

UN SYSTEME NEURO-FLOU POUR LA RECONNAISSANCE DE MONTANTS NUMERIQUES DE CHEQUES ARABES

N.E. Ayat^{1,2}

M. Cheriet^{1,2}

C.Y. Suen²

¹ Laboratoire d'Imagerie, Vision et d'Intelligence Artificielle, École de Technologie Supérieure, Université du Québec. 1100, notre Dame-Ouest, Montréal(Québec) H3C 1K3, Canada.

² Centre For Pattern Recognition and Machine Intelligence, Concordia University. 1455 de Maisonneuve Blvd. West, Suite GM-606, Montréal(Québec) H3G 1M8, Canada.

e_mail :ayat@livia.etsmtl.ca, cheriet@gpa.etsmtl.ca, suen@cenparmi.concordia.ca.

RESUMÉ

Nous proposons un système de reconnaissance de chaînes de chiffres indiens dédié à la lecture automatique des montants numériques de chèques arabes Saoudiens. Dans ce travail, nous présentons un nouveau jeu de primitives pour la caractérisation des chiffres du montant.

Le système développé s'articule autour de quatre modules distincts. Un module de prétraitements, un module de segmentation, un module de reconnaissance et un module de détection de symboles de délimitation. Ce dernier est un classifieur neuro-flou chargé d'extraire l'information topologique et contextuelle pour reconnaître les délimiteurs. Sa sortie sera combinée avec celle du module de reconnaissance pour la reconstitution du montant. Les résultats obtenus sur les bases de données VID et VCA de CENPARMI sont prometteurs.

Mots clés : *montant numérique, reconnaissance, morphologique, statistique, réseau de neurones, logique floue, classification, règles, apprentissage, inférence.*

1. INTRODUCTION

La lecture automatique des montants de chèques nécessite le décodage de leurs expressions en chiffres et en mots. Nous présentons dans ce travail une partie d'un système global de reconnaissance de chèques arabes en cours de développement à CENPARMI, dont la tâche est la reconnaissance des montants numériques (fig. 1).

Dans la littérature, très peu de travaux ont abordé des applications de lecture de chèques arabes. Ces derniers et plus particulièrement ceux du moyen orient, présentent deux caractéristiques spécifiques : le montant littéral est écrit en arabe, et les chiffres utilisés dans le montant numérique sont des chiffres indiens qui, hormis

leurs formes, utilisent le même codage décimal que les chiffres arabes. On y dénombre dix chiffres indiens.

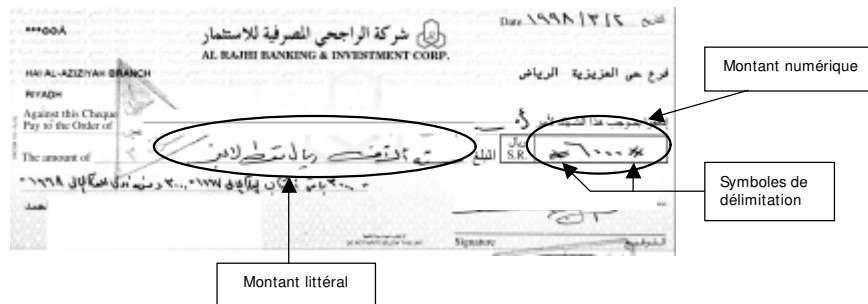


Fig. 1 Échantillon d'un chèque de la base de données.

Les images de montants numériques considérées présentent la difficulté de comprendre outre des chiffres, des symboles de formes assez variables. Ces derniers servent à indiquer le début de la chaîne de chiffres ou sa fin. Aussi, ces images peuvent contenir des tracés de bruits et des ratures. D'où la complexité de la tâche de reconnaissance. Cependant, ces images de montant sont composées de chaînes de chiffres isolés, rendant moins complexe l'opération de segmentation.

Quatre modules composent notre système de lecture automatique de montants numériques. Un module de prétraitements, un module de segmentation, un module de reconnaissance et un module de détection de symboles de délimitation (fig. 2). La phase d'agrégation ne sera pas abordée dans cet article.

Dans (sect. 2) nous décrivons les opérations de prétraitements réalisées. La partie segmentation est brièvement décrite dans (sect. 3). Afin de reconnaître les chiffres du montant, les éléments segmentés sont traités par un réseau de neurones à rétropropagation de gradients. Ce dernier est décrit dans (§ 4.2). L'extraction des primitives est auparavant détaillée dans (§ 4.1). Par ailleurs, nous avons réalisé un classifieur à base de logique floue chargé de traiter les tracés qui ne s'apparentent pas à des chiffres. Ce module a pour tâche de détecter les symboles de délimitation qui existent dans l'image de montant en extrayant l'information topologique et contextuelle inhérente. Les résultats de ce dernier seront combinés avec le module de reconnaissance pour la reconstitution de la chaîne de chiffres ASCII. Le module de détection de symboles de délimitation est décrit dans (sect. 5). Enfin dans (sect. 6) nous présentons une conclusion.

2. PRETRAITEMENTS

L'image de montant extraite subit divers traitements avant son découpage et sa reconnaissance. Ainsi, pour pallier aux coupures survenues durant les phases d'acquisition et de binarisation, nous procédons à un épaississement du contour externe du tracé. Ceci permet d'éviter la fermeture d'éventuelles boucles de taille petite; gardant intactes des propriétés morphologiques importantes dans l'image. Un filtrage est ensuite réalisé pour adoucir les contours. Par ailleurs, la ligne de base du tracé n'étant pas toujours horizontale nous appliquons une procédure de normalisation dans le but de ramener l'orientation du signal d'écriture à l'horizontale. Nous avons utilisé l'algorithme de Madvanath décrit dans (art. 1). La ligne de base calculée est l'axe d'inertie de l'image.

Pour corriger les recouvrement verticaux présents dans l'image, la démarche que nous avons choisie est celle proposée par Guillevic & al. dans (art. 2). L'estimation de l'inclinaison moyenne des chiffres est basée sur le calcul d'histogrammes. Ces derniers sont calculés à des orientations différentes et

régulières. L'orientation de l'histogramme correspondant à la dérivée première maximale est celle retenue.

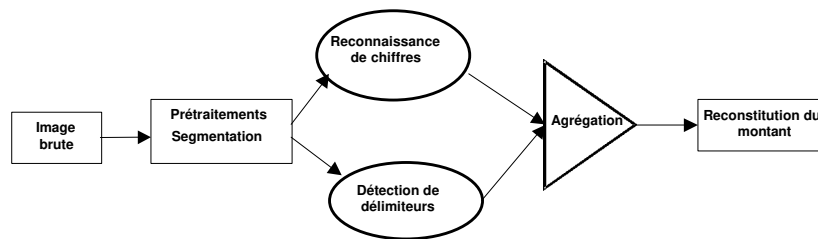


Fig. 2 Architecture du système.

3. SEGMENTATION

Le but de cette étape est de générer un découpage de l'image en éléments susceptibles d'être reconnus par le classifieur. La segmentation réalisée est une segmentation explicite par analyse des contours et extraction des composantes connexes. Aussi, préalablement à la segmentation, nous réalisons un filtrage sur l'image. Ainsi, les segments de droites, les ratures situées aux extrémités de l'image et les bruits sont supprimés. La détection du contour se fait par suivi en run-length des pixels noirs de l'image.

4. RECONNAISSANCE

Nous décrivons dans cette partie, le module de reconnaissance de chiffres indiens que composent les montants de chèques arabes Saoudiens (fig. 3). Les primitives choisies pour décrire nos entités sont des primitives hybrides injectées dans une même machine de classification. Le module de reconnaissance est un réseau de neurones à retropagation de gradients d'une couche cachée. L'ensemble de caractéristiques choisi et son extraction sont décrits dans (§ 4.1). Quelques résultats sont présentés dans (§ 4.3).

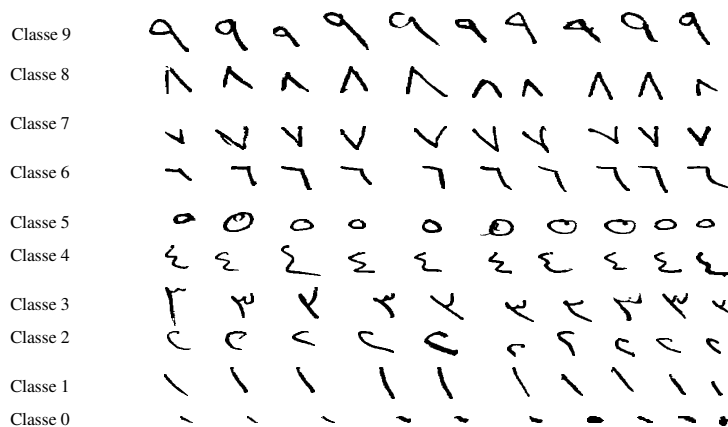


Fig. 3 Quelques échantillons de la base de données de chiffres.

4.1 Extraction des caractéristiques

Les systèmes de reconnaissance de l'écriture nécessitent généralement deux étapes : une étape d'extraction de primitives et une étape de classification dans laquelle des règles de décision pour séparer les classes sont définies.

Dans la littérature, plusieurs travaux ont porté sur l'élaboration de nouvelles primitives à même de procurer un pouvoir discriminant inter-classes accru tout en minimisant la variabilité intra-classes. Ces primitives sont généralement classées en deux familles : les primitives morphologiques (boucles, concavités, dépassements, points extrêmes du tracé, intersections, ...etc) et les primitives statistiques qui dérivent des mesures de distribution spatiale des pixels (zonage, moments invariants, descripteurs de Fourier...etc). Les caractéristiques morphologiques et statistiques sont complémentaires dans la mesure où deux sortes de propriétés sont mises en relief. Le système visuel humain semble appréhender les mêmes approches pour reconnaître des scènes ou des objets.

Afin de caractériser nos images de chiffres, nous avons opté pour des primitives hybrides combinant des statistiques sur le contour et sur les pixels définissant certaines formes morphologiques. Ceci revient à opérer une reconnaissance grossière via les propriétés structurelles et une reconnaissance plus raffinée par analyse des distributions spatiales des formes composant l'objet à reconnaître. Ces statistiques constitueront un seul vecteur de primitives. Nous décrivons ci-dessous les primitives choisies et leur extraction.

4.1.1 Primitives morphologiques

L'information locale comme celle contenue dans les gradients et courbures des pixels du contour est souvent non suffisante pour pouvoir représenter un motif donné. Par ailleurs, l'aspect morphologique est très important dans la perception humaine, et son étendue spatiale est tout aussi importante. Dans ce sens, nous avons utilisé un premier jeu de caractéristiques basé sur les régions.

Cheriet & al. dans (art. 3) et Strathy & al. dans (art. 4) ont utilisé les régions pour la segmentation de mots et des chaînes de chiffres. Néanmoins, la caractérisation des images de caractères en analysant certaines formes géométriques y figurant a été aussi l'objet d'autres travaux tels que ceux exposés dans (art. 5), (art. 6) et (art. 7). Heutte & al. dans (art. 8) se sont assez étendus sur le sujet.

La moitié des chiffres indiens comprend des formes géométriquement simples et assez discriminantes. Ces chiffres correspondent aux classes 7, 8, 5, 9 et le 3 qui contiennent respectivement une région en vallée, une région en bosse, une boucle, une boucle, et deux régions en vallée (fig. 4). Ces régions élémentaires serviront pour la reconnaissance (tab. 1).

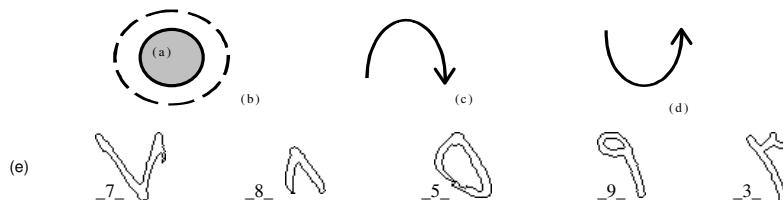


Fig. 4 (a) Région en boucle; (b) région ouverte; (c) région en bosse; (d) région en vallée; (e) chiffres contenant des régions discriminantes.

Lors de l'extraction des primitives, l'image est divisée en une grille de 16 zones égales. Le nombre de pixels du contour dans chacune des zones, appartenant à chacune des 4 régions est déterminé. Il en résulte un vecteur de 4 valeurs pour chaque zone de la grille (fig. 5).

Nombre de pixels de région en boucle	Nombre de pixels de région en vallée	Nombre de pixels de région en bosse	Nombre de pixels de région ouverte
--------------------------------------	--------------------------------------	-------------------------------------	------------------------------------

Fig. 5 Vecteur de primitives morphologique de zone.

	(a)	(b)	(c)	(d)	(e)	(f)
(9)	٩	٩	٩	٩	٩	٩
(8)	٨	٨	٨	٨	٨	٨
(7)	٧	٧	٧	٧	٧	٧
(6)	٦	٦	٦	٦	٦	٦
(5)	٥	٥	٥	٥	٥	٥
(4)	٤	٤	٤	٤	٤	٤
(3)	٣	٣	٣	٣	٣	٣
(2)	٢	٢	٢	٢	٢	٢
(1)	١	١	١	١	١	١
(0)	٠	٠	٠	٠	٠	٠
(virgule)	,	,	,	,	,	,

Tableau 1 Régions des classes considérées: (a) les 11 classes de caractères; (b) leurs contours; (c) boucles; (d) régions en bosse; (e) régions ouvertes; (f) régions en vallée.

4.1.2 Primitives statistiques

Les primitives statistiques utilisées sont basées sur le code en chaîne du contour des caractères. Ce dernier est calculé par un suivi en run-length de l'image. Un adoucissement du contour est par la suite réalisé en appliquant un moyenneur de longueur cinq à chaque pixel des frontières. Cet ensemble de primitives est un vecteur de 13 valeurs représentant les statistiques sur les directions de Freeman et les courbures des pixels dans chacune des 16 zones définies par la grille qui englobe l'image. Soit un vecteur de $13 * 16 = 208$ valeurs. Les valeurs de courbure calculées sont quantifiées en cinq valeurs. L'ensemble des primitives est composé donc de $(8+5+4) * 16 = 272$ valeurs. Afin de normaliser ces primitives, nous avons converti ces nombres en pourcentage par rapport aux nombre de pixels du contour dans chaque zone (fig. 6).

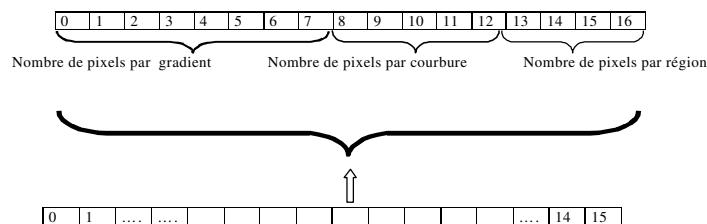


Fig. 6 Format du vecteur de primitives.

4.2 Classification

Lors de la classification, nous avons utilisé un réseau de neurones à rétropropagation de gradients d'erreurs à une couche cachée. Les 272 primitives développées précédemment sont les entrées du réseau de neurones. La couche cachée est composée de 40 neurones. Les classes à discriminer sont les 10 chiffres plus la virgule, d'où le choix de 11 neurones pour la couche de sortie. La fonction d'activation des neurones est la tangente hyperbolique. L'initialisation des poids et biais du réseau est réalisée suivant une distribution gaussienne. A noter aussi que ces poids sont normalisés selon le fan-in des neurones correspondants. La base de données de chiffres indiens VID utilisée pour l'apprentissage du réseau de neurones a été extraite à partir d'images de montants numériques associées à la base de chèques Saoudiens développée récemment à CENPARMI. Elle comporte un ensemble de 6630 images dont 4682 images ont été utilisées pour l'apprentissage et 1948 images pour le test.

4.3 Résultats expérimentaux

Nous avons effectué plusieurs manipulations avec différents cas de figures. Pour un seuil de rejet égal à zéro, nous avons obtenu des taux de reconnaissance de 98.72% sur l'ensemble d'apprentissage et 95.87% sur l'ensemble de test. Le taux d'erreur sur l'ensemble de test est de 2.71%. Nous rapportons dans (tab. 2) et (tab. 4(a)) respectivement les taux de reconnaissance par classe et la matrice de confusion obtenus.

Classe	0	1	2	3	4	5	6	7	8	9	10
TRPC(app)	0.991	0.980	0.997	1.000	0.984	0.997	0.981	1.000	0.993	0.991	0.864
TRPC(test)	0.968	0.921	0.969	0.976	0.922	0.994	0.972	0.941	0.949	0.959	0.846

Tableau 2 Taux de reconnaissance par classe sur les ensembles d'apprentissage et de test.

Les résultats montrent que les deux classes les plus confuses sont les classes 0 et 10 représentant respectivement le zéro décimal et la virgule. Cette confusion est due à la grande similarité qui caractérise chacune des classes. En fait, l'humain lui-même déciderait difficilement sur l'identité de la virgule si ce n'est l'information de position et le contexte. Enfin dans pareils cas, il nous a paru plus judicieux de prendre en compte, lors de la reconnaissance du montant, la position du segment et son contexte pour trancher.

Nous avons refait l'apprentissage en considérant seulement les 10 classes de chiffres. La virgule étant omise, les résultats sont meilleurs. Ainsi, nous avons obtenus des taux de reconnaissance de 99.69% sur l'ensemble d'apprentissage et 96.69% sur l'ensemble de test. Les taux d'erreur et de rejet sur l'ensemble de test étant respectivement 1.99% et 1.21%. Les tableaux (tab. 3) et (tab. 4(b)) présentent respectivement les résultats de simulation pour les taux de reconnaissance par classe et la matrice de confusion obtenus.

Classe	0	1	2	3	4	5	6	7	8	9
TRPC(app)	0.999	0.985	0.997	1.000	0.989	1.000	0.994	1.000	1.000	0.991
TRPC(test)	0.983	0.926	0.953	0.964	0.922	0.988	0.972	0.918	0.949	0.938

Tableau 3 Taux de reconnaissance par classe sur les ensembles d'apprentissage et de test.

	0	1	2	3	4	5	6	7	8	9	10
0		9				2					15
1	8						1			1	4
2	1			1	1	1					
3			1								
4	1	1	4								
5	1										
6			1								
7		2		1							1
8	1										
9	1			1							
10	9										

	0	1	2	3	4	5	6	7	8	9
0			12			1	2		1	
1	9									
2	1			2		1				1
3										
4	1	1	5							
5	1									
6			1							
7			1		2					
8	1									
9	1			1						

Tableau 4 (a) Matrice de confusion sur l'ensemble de test (cas des 11 classes); (b) matrice de confusion sur l'ensemble de test(cas des dix classes).

5. DETECTION DE DELIMITEURS

Les montants de chiffres dont nous disposons ne sont pas exempts de bruits ou de tracés indésirables. Par ailleurs, les procédures de prétraitements appliquées n'ont pour tâche que l'élimination des segments de droites dus à une mauvaise saisie de l'image, les petits tracés, ou les tracés présents aux extrémités haute et basse du montant.

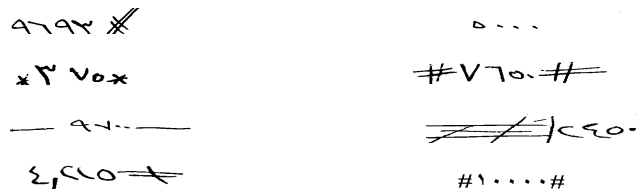


Fig. 7 Quelques échantillons de la base de données de montants du CENPARMI.

Une analyse exhaustive de la base de données des images de montants disponible nous a permis de dresser une synthèse des différentes configurations existantes (fig. 7). Ces dernières peuvent être représentées par la syntaxe : [<délimiteur> <montant> [<délimiteur>] ; l'élément [<délimiteur>] étant optionnel.

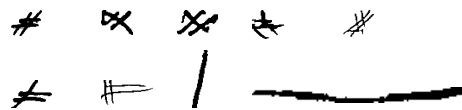


Fig. 8 Quelques symboles de délimitation présents dans les montants.

Ne sachant pas à priori le nombre de chiffres du montant à reconnaître, il nous est difficile de détecter la position exacte de l'information utile dans le tracé. Aussi, la variabilité de la taille des chiffres indiens (la classe zéro par exemple qui est de taille petite) rend l'estimation exacte du nombre de chiffres qui existent dans la chaîne difficile sans passer par une préclassification additionnelle.

Afin d'extraire la chaîne de chiffres utile, il incombe au système de détecter de façon fiable les tracés qui correspondent à des symboles de début de chaîne, de fin de chaîne ou de rature qui délimitent le montant (fig. 8). Une première approche intuitive suggère l'utilisation du classifieur de chiffres avec option de rejet pour la détection des symboles de délimitation. Ainsi, le réseau de neurones étant entraîné

seulement sur les 10 classes de chiffres indiens et la virgule, ne saura reconnaître des motifs qui lui sont inconnus. Un seuil de rejet bien choisi permet de prédire les chiffres des autres symboles à écarter. Cependant cette solution n'est pas suffisante pour gérer la variabilité de nos symboles de délimitation. En effet, mis à part leurs formes, d'autres informations topologiques peuvent y être extraites pour discriminer davantage les chiffres des non-chiffres qui existent dans le montant. De la, il est important de combiner les deux sources d'informations afin de trancher sur la nature du tracé traité. Dans le même ordre d'idées, nous avons développé un classifieur flou à base de règles pour détecter les délimiteurs. L'objectif est de prendre en compte l'information de connexité et les propriétés topologiques intrinsèques aux images de montants dans le processus de prise de décision.

5.1 Classification : approche hybride neuro-floue

Notre but ultime étant d'extraire la chaîne de chiffres du montant, nous avons développé un classifieur flou dont les sorties doivent être combinées avec celle du module de reconnaissance de chiffres pour la détection des délimiteurs.

Dans les images de montants numériques, les symboles de délimitation ne sont pas contraint à des modèles particuliers (fig. 8). D'où la difficulté de la tâche de détection qui devra considérer la variabilité de ces symboles avec toutes les configurations présentes. Une façon de procéder est de poser des conditions d'appartenance sous forme d'un ensemble de clauses logiques dont le comportement dépend d'un ensemble des primitives que nous développons dans (§ 5.2). Ces primitives sont au nombre de 11 et définissent les variables d'entrée de nos règles floues. Ces dernières sont apprises selon les configurations des motifs qui sont présentés au classifieur. Nous avons utilisé une architecture similaire à celle proposée dans (art. 9). Cette architecture est en quelques sortes une approche connexionniste de la logique floue. Les règles de classification et les fonctions d'appartenance des ensembles flous qui définissent nos primitives sont apprises sur les images de montants numériques. Les paramètres obtenus après apprentissage seront interprétés sous formes de règles SI-ALORS. Nous décrivons ci-dessous l'ensemble de primitives que nous avons choisi pour caractériser les délimiteurs. Une description de l'architecture du classifieur ainsi que ses règles d'apprentissage sont aussi fournies dans (§ 5.3). Enfin, quelques résultats sont présentés dans (§ 5.4) pour valider notre choix.

5.2 Choix de primitives

Nous avons défini des critères géométriques et topologiques (fig. 9(a)) basés sur les mesures suivantes :

- rapport de la hauteur sur la longueur de la fenêtre englobant l'image,
- hauteur du segment courant normalisée à la moyenne des hauteurs des segments courant et suivant,
- hauteur du segment courant normalisée à la moyenne des hauteurs des segments courant et précédent,
- largeur du segment courant normalisée à la moyenne des largeurs des segments courant et suivant,
- largeur du segment courant normalisée à la moyenne des largeurs des segments courant et précédent,
- nombre de pixels du segment courant,
- rapport des masses du segment courant avec le segment suivant,
- rapport des masses du segment courant avec le segment précédent,
- distance du segment courant par rapport au segment suivant,
- distance du segment courant par rapport au segment précédent,

- nombre de points de jonction du segment courant (JPN) (fig. 9(b)).

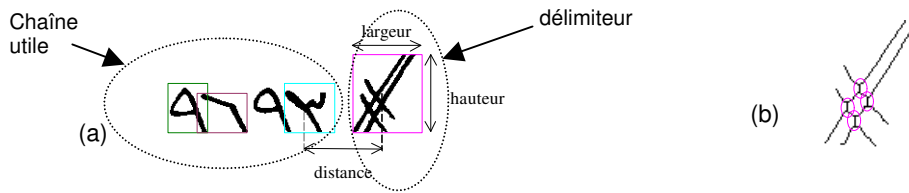


Fig. 9 (a) Quelques mesures géométriques relatives à un délimiteur ; (b) délimiteur aminci avec 8 points de jonction.

5.3 Architecture

Le classifieur développé est composé de trois couches : une couche d'entrée, une couche de règles et une couche de sortie (fig. 10). Il comporte 11 entrées et trois sorties. Ses entrées sont les primitives détaillées ci-dessus. Les sorties représentent les trois décisions possibles que peut prédire le modèle. Ainsi, la décision pour le tracé traité peut être un délimiteur situé avant la chaîne de chiffres, un délimiteur situé après la chaîne de chiffres ou un chiffre. La couche d'entrée fournit le degré de vérité de chaque primitive sur l'ensemble des fonctions d'appartenance définies. Le degré de vérité d'une entrée correspond à la fonction d'appartenance de la partition qui fournit le plus haut degré. L'apprentissage du classifieur se fait en deux étapes. Une étape de génération de règles et une autre d'adaptation des fonction d'appartenance. La génération de règles s'opère selon la distribution de l'ensemble d'apprentissage en termes linguistiques flous. Durant cette phase, chaque fois qu'un motif d'apprentissage est présenté au classifieur, ce dernier vérifie si ce motif est déjà représenté par un **nœud** de la couche de règles. Sinon, il crée un nouveau **nœud** dans cette couche. Ces **nœuds** réalisent un ET sur les ensembles flous associés (art. 10).

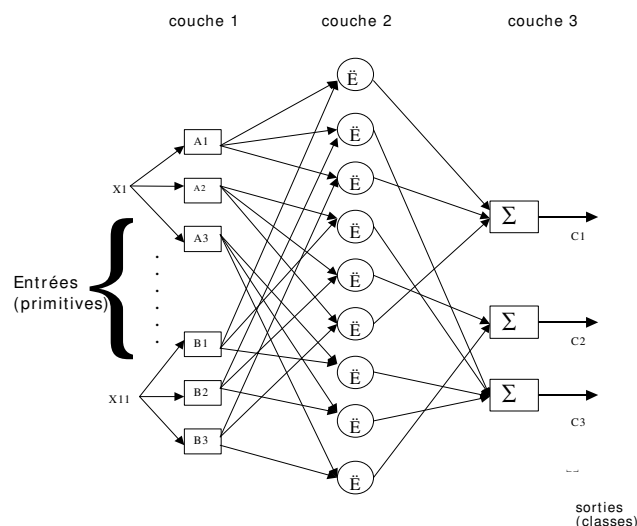


Fig. 10 Prototype du classifieur neuro-flou utilisé.

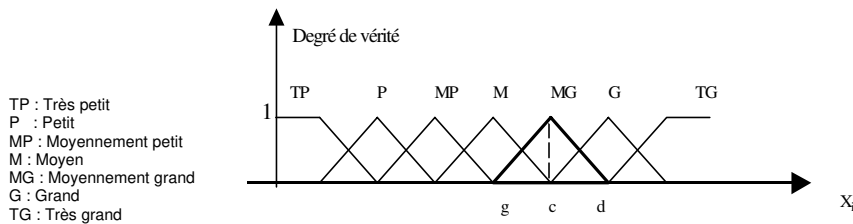


Fig. 11 Répartition initiale des fonctions d'appartenance d'une entrée du classifieur.

L'étape de création de clauses terminée, l'algorithme doit adapter la forme de chaque fonction d'appartenance de chacune des partitions des entrées du classifieur (fig. 11). Nous avons utilisé des fonctions d'appartenance triangulaires. L'algorithme d'adaptation utilisé est la correction par rétropropagation d'erreurs. L'adaptation des fonctions d'appartenance consiste à modifier les paramètres g , c et d (fig. 11) qui définissent leurs formes de façon à réduire l'erreur quadratique calculée à la sortie du classifieur. La classification est réalisée en faisant une opération de sommation sur les nœuds de la couche de règles associés à chaque classe et à considérer celle procurant la plus haute valeur de sortie.

5.4 Résultats

Pour l'apprentissage du détecteur de délimiteurs, nous avons utilisé 485 images issues de la base de données de montants VCA de CENPARMI qui contient 1327 images de chèques. L'apprentissage a été fait sur les deux segments extrêmes de part et d'autre du montant, soit 4 segments par montant. 1932 motifs différents ont été présentés au classifieur. Nous avons considéré un classifieur neuro-flou avec au maximum trois partitions pour chaque entrée. Nous avons obtenu un taux d'erreurs de 6,88 %, avec 200 règles de classification (200 nœuds dans la couche 2). Dans (fig. 12) ci-dessous sont présentées quelques règles de détection inférées par le classifieur neuro-flou.

```
#Règle 78:
SI (AR) indifférent ET (FRH) grand ET (BRH) indifférent ET (FRW) petit ET
(BRW) petit ET (SW) indifférent ET (FWR) indifférent ET (BWR) indifférent ET
(FD) petit ET (BD) petit ET (JPN) petit ALORS CHIFFRE
#Règle 194:
SI (AR) indifférent ET (FRH) grand ET (BRH) grand ET (FRW) grand ET
(BRW) grand ET (SW) indifférent ET (FWR) petit ET (BWR) indifférent ET
(FD) grand ET (BD) petit ET (JPN) grand ALORS DELIMITEUR ARRIERE
#Règle 158:
SI (AR) indifférent ET (FRH) petit ET (BRH) indifférent ET (FRW) grand ET
(BRW) indifférent ET (SW) indifférent ET (FWR) indifférent ET
(BWR) indifférent ET (FD) petit ET (BD) petit ET (JPN) grand
ALORS DELIMITEUR AVANT
```

Fig. 12 Quelques règles inférées par le classifieur.

6. CONCLUSION

Nous avons proposé dans ce travail un prototype d'un système de lecture automatique de montants numériques de chèques arabes Saoudiens. Cet article s'articulait essentiellement autour de deux parties. Une partie de reconnaissance de

chiffres et une partie de détection de délimiteurs. Nous avons présenté dans la première un nouveau jeu de primitives basé sur les caractéristiques morphologiques et locales. Les résultats sont très satisfaisants au vu de la taille de la base de donnée utilisée. Dans la deuxième partie nous avons proposé une méthode de détection des symboles de délimitation basée sur la classification neuro-floue. Pour ce faire, nous avons considéré un ensemble de primitives géométriques et topologiques propres aux chaînes de chiffres et dont la pertinence a été montré par les résultats exposés. L'ultime objet étant d'avoir des chaînes de chiffres correctement reconstituées, les résultats des deux parties doivent être combinés correctement. L'agrégation des deux modules est en cours de réalisation et fera l'objet d'une publication future.

Références

- [1] S.Madhvanath, V.govindaraju, "Using Holistic Features in Handwritten Word Recognition", United States Postal Service (USPS), (1992) vol.1, pp. 183-198.
- [2] D.Guillevic, "Unconstrained handwriting recognition applied to the processing of bank cheques", Doctoral Thesis (1995), Concordia University, Montreal.
- [3] M.Cheriet, C.Y.Suen. "Extraction of key letters for cursive script recognition". Pattern Recognition Letters 14 (1993), pp. 1009-1017.
- [4] N.W.Strathy . Master Thesis (1993), Concordia University, Montreal.
- [5] Y.C. Chim, A.A. Kassim, Y.Ibrahim, "Dual Classifier System for Handprinted Alphanumeric Character Recognition", Pattern Analysis & Applications (1998) 1 :pp. 155-162.
- [6] Il-Seok Oh, Ching Y.Suen, "Distance features for neural network-based recognition of handwritten characters", IJDAR 1998, pp. 73-88.
- [7] S.Loncaric, "A Survey of Shape Analysis Techniques", Pattern Recognition (1998), Vol.31, No. 8, pp. 983-1001.
- [8] L.Heutte, T.Paquet, J.V. Moreau, Y.Lecourtier, C.Olivier, "A structural/statistical feature based vector for handwritten character recognition", Pattern Recognition Letters 19 (1998), pp. 629-641.
- [9] D.Nauck, R.Kruse, "Generating Classification Rules with the Neuro-Fuzzy System NEFCLASS", NAFIPS'96 , (1996) USA,Berkeley.
- [10] J.Roger Jang, "Neuro-Fuzzy Modeling : Architectures, Analyses and Applications", Doctoral Thesis(1992), University of California, Berkeley.
- [11] M.Cheriet, Y.S. Huang, C.Y.Suen, "Background Region-Based Algorithm for The segmentation of Connected Digits". In Proc. 11th International Conference on pattern recognition (1992), vol.2, pp. 619-622.
- [12] F.Said, R.Yacoub, C.Y.Suen, "Recognition of English and Arabic Numerals using a Dynamic Number of Hidden Neurons". ICDAR 1999, pp. 237-240, Bangalore, INDIA.