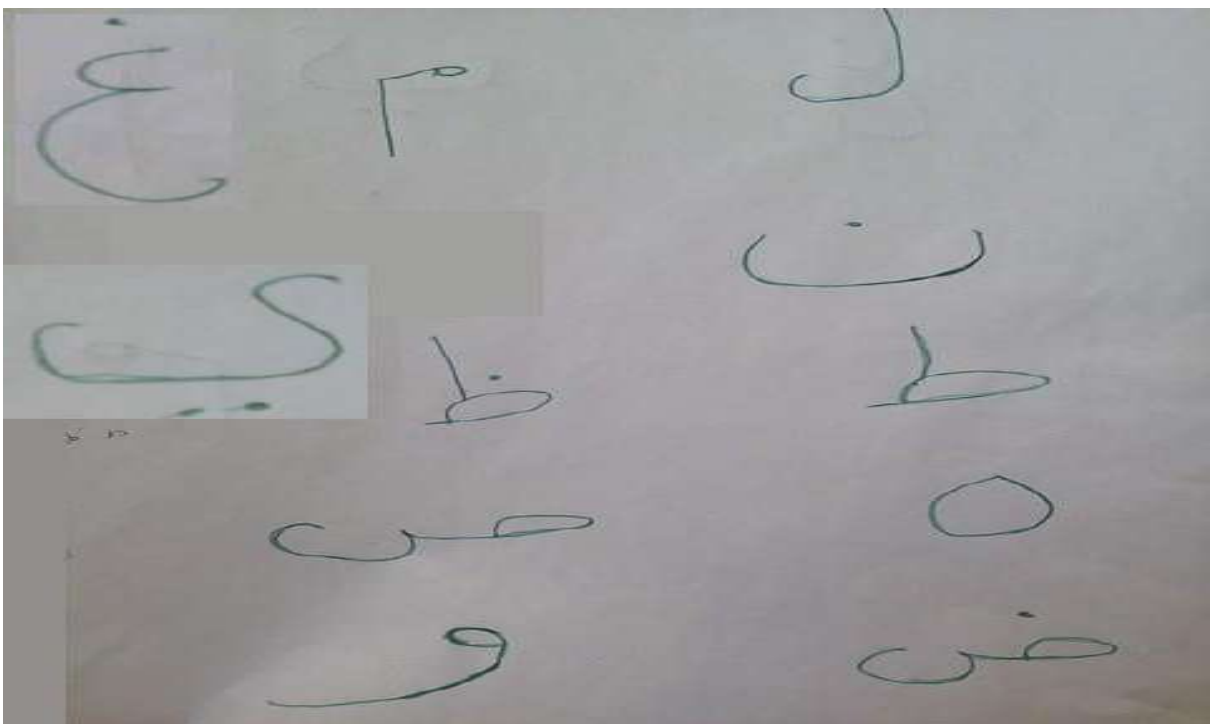
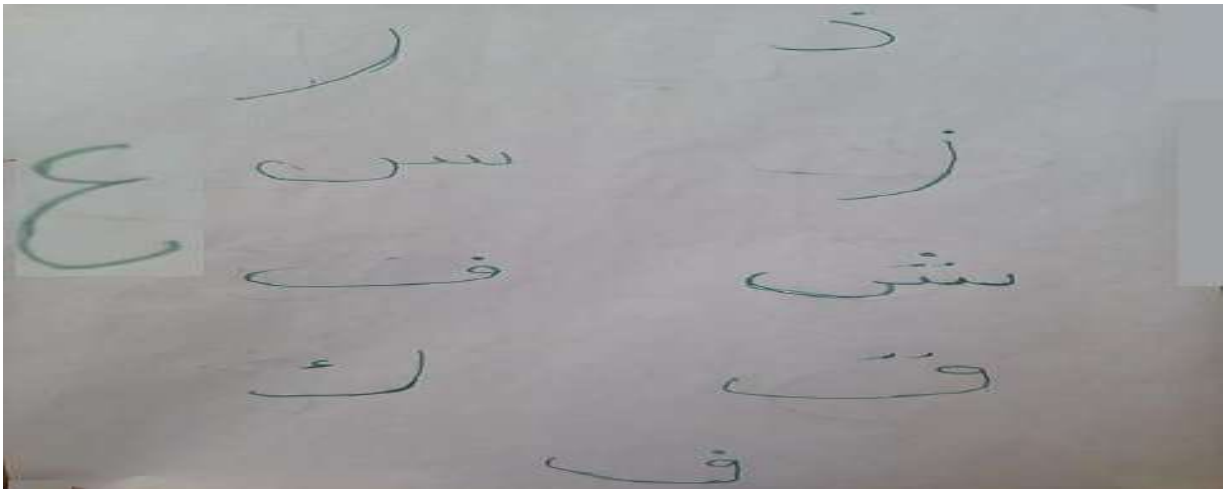


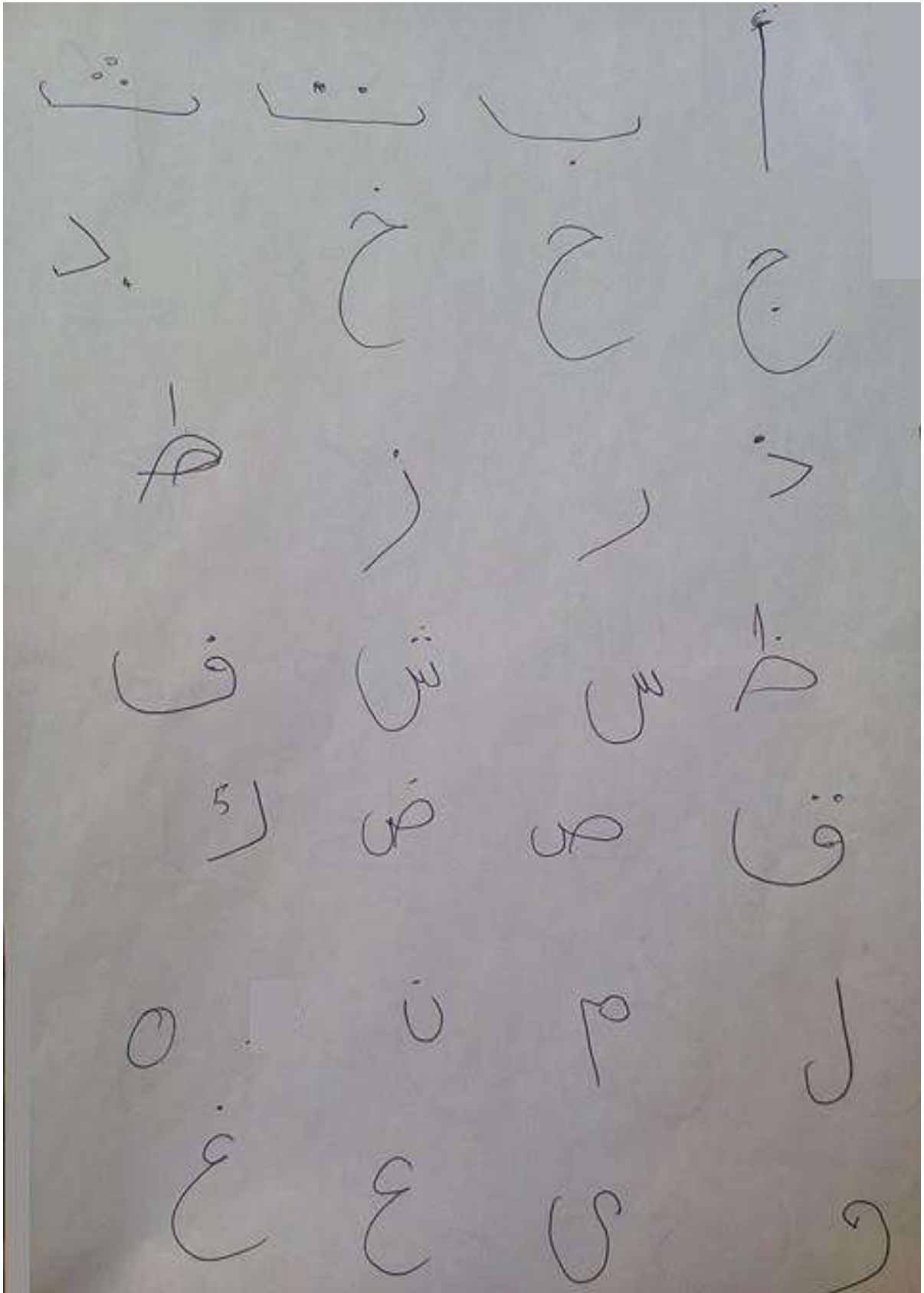


Scripteur :3



Scripteur :4





Bibliographie

- [1] B.GOSSELIN, « application de réseaux de neurones artificielles a la reconnaissance de caractères manuscrites », thèse doctorat en sciences appliquées .faculté polytechnique de Mons 95/96.
- [2] A.BELAID ,LORIA –CNRS ‘reconnaissance automatique de l’écriture et du documents ‘, compus scientifique B.P 239.Vandœuvre-lès-Nancy.
- [3] J.Wiliy ,Sons, ‘Moments and moment invariants in pattern recognition 2009 ‘Ltd ISBN :978-0-470-69987-4.
- [4] L Wang ,G Healey (1998) ‘using Zernike moments for the illumination and geometry invariant classification of multispectral texture’, IEEE transaction on image processing , vol 7,n2,pp196-203.
- [5] BELKASIM, I ,SHIDHAR, M and AHMADI,M (1991)’pattern recognition with moment invariants’ pattern recognition, vol24,n12,pp.1117-38.
- [6] M, Merciment, K, Gulez and TAREK VELIMUNCU ,’real object recognition using moment invariants’ vol 30,part 6,December 2005,pp.765-775.
- [7] Reddy et al., International Journal of Advanced Research in Computer Science and Software Engineering 3(4), April - 2013, pp. 140-143 .
- [8] ‘Invariant Moments Based Feature Extraction for Handwritten Devanagari Vowels Recognition’’ R. J. Ramteke Department of Computer Science North Maharashtra University, Jalgaon, (Maharashtra), INDIA 425001