



**Université Sultan Moulay Slimane
Faculté des Sciences et Techniques
Département informatique
Beni Mellal**



N° d'ordre : ...

THESE

Présentée à la Faculté des Sciences et Techniques,
pour obtenir le Grade de Docteur en informatique.

Reconnaissance des Formes par une Classe d'Invariants et Applications à la reconnaissance des documents Tifinaghes

Par

Rachid EL AYACHI

Encadrant: M. FAKIR

Co-encadrant: B. BOUIKHALENE

Soutenue le 2/7/2012, devant les membres de jury :

- | | |
|-----------------------|---|
| S. MELLIANI | : Professeur à la Faculté des Sciences et Techniques
- Beni Mellal |
| A. ZEROUAL | : Professeur à la Faculté des Sciences Semlalia
- Marrakech |
| M. HASSANI | : Professeur à la Faculté des Sciences Semlalia
- Marrakech |
| M. OUKESSOU | : Professeur à la Faculté des Sciences et Techniques
- Beni Mellal |
| B. BOUIKHALENE | : Professeur à la Faculté Polydisciplinaire
- Beni Mellal |
| M. FAKIR | : Professeur à la Faculté des Sciences et Techniques
- Beni Mellal |

2012

Remerciements

Je tiens tout d'abord à exprimer ma plus profonde estime et mes vifs remerciements à mon directeur de thèse le professeur Mohamed FAKIR pour sa disponibilité permanente, son soutien persévérant et ses conseils pertinents.

Aussi, je remercie sincèrement le professeur Belaid BOUIKHALENE pour son encadrement considérable, ses idées précieuses et son soutien constant.

Je remercie vivement tous les membres de mon jury de thèse qui ont pris de leurs temps pour lire et juger mon travail ainsi que pour leur déplacement le jour de la soutenance.

Egalement, je remercie infiniment les rapporteurs de ma thèse pour leurs corrections et leurs remarques appropriées vis-à-vis de mon document.

Mes chaleureux remerciements à mes parents, ma femme, mon frère, mes sœurs et les membres de la famille qui m'ont aussi apporté leur aide nécessaire au cours de mes études et qui m'ont supporté dans les moments de stress.

En fin, un grand hommage à tous mes collègues et amis du Lycée Qualifiant Tarik Ibn Ziad d'El ksiba, de la Faculté des Sciences et Techniques de Beni Mellal et de l'Equipe de Traitement de l'Information et de Télécommunications (TIT).

PUBLICATIONS ET COMMUNICATIONS

Publications :

1. R. El Ayachi, M. Fakir and B. Bouikhalene « Recognition of Tifinaghe Characters Using Dynamic Programming & Neural Network », "Document Recognition and Understanding", ISBN 978-953-307-995-0, pp: 35-52, INTECH, 2011.
2. R. El Ayachi, M. Fakir et B. Bouikhalene, « Recognition of TIFINAGHE Characters Using A multilayer Neural Network », International Journal Of Image Processing (IJIP), vol. 5, Issue2, pp: 109-118, 2011.
3. R. El Ayachi, K. Moro, M. Fakir et B. Bouikhalene, « On The Recognition Of Tifinaghe Scripts », Journal of Theoretical and Applied Information Technology (JATIT), vol. 20, No. 2, pp: 61-66, 2010.
4. R. El Ayachi, B. Bouikhalene et M. Fakir, «Transformation de Fourier et Moments Invariants Appliqués à la Reconnaissance des Caractères Tifinaghe», eTI, Revue électronique internationale en TI, ISSN 1114-8802, 2011. (Accepté pour publication).

Communications:

1. R. El Ayachi, M. Fakir et B. Bouikhalene, « Reconnaissance des caractères par l'utilisation des invariants », La Première Journée Nationale de Mathématiques et applications en Informatique et Télécommunication, Juillet 2011, Beni Mellal-Maroc.
2. R. El Ayachi, B. Bouikhalene et M. Fakir, « Transformation de Fourier et Moments Invariants Appliqués à la Reconnaissance des Caractères Tifinaghe », SITACAM'11, Agadir-Maroc, 06-07 Mai 2011. pp : 41-54.
3. R. El Ayachi et M. Fakir, « Recognition of Tifinaghe Characters Using Invariants Moments & Dynamic Programming », CMT 2010, Casablanca-Maroc.
4. R. El Ayachi, K. Moro, M. Fakir et B. Bouikhalene, « Off Line Tifinaghe Scripts Recognition Using Dynamic Programming», ISIVC 2010, Rabat-Maroc.
5. R. El Ayachi et M. Fakir, « Off Line Printed Amazigh Scripts Recognition », MICS 2010, Rabat-Maroc.
6. R. El Ayachi et M. Fakir, « Recognition of Tifinaghe Characters Using Neural Network », ICMCS 2009, Ouarzazat-Maroc.
7. R. El Ayachi et M. Fakir, « Reconnaissance de Caractères Tifinagh », TELECOM 2009, Agadir-Maroc.
8. R. El Ayachi et M. Fakir, « An Off Line Tifinagh Characters Recognition », PRIP 2009, Bularus, Russia.

TABLE DES MATIERES

INTRODUCTION GENERALE.....	1

CHAPITRE I : ETAT DE L'ART.....	4
1. Introduction.....	4
2. Etude bibliographique.....	4
3. Différents aspects de l'OCR.....	7
3.1 Reconnaissance en-ligne (on-line).....	8
3.2 Reconnaissance hors-ligne (off-line).....	8
4. Problèmes liés à l'OCR.....	9
5. Organisation générale d'un système de reconnaissance.....	10
5.1 Acquisition.....	11
5.2 Prétraitement.....	11
5.2.1 Normalisation.....	11
5.2.2 Réduction du bruit.....	11
5.2.3 Correction de l'inclinaison.....	11
5.2.4 Segmentation.....	12
5.3 Extraction des attributs.....	12
5.3.1 Analyse globale.....	13
5.3.2 Analyse structurelle.....	14
5.4 Reconnaissance.....	15
5.4.1 Méthodes de reconnaissance.....	15
5.4.2 Approches de reconnaissance.....	15
6. Mesures de performance.....	16
7. Conclusion.....	17

CHAPITRE II : OUTILS MATHEMATIQUES POUR L'EXTRACTION DES ATTRIBUTS.....	18
1. Introduction.....	18
2. Définition d'un groupe.....	18
3. Action d'un groupe sur un ensemble.....	18
3.1 Action transitive.....	19

3.2 Orbite.....	19
3.3 Stabilisateur.....	19
4. Représentations d'un groupe.....	19
5. Action d'un groupe sur un espace.....	19
6. Image sur X.....	20
7. Problème de reconnaissance des formes.....	20
8. Caractérisation des systèmes d'invariants.....	20
8.1 Analyse de Fourier et ses variantes.....	20
8.1.1 Transformation de Fourier.....	20
8.1.2 Transformation de Fourier classique.....	21
8.1.3 Descripteurs de Fourier génériques.....	21
8.1.4 Descripteurs de Fourier généralisés.....	22
8.1.5 Descripteurs de Fourier-Mellin.....	22
8.2 Fonctions spéciales.....	23
8.2.1 Harmoniques sphériques.....	24
8.2.2 Moments invariants.....	26
8.2.3 Moments invariants et transformation de Fourier.....	28
8.2.4 Moments invariants modifiés.....	31
8.2.5 Transformation de Walsh.....	33
9. Conclusion.....	34

CHAPITRE III : PRETRAITEMENT.....	35
1. Introduction.....	35
2. Normalisation.....	35
3. Filtrage.....	36
4. Correction de l'inclinaison.....	37
5. Segmentation.....	39
5.1 Segmentation en lignes.....	40
5.2 Segmentation en caractères.....	40
6. Conclusion.....	41

CHAPITRE IV : METHODES DE CLASSIFICATION.....	42
1. Introduction.....	42

2. Notion de structure.....	42
3. Notion de similarité.....	42
4. Mesure de distance.....	43
5. Programmation dynamique.....	44
6. Réseaux de neurones.....	46
6.1 Introduction.....	46
6.2 Neurones biologiques.....	47
6.3 Neurone artificiel.....	48
6.4 Fonctions d'activation.....	49
6.5 Structure d'un réseau de neurones.....	50
6.5.1 Réseau multicouche.....	50
6.5.2 Réseau à connexions locales.....	51
6.5.3 Réseau à connexions récurrentes.....	52
6.5.4 Réseau à connexions complexes.....	52
6.6 Apprentissage.....	53
6.6.1 Apprentissage supervisé.....	53
6.6.2 Apprentissage non supervisé.....	53
6.6.3 Exemple d'utilisation d'un réseau multicouche.....	53
6.7 Mémoires associatives.....	56
6.7.1 Fonctionnement de la mémoire associative.....	57
6.7.2 Types de mémoires associatives.....	58
6.7.3 Exemple d'utilisation d'une mémoire hétéro-associative.....	60
7. Conclusion.....	61

CHAPITRE V : RESULTATS EXPERIMENTAUX.....	62
1. Introduction.....	62
2. Prétraitement.....	63
2.1 Normalisation.....	63
2.2 Correction d'inclinaison.....	63
2.3 Segmentation.....	64
2.3.1 Segmentation en lignes.....	64
2.3.2 Segmentation en caractères.....	66
3. Extraction d'attributs.....	67

3.1 Transformation de Walsh.....	67
3.2 Moments invariants.....	68
3.3 Moments invariants modifiés.....	68
3.4 Méthode d’histogramme.....	70
4. Classification.....	74
4.1 Traitement 1.....	76
4.2 Traitement 2.....	77
4.3 Traitement 3.....	77
4.4 Résultats et remarques.....	78
5. Conclusion.....	79

CONCLUSION GENERALE.....	88
BIBLIOGRAPHIE.....	90

ANNEXE : CARACTERISTIQUES DES CARACTERES TIFINAGHE.....	100

LISTE DES FIGURES

Fig.1	Ecriture en-ligne et hors-ligne.....	8
Fig.2	Exemples de communication Homme-machine.....	9
Fig.3	Système de reconnaissance des documents.....	10
Fig.4	Angle d'inclinaison.....	12
Fig.5	Analyse globale.....	13
Fig.6	Analyse structurale.....	14
Fig.7	Système des coordonnées sphériques.....	24
Fig.8	Paramètres : Détermination des marges.....	35
Fig.9	(a) Avant normalisation.....	36
	(b) Après normalisation.....	36
Fig.10	Angle d'inclinaison.....	38
Fig.11	(a) Avant la correction d'inclinaison.....	38
	(b) Histogramme horizontal.....	38
	(c) Après la correction d'inclinaison.....	38
Fig.12	Détection des lignes et caractères.....	39
Fig.13	Segmentation en lignes.....	40
Fig.14	Segmentation en caractères.....	40
Fig.15	Codage de Freeman.....	42
Fig.16	(a) Principe de recherche.....	45
	(b) Principe de la PD.....	45
Fig.17	Neurone biologique.....	48
Fig.18	Neurone artificiel.....	49
Fig.19	Réseau multicouche.....	51
Fig.20	Réseau à connexions locales.....	51
Fig.21	Réseau à connexions récurrentes.....	52
Fig.22	Réseau à connexions complexes.....	52
Fig.23	Réseau multicouche avec une couche cachée.....	53
Fig.24	Principe d'apprentissage.....	54
Fig.25	Les étapes de la mémorisation.....	57
Fig.26	Exemple de mémoire associative.....	58
Fig.27	Hétéro-associative.....	59

Fig.28	Auto-associative.....	59
Fig.29	Mémoire hétéro-associative.....	60
Fig.30	Système de reconnaissance.....	62
Fig.31	(a) Avant normalisation.....	65
	(b) Histogramme horizontal.....	65
	(c) Après normalisation.....	65
	(d) Histogramme vertical.....	65
Fig.32	(a) Avant correction.....	65
	(b) Histogramme horizontal de (a).....	65
	(c) Résultat de correction.....	65
	(d) Histogramme après correction.....	65
Fig.33	(a) Segmentation en lignes.....	66
	(b) Histogramme horizontal.....	66
Fig.34	Segmentation en caractères.....	66
Fig.35	(a) Calcul de la distance d	67
	(b) Colonnes de segmentation.....	67
Fig.36	Contours externes utilisés pour calculer les moments invariants modifiés....	70
Fig.37	Principe utilisé pour suivre le chemin du contour.....	70
Fig.38	Organigramme de calcul.....	71
Fig.39	Organigramme de la méthode d'histogramme.....	72
Fig.40	Premier traitement.....	76
Fig.41	Deuxième traitement.....	77
Fig.42	(a) Calcul de L1.....	78
	(b) Calcul de L2.....	78
Fig.43	Exemple de texte en Tifinaghe.....	103
Fig.44	Plan Unicode associe au Tifinaghe	104

LISTE DES TABLEAUX

Tab.1	Travaux concernant la reconnaissance des caractères Tifinaghe.....	7
Tab.2	Fonctions d'activation.....	50
Tab.3	Transformation de Walsh.....	67
Tab.4	Moments invariants.....	68
Tab.5	Moments invariants modifiés	68
Tab.6	Contours des caractères Tifinaghe	69
Tab.7	Contours externes.....	69
Tab.8	Propriétés des contours des caractères Tifinaghe.....	73
Tab.9	Tableau résultant.....	75
Tab.10	Caractères identiques.....	75
Tab.11	Traitements 1, 2 et 3.....	76
Tab.12	Variables SUP et INF.....	77
Tab.13	Variable TA.....	78
Tab.14	Taux de reconnaissance et taux d'erreur.....	78
Tab.15	Taux de reconnaissance et taux d'erreur dans le cas d'ajout des traitements (1, 2 et 3).....	79
Tab.16	Base de référence.....	80
Tab.17	Base de teste.....	83
Tab.18	Matrice de confusion (Moments invariants et R.N).....	84
Tab.19	Matrice de confusion (Moments invariants, Traitements [1, 2 et 3] et R.N)...	85
Tab.20	Matrice de confusion (Transformation de Walsh et R.N).....	86
Tab.21	Matrice de confusion (Transformation de Walsh, Traitements [1, 2 et 3] et R.N).....	87
Tab.22	Alphabet de base Tifinaghe – IRCAM.....	102

La reconnaissance des formes est un champ en plein expansion dans le domaine de vision par ordinateur depuis les années 60. Elle a connu un grand essor en particulier la reconnaissance des documents.

Depuis les années 90, la reconnaissance des documents Latins, Arabes et Chinois a attiré l'attention des chercheurs et plusieurs applications montrent clairement le rôle primordial de la reconnaissance d'un texte en l'occurrence le tri automatique du courrier, le traitement automatique de dossiers administratifs, ou encore l'enregistrement des chèques bancaires.

Par ailleurs, les documents Tifinaghes sont peu traités, ce qui nécessite beaucoup d'importance et plusieurs efforts, afin d'augmenter la part de la reconnaissance de ces derniers au niveau des recherches scientifiques.

La notion de document est très générale et il existe une variante de définitions. Une définition appropriée d'un document est la suivante : un document est le support physique pour conserver et transmettre de l'information. Selon le support choisi, un document peut être textuel, graphique ou multimédia (sonore, vidéo). Les documents textuels se divisent en deux catégories : les documents imprimés et les documents manuscrits. Parmi les documents imprimés on distingue les documents à structures simples (contiennent des textes uniquement) et les documents à structures complexes (contiennent des textes et des images). Dans cette thèse on s'intéresse aux documents à structures simples écrits en Tifinaghe, et comportent des textes formés de caractères Tifinaghes. Donc, la reconnaissance de ce type de document se base sur la reconnaissance des caractères Tifinaghes formant ce document.

Tifinaghe est l'ensemble d'alphabets utilisés par la population Amazighe. Au maroc, il existe trois dialectes : Tarifite, Tamazighte et Tachelhite. La création de l'Institut Royal de la Culture Amazighe (IRCAM) par sa majesté le roi Mohammed VI a permis de choisir l'alphabet Tifinaghe en trente trois caractères (Annexe : Tableau 22).

Un système de reconnaissance des caractères est un outil qui permet aux ordinateurs d'identifier les caractères sans intervention humaine. En général, il comporte quatre phases :

- Acquisition
 - Prétraitement
 - Extraction d'attributs
 - Classification
- L'acquisition a pour objectif d'obtenir une image numérisée en utilisant des capteurs physiques, tel que un scanner, un appareil photo ou une table à digitaliser selon le type de reconnaissance.

- Le prétraitement est une phase importante qui comporte en général les fonctionnalités suivantes:
 - Binarisation : qui transforme une image de niveaux de gris en une image composée de deux valeurs 0 (Noir) et 1 (Blanc).
 - Normalisation : a pour but de supprimer les zones inutiles de l'image, afin de réduire le temps d'exécution.
 - Réduction du bruit : consiste à éliminer les erreurs aléatoires dans les valeurs de pixels en utilisant le filtrage (le filtre médian par exemple).
 - Correction d'inclinaison : consiste tout d'abord à détecter l'angle d'inclinaison puis la rotation du document dans le sens adéquat.
 - Segmentation : a pour rôle de segmenter le texte en lignes, puis, les lignes en caractères.
- L'extraction d'attributs s'intéresse au dégagement de quelques paramètres à partir des caractères segmentés, afin de les utiliser dans la reconnaissance. Cette étape doit prendre en considération la représentation du caractère, des descripteurs qui doivent être invariants par rapport à la translation, la rotation et le changement d'échelle. Cette invariance peut être réalisée à l'aide des descripteurs invariants à base d'analyse harmonique sur un espace localement compact (les descripteurs de Fourier, les descripteurs associés aux fonctions spéciales: les moments invariants, la transformation de Walsh, les moments de Zernike et les harmoniques sphériques).
- La classification est la dernière phase dans un système de reconnaissance. Elle démontre la robustesse du système car l'efficacité repose sur le résultat de la décision. C'est pour cela que le système doit faire recourt aux approches de reconnaissance les plus célèbres et connues dans ce domaine. Parmi ces approches, on peut citer : la programmation dynamique et les réseaux de neurones.

Dans le cas des caractères Tifinaghes, l'utilisation des descripteurs invariants dans l'extraction d'attributs peut engendrer des problèmes lors de la reconnaissance, qui sont liés à l'existence de quelques caractères Tifinaghes qui se ressemblent par rotation ou par changement d'échelle. Pour remédier à ce problème, nous avons proposé une méthode basée sur le calcul des histogrammes des contours des caractères.

D'une façon générale, on distingue deux types d'application pour la reconnaissance des caractères: en-ligne (on-line) et hors ligne (off-line).

Ce travail aborde le cadre de la reconnaissance hors ligne de documents Tifinaghes imprimés. L'objectif est de développer des systèmes de reconnaissance qui ont un taux de reconnaissance élevé et, par conséquence, un taux d'erreur faible.

Le plan de cette thèse est constitué de cinq chapitres organisés comme suit:

- Le premier chapitre est consacré à l'état d'art sur la Reconnaissance Optique des Caractères (ROC) ; les différents types de méthodes de reconnaissance, ainsi que les étapes nécessaires à la réalisation d'un système de reconnaissance.
- Dans le deuxième chapitre, nous listons quelques grandes familles de primitives couramment utilisées à titre d'exemple les moments invariants, les descripteurs de Fourier, les descripteurs de Fourier-Mellin et les fonctions spéciales.
- Les étapes du prétraitement sont détaillées dans le troisième chapitre. Ces étapes sont : la normalisation, le filtrage, la correction d'inclinaison et la segmentation.
- Le quatrième chapitre présente les différentes approches utilisées dans la phase de classification à savoir la programmation dynamique et les réseaux de neurones.
- Pour conclure, les points importants du document, notamment la partie expérimentale et les résultats obtenus sont cités dans le cinquième chapitre. Nous évoquons les perspectives et les voies de recherche qui semblent prometteuses pour aboutir à un système de reconnaissance fiable.

1- Introduction

La Reconnaissance Optique de Caractères (ROC) fait l'objet de l'avenir de la communication homme-machine. Elle est utilisée dans plusieurs domaines où le texte est la base de travail, principalement en bureautique, pour des buts d'indexation et d'archivage automatique de documents, en Publication Assistée par Ordinateur (PAO) pour faciliter la composition à partir d'une sélection de plusieurs documents, dans la poste pour le tri automatique des courriers, dans les banques pour faciliter la lecture des montants des chèques, ...etc.

La reconnaissance de l'écriture relève du domaine de la reconnaissance des formes qui s'intéresse aux formes de caractères. Le but est d'attribuer à une forme un identifiant des prototypes de référence déterminés préalablement. La reconnaissance des caractères Tifinaghes a débuté à mes connaissances en 1988 [1]. Dans ce chapitre, on cite quelques références concernant la ROC, et éventuellement ceux effectués sur les caractères Tifinaghe et les documents Tifinaghes.

2- Etude bibliographique

La recherche dans le domaine de la reconnaissance des caractères a connu un très vaste essor au cours des dernières décennies. Elle s'est intéressée aux domaines des caractères Latins [2, 3, 4, 5, 6], Japonais, Chinois et Arabe [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]. Cependant le problème lié à la reconnaissance des caractères Tifinaghes est presque non abordable.

Il existe peu d'articles qui sont publiés dans le domaine de reconnaissance des caractères Tifinaghes. Parmi ces travaux nous citons :

- Djematen et al [18] ont donné une méthode statistique de reconnaissance de caractères berbères manuscrits basée sur la position des points caractéristiques dans le rectangle-enveloppe de l'image du caractère.
- Es Saady et al [19] ont traité dans leur papier les caractères Amazighes imprimés isolés en utilisant un système automatique de reconnaissance de caractères basé sur le formalisme des automates à états finis.
- Fakir et al [20] ont évalué quinze algorithmes de squelettisation utilisant l'approche parallèle. Aussi, ils ont fait une comparaison entre ces algorithmes selon plusieurs critères ; le nombre d'itérations, le temps d'exécution, le taux de reconnaissance et le respect des propriétés du squelette.

- Bouikhalene et al [21] ont proposé une méthode pour la reconnaissance automatique des caractères Tifinaghe. Après acquisition de l'image, ils ont procédé à un seuillage, puis à la localisation des cavités des caractères, ce qui permet de caractériser la forme de l'image. Pour la classification ils ont utilisé les réseaux de neurones multicouches.

- Ait Ouguengay et Taalabi [22] ont proposé un réseau de neurones artificiels (RNA) pour la reconnaissance de caractères Amazighes. Le réseau de neurones utilisé est un perceptron multicouches à une seule couche cachée. Ce dernier a été entraîné sur une base de données qui contient des patterns de la graphie Amazighe de différentes fontes et tailles.

- El Ayachi et al [23] ont présenté un système de reconnaissance de l'écriture Tifinaghe basé sur les moments invariants et la transformée de Walsh utilisant la programmation dynamique. Le système proposé contient trois parties principales: le prétraitement, l'extraction de caractéristiques et la reconnaissance. Les moments invariants et la transformation de Walsh sont utilisés pour l'extraction d'attributs. Dans la phase de reconnaissance, ils ont adopté la programmation dynamique.

- Amrouch et al [24] ont présenté un système automatique de reconnaissance de caractères Amazighes basé sur le Modèle de Markov caché (HMM).

- El Ayachi et al [25] ont développé une méthode hors-ligne où les moments invariants sont calculés à l'aide de la transformation de Fourier avec un taux de reconnaissance de 93.27%.

- Moro et al [26] ont réalisé un système de reconnaissance des caractères basé sur les réseaux de neurones. Dans le prétraitement, ils ont appliqué l'égalisation d'histogramme, la correction d'inclinaison et la squelettisation. La classification a été réalisée en se basant sur la distance euclidienne. Dans la phase d'extraction d'attributs, ils ont adopté une méthode qui consiste à faire la somme des valeurs des pixels au niveau horizontal, vertical et des deux diagonales. Un taux de reconnaissance de 92.52% a été obtenu.

- El Ayachi et al [27] ont proposé un système de reconnaissance des caractères Tifinaghe utilisant les réseaux de neurones multicouches. Ils ont adopté le filtre Médiane pour réduire le bruit ainsi que la transformation de Hough pour corriger l'inclinaison. Dans la phase d'extraction des paramètres, la transformation de Walsh est utilisée. Mais la méthode présente une faiblesse due à la rotation des caractères.

- Bencharef et al [28] ont utilisé les descripteurs métriques basés sur le calcul de la métrique riemannienne. Ces descripteurs sont connus par leur fiabilité vis-à-vis du changement d'échelle, de l'existence de bruit et des déformations géométriques. Pour la

classification des caractères Tifinaghes, les séparateurs à vaste marge et les réseaux de neurones ont été employés. Un taux de reconnaissance de 98% a été obtenu pour une base d'apprentissage constituée de 15 échantillons pour 10 caractères.

- El kessab et al [29] ont développé un système de reconnaissance des caractères Tifinaghe manuscrits, basé sur l'utilisation des réseaux de neurones et le modèle de Markov caché. La morphologie mathématique est adoptée dans la phase d'extraction. Un taux de reconnaissance de 93.80% a été obtenu pour une base de test de 1275 images.

- Fakir et al [30] ont effectué une comparaison entre deux algorithmes : les cartes auto-organisatrices de Kohonen et les k-plus proche voisins flous. La technique du codage rétinien est employée dans l'extraction d'attributs. Les différents tests ont été appliqués sur les caractères Tifinaghes.

- Bazdouz et al [31] ont utilisé les MMC pour la reconnaissance des caractères Tifinaghe isolés. La taille de la base de données est de 42 caractères. Un taux de reconnaissance de 87% a été obtenu.

- Es-Saady et al [32] ont proposé un système de reconnaissance des caractères Amazigh. Une méthode basée sur les axes horizontal et vertical est utilisée pour le calcul des attributs. Les réseaux de neurones sont utilisés dans la phase de classification. Un taux de reconnaissance de 96.32% a été obtenu pour une base de test qui contient 20150 images.

Nous pouvons résumer les travaux réalisés dans ce domaine de reconnaissance des caractères berbères au tableau ci-dessous :

Référence et auteurs	Méthode d'extraction	Méthode de classification	Type de caractères	Base de données	Taux de reconnaissance
Essady [19]	Chaine de Freeman	Automate	Imprimé	322	94.7%
Fakir [20]	M. Invariants	thinning algorithms	Imprimé	-	90%
Bouikhalene [21]	Cavité	R.N	Imprimé	-	-
El ayachi [23]	T. Walsh	R.N	Imprimé	-	90.63%
Amrouch [24]	Chaine de Freeman	HMM	Imprimé	-	-
El ayachi [25]	M. Invariants	P.D	Imprimé	627	93.27

Moro [26]	Sommation des valeurs des pixels au niveau horizontal, vertical et des deux diagonales	R.N	Imprimé	2100	92.52%
Bencharef [28]	Descripteurs métrique	SVM	Imprimé + Manuscrit	150	98%
Badre [29]	Détection des zones caractéristiques	R.N	Manuscrit	1275	93.80%
Fakir [30]	Codage rétinien	Distance Euclidienne	Imprimé	200	-
Bazdouz [31]	T. Hough	M.M.C	Imprimé	42	87%
Es-saady [32]	Axes horizontal et vertical	R.N	Imprimé + Manuscrit	20150	96.32%
Bencharef [33]	Geodesic Descriptors	R.N	Imprimé	2175	99%
Bencharef [34]	Riemannian geometry	R.N	Imprimé + Manuscrit	2505	98%

Tableau 1 : Travaux concernant la reconnaissance des caractères Tifinaghe

3- Différents aspects de l'ROC

A mes connaissances il n'existe pas de système universel de ROC qui permet de reconnaître n'importe quel caractère dans n'importe quelle fonte. Tout dépend du type des données traitées et bien évidemment de l'application visée. Il existe deux modes de classification des systèmes ROC : Les systèmes qualifiés de « en-ligne » et les systèmes qualifiés « hors-ligne » (Fig.1) suivant le mode d'acquisition [2,8].

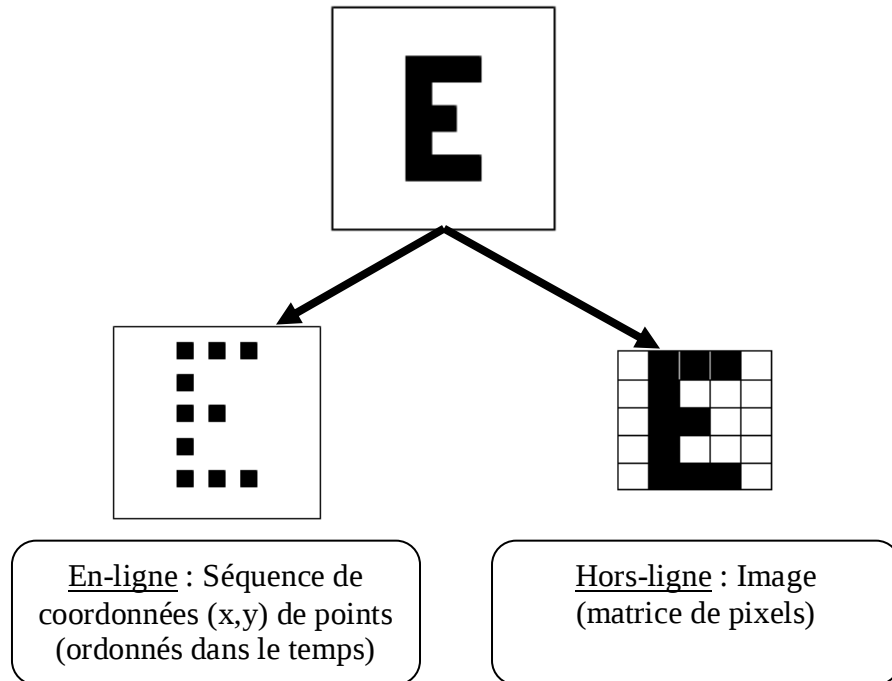


Fig.1 : Ecriture en-ligne et hors-ligne

3-1- Reconnaissance en-ligne (on-line)

La reconnaissance en-ligne s'opère en temps réel (pendant l'écriture), il s'agit de reconnaître l'écriture au fur et à mesure de son tracé. Ce mode est réservé généralement à l'écriture manuscrite. L'acquisition de l'écrit est généralement assurée par une tablette graphique munie d'un stylo électronique (Fig.2).

La reconnaissance en-ligne présente un avantage majeur ; c'est la possibilité de correction et de modification de l'écriture de manière interactive vue la réponse en continu du système [2].

3-2- Reconnaissance hors-ligne (off-line)

Pour la reconnaissance hors-ligne, il s'agit de reconnaître les documents imprimés et les manuscrits écrits au préalable. L'image du document peut être numérisée à l'aide d'un scanner (Fig.2).

La reconnaissance hors-ligne peut être classée en plusieurs types :

- **Reconnaissance de texte :**

Il s'agit de reconnaître un texte de structure limitée à quelques lignes ou mots. La recherche consiste en un simple repérage des mots dans les lignes, puis à un découpage de chaque mot en caractères [35].

- **Reconnaissance de documents (analyse de documents) :**

Dans ce cas (analyse de document), il s'agit de données bien structurées dont la lecture nécessite la connaissance de la typographie et de la mise en page du document. Ici, la démarche n'est plus un simple prétraitement, mais une démarche experte d'analyse de document ; il y'a localisation des régions, séparation des régions graphiques et photographiques, étiquetage sémantique des zones textuelles à partir de modèles et détermination de la structure du document [8].

- **Reconnaissance de l'imprimé :**

Les caractères imprimés sont dans le cas général alignés horizontalement et séparés verticalement, ce qui simplifie la phase de lecture [35].

- **Reconnaissance du manuscrit :**

Dans le cas du manuscrit, les caractères sont souvent ligaturés et leur graphisme est inégalement proportionné à cause de la variabilité intra et inter-scripteurs. Ce cas nécessite généralement l'emploi de techniques de délimitation spécifiques et souvent des connaissances contextuelles pour guider la lecture [36].

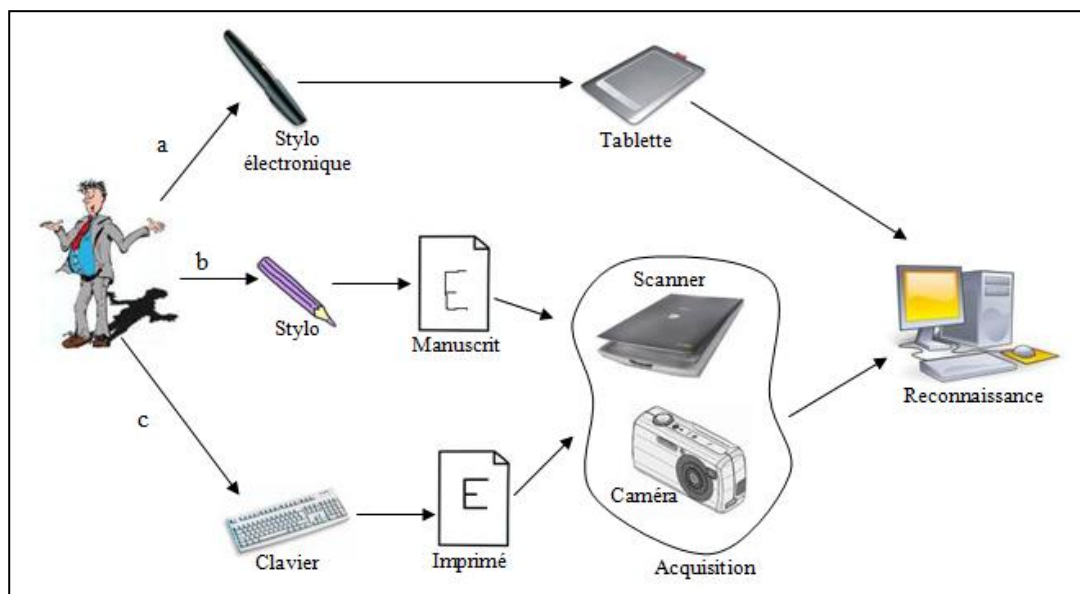


Fig.2 : Exemples de communication Homme-machine

(a : en-ligne, b et c : hors-ligne)

4- Problèmes liés à la ROC

La reconnaissance des caractères n'est pas une tâche facile; il existe plusieurs problèmes qui peuvent intervenir et qui rendent le rôle de la ROC plus compliqué. Parmi ces problèmes on trouve :

- La qualité du document : un document photocopie plusieurs fois est plus difficile à traiter que la copie originale. L'écriture peut devenir dégradée avec des parties du texte qui manquent.
- L'impression : le type d'imprimante utilisée peut influencer sur la qualité. Une imprimante à jet d'encre peut introduire des tâches d'encre et un étalement des caractères, une imprimante laser peut générer des lignes ou des fonds...
- L'acquisition : les techniques d'acquisition adoptées peuvent engendrer des défauts (inclinaison, bruit).
- La séparation : Certains documents contiennent des textes et des graphiques, la ROC doit être capable de les localiser.
- La correction d'inclinaison : pour un texte incliné, il faut détecter l'angle d'inclinaison et la corriger.
- Le chevauchement : il est difficile de segmenter les caractères qui se chevauchent entre eux.

5- Organisation générale d'un système de reconnaissance

La reconnaissance des documents est considérée typiquement comme un problème de reconnaissance des formes [23, 27].

D'une façon générale, le système de reconnaissance des caractères englobe un ensemble d'étapes qui s'exécutent successivement (Fig.3).

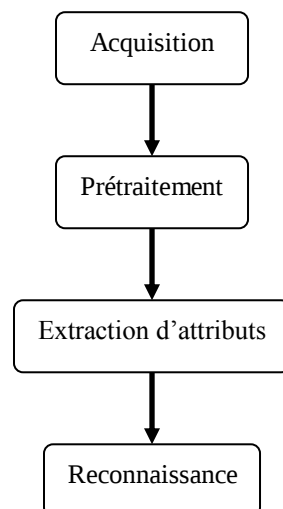


Fig.3 : *Système de reconnaissance des documents*

5-1- Acquisition

L'Acquisition est la première phase à appliquer dans un système de reconnaissance des caractères. Elle consiste à obtenir une image numérisée d'un texte, en utilisant des capteurs physiques et de la convertir en grandeurs numériques adaptées au système (Fig.3).

5-2- Prétraitement

Le prétraitement est l'une des phases d'un système de reconnaissance, il s'applique après l'acquisition, il englobe certaines fonctions à exécuter avant la phase de l'extraction des attributs, afin de produire une version nettoyée de l'image d'entrée.

Cette phase comporte :

- Normalisation de la position
- Réduction du bruit
- Correction de l'inclinaison de la ligne médiane
- Segmentation du document en lignes et en caractères.

5-2-1- Normalisation

La normalisation de la position est la première fonction du prétraitement, elle a pour but d'éliminer les zones indésirables dans l'image numérisée afin de minimiser l'espace utilisé de la mémoire et de réduire le temps de traitement [23].

5-2-2- Réduction du bruit

Le bruit est un parasite dégradant la qualité d'image. Par conséquent, il a une influence négative sur l'efficacité de la ROC. Donc, avant d'entamer les phases suivantes, il faut réduire le bruit afin d'améliorer la qualité d'image et augmenter le taux de succès de la ROC, et cela en utilisant le filtre médiane par exemple.

5-2-3- Correction de l'inclinaison

L'inclinaison est l'un des problèmes le plus fréquent dans un système de reconnaissance qui rend l'étape de segmentation très compliquée.

L'utilisation d'un scanner peut mener à un non-alignement de l'image du document.

L'angle d'inclinaison est l'angle (Fig.4) que font les lignes de textes de document avec la direction horizontale.

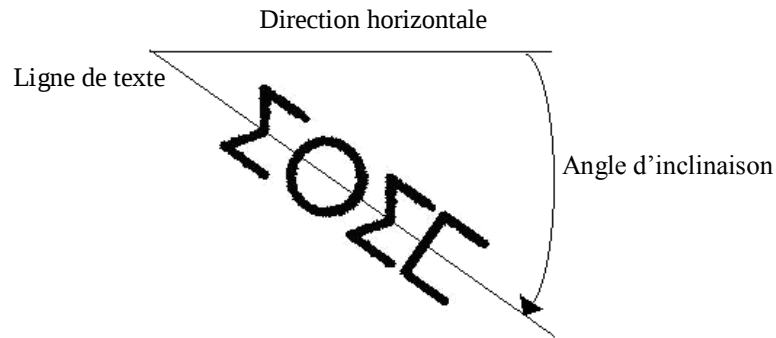


Fig.4 : Angle d'inclinaison

La correction de l'inclinaison rend le processus de segmentation facile. Au début, il faut déterminer l'angle d'inclinaison, après on fait une rotation dans le sens approprié selon l'angle détecté [12, 23, 37].

5-2-4- Segmentation

Dans cette étape, la segmentation d'un document repose sur deux principes : la segmentation en lignes et la segmentation en caractères.

La segmentation en lignes consiste à séparer les différentes lignes du texte, et cela peut être effectué par la méthode qui se base sur la décomposition de l'image en composantes connexes [7] ou bien la technique qui utilise l'histogramme des projections horizontales [27, 38].

La segmentation en caractères a pour objectif de détecter les caractères dans chaque ligne segmentée auparavant. On peut utiliser l'histogramme des projections verticales pour déterminer les espaces entre les caractères et pouvoir les séparer. Cependant cette technique peut ne pas être efficace dans certains cas où les caractères se chevauchent (le cas de l'écriture arabe). Donc, afin de résoudre ce problème, d'autres techniques peuvent être utilisées en se basant sur le contour, le squelette ou les composantes connexes [39].

5-3- Extraction des attributs

L'intérêt d'un système ROC consiste à identifier les caractères d'une image à l'aide de l'extraction des attributs.

Cette dernière vient après la phase de prétraitement, elle est l'étape la plus importante en ROC. La reconnaissance d'un caractère passe d'abord par l'analyse de sa forme et l'extraction de ses traits caractéristiques (primitives) qui seront exploités pour son identification.

La difficulté en reconnaissance des caractères manuscrits réside dans l'infinité de formes possibles écrites par différents scripteurs. Il est important de signaler que des échantillons de caractères représentant le même caractère écrits par la même main ne peuvent être identiques ni en forme ni en taille.

Cependant, les êtres humains peuvent reconnaître ces échantillons grâce à l'ajout d'autres primitives supplémentaires permettant l'identification des caractères. En conséquence, le transfert de l'ensemble d'informations des êtres humains à l'ordinateur permettent la reconnaissance automatique des caractères.

Les primitives peuvent être obtenues à partir de deux principaux types d'analyses [35, 38, 40]: globale et structurelle.

5-3-1- Analyse globale

Dans cette analyse (Fig.5), trois méthodes sont utilisées pour élaborer la matrice de caractères: Distribution des points, Transformations et Mesures physiques.

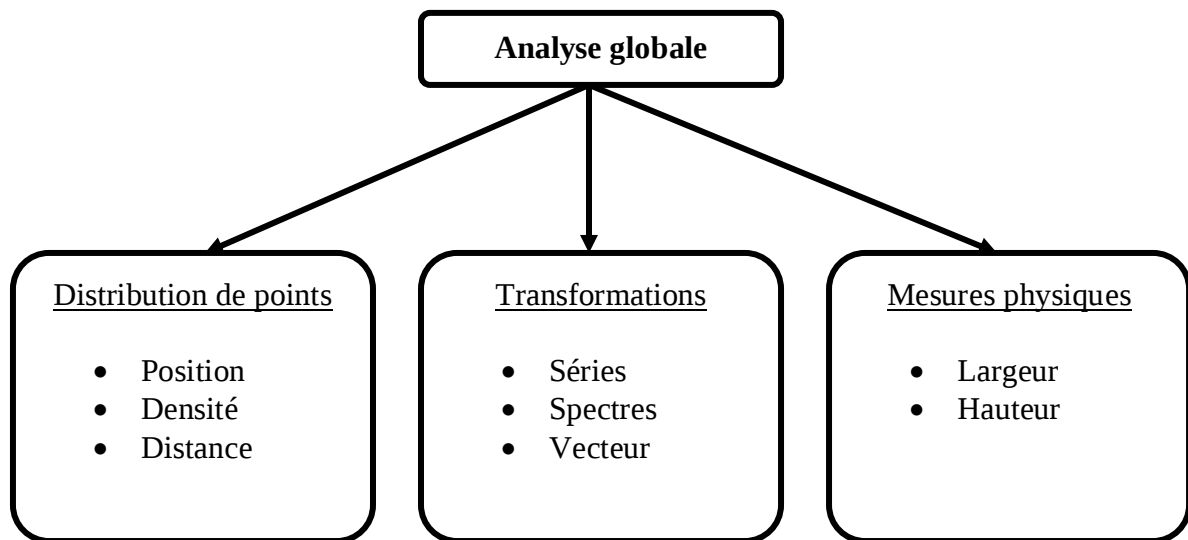


Fig. 5 : Analyse globale

- **Distributions de points :**

La distribution des points de la matrice permet d'évaluer la densité, les distances séparant certains points de référence et les informations de position.

- **Transformations :**

La transformation de la matrice de caractère en une série, un spectre ou un vecteur permet d'extraire d'autres caractéristiques. Pour réduire la capacité de mémoire utilisée et le temps d'exécution, d'autres transformations ont été proposées. Parmi ces transformations, on trouve : Fourier [25] et Walsh [27].

- **Mesures physiques :**

Le balayage horizontal et vertical de la matrice du caractère permet le calcul de la largeur et de la hauteur [41].

5-3-2- Analyse structurelle

Ce deuxième type d'analyse (Fig.6) adopte trois méthodes : segment de ligne et bords, étendue du caractère et ligne centrale du caractère.

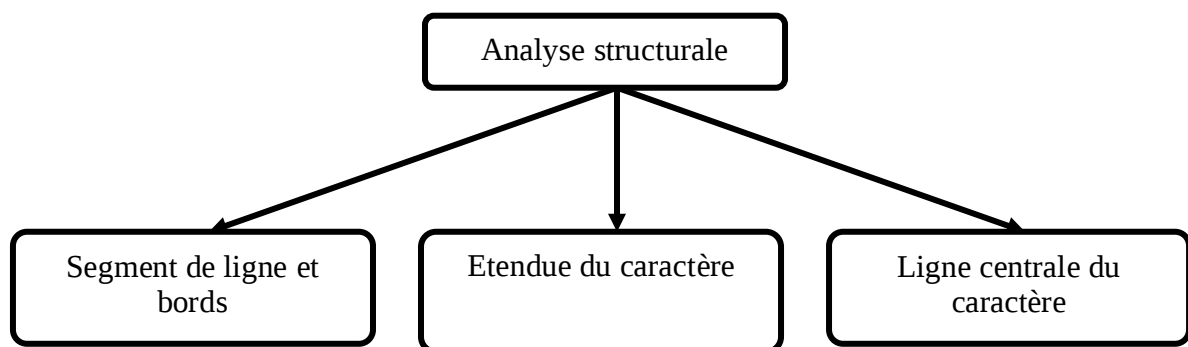


Fig. 6 : Analyse structurelle

- **Segment de ligne et bords :**

Les lignes peuvent être détectées à l'aide de plusieurs approches, parmi ces approches on peut citer: le déplacement d'une fenêtre à travers l'image toute entière ou bien par l'utilisation de la méthode d'histogramme.

Les bords peuvent être aussi détectés par examen des zones horizontale et verticale de l'image.

- **Etendue du caractère :**

L'étendue d'un caractère est obtenue par traçage du contour. Des informations relatives à la longueur, l'orientation et la position de tous les points du contour peuvent être obtenues. Les primitives suivantes peuvent être obtenues par traçage du contour [35] : (bouts de ligne, longueur du segment de ligne, inclinaison du segment de ligne)

- ***Ligne centrale du caractère :***

La ligne centrale du caractère est généralement obtenue par procédé de squelettisation. Ceci contient la majorité des informations nécessaires à la reconnaissance de caractères.

5-4- Reconnaissance

Dans un système ROC, la reconnaissance est la dernière étape à appliquer, elle utilise les caractéristiques dégagées lors de l'étape d'extraction d'attributs pour identifier le segment et l'attribuer à un modèle de référence [40].

5-4-1- Méthodes de reconnaissance

Deux méthodes s'opposent en reconnaissance des mots : globale et analytique.

- ***Méthode globale :***

A priori, la méthode globale de reconnaissance est séduisante car elle ne rencontre pas de difficultés provenant de la segmentation. Toutefois, elle ne peut se concevoir que dans le cas d'un vocabulaire limité. Cette dernière repose sur une caractérisation globale du mot.

- ***Méthode analytique :***

Contrairement à la méthode globale, la méthode analytique cherche à identifier les graphèmes issus de la segmentation (caractères) pour reconstituer les mots. Elle présente l'intérêt de pouvoir se généraliser à la reconnaissance d'un vocabulaire étendu.

5-4-2- Approches de reconnaissance

Les principales approches de reconnaissance sont: l'approche statistique, l'approche structurelle, l'approche stochastique et l'approche hybride.

- ***Approche statistique :***

Elle est basée sur l'étude statistique des mesures que l'on effectue sur les formes à reconnaître. Parmi les méthodes statistiques les plus utilisées, on trouve : les réseaux de neurones [9, 27, 42], approche bayésienne [43].

- ***Approche structurelle :***

Elle repose sur la structure physique des caractères. Chaque caractère est représenté par des paramètres liés à sa structure intrinsèque [44].

- **Approche stochastique :**

Elle utilise un modèle de reconnaissance qui prend en compte la grande variabilité de la forme. Les méthodes qui se basent sur l'approche stochastique sont robustes et fiables du fait de l'existence d'algorithmes d'apprentissage efficaces. Parmi les méthodes les plus répondues dans cette approche sont les méthodes utilisant les modèles de Markov cachés (HMM) [45].

- **Approche hybride :**

Pour améliorer les performances de reconnaissance, la tendance aujourd'hui est de construire des systèmes hybrides qui utilisent différents types de caractéristiques, et qui combinent plusieurs classifieurs en couches.

6- Mesures de performance

Le taux de reconnaissance correspond au rapport :

$$T_{reco} = \frac{N_{corr}}{N_{tot}} \quad (1)$$

Où :

T_{reco} : Taux de reconnaissance.

N_{corr} : Nombre de documents correctement reconnus.

N_{tot} : Nombre total de documents.

Lorsque le système de reconnaissance renvoie une mesure de confiance, il est possible d'établir un seuil de rejet, et donc d'évaluer le taux de reconnaissance en fonction de ce seuil :

$$T_{reco}(seuil) = \frac{N_{corr}(seuil)}{N_{tot}} \quad (2)$$

Où :

$T_{reco}(seuil)$: Taux de reconnaissance pour un seuil donné.

$N_{corr}(seuil)$: Nombre de documents correctement reconnus dont le score de confiance est supérieur à un seuil donné.

N_{tot} : Nombre total de documents.

- **Taux de bonne classification sans coûts :**

La première mesure à laquelle on va s'intéresser c'est le taux de bonne classification simplifiée (Tbc_s) ou sans coût. Il s'agit de l'indicateur le plus naturel et le plus évident permettant d'évaluer les performances d'un système de classification. Cette valeur, simple à

calculer, correspond au nombre d'éléments (caractères) correctement identifiés par le système. La définition du taux de bonne classification sans la prise en compte de rejet est :

$$Tbc_s = \frac{N_e}{N_t} \quad (3)$$

Où :

N_e : Nombre d'éléments correctement identifiés.

N_t : Nombre d'éléments total.

On obtient le taux d'erreur par :

$$Te_s = 1 - Tbc_s \quad (4)$$

Lorsque le rejet d'une forme est possible, un troisième taux dit « Taux de rejet » (Tr) est intégré. Il mesure le nombre d'éléments sur lesquels le système n'a pas pris de décision. On obtient ainsi :

$$Te_s = 1 - Tbc_s - Tr \quad (5)$$

De manière analogue, on peut également définir le taux de substitution de la façon suivante :

$$T_{sub}(seuil) = \frac{N_{corr}(seuil)}{N_{tot}} \quad (6)$$

Où :

$T_{sub}(seuil)$: Taux de substitution pour un seuil donné.

$N_{corr}(seuil)$: Nombre de documents mal reconnus dont le score de confiance est supérieur à un seuil donné.

N_{tot} : Nombre total de documents.

7- Conclusion

Dans ce chapitre, on a présenté une étude générale sur la reconnaissance des documents. En premier lieu, on a donné certains concepts généraux liés à la reconnaissance des caractères, ainsi que les principaux problèmes rencontrés par la ROC. Ensuite, nous avons abordé les différentes étapes intervenant dans la conception d'un système de reconnaissance des documents.

1- Introduction

L'extraction des attributs (des primitives) consiste à transformer une image (caractères, graphèmes,...) en un vecteur de primitives de taille fixe.

Cette transformation revient à changer l'espace de représentation de données, du plan de l'image vers un espace à N dimensions.

Le choix de primitives est critique, et influence nettement le résultat. Ces primitives doivent avoir deux propriétés :

- Être discriminantes : permettre une bonne différenciation entre les données de symboles à reconnaître.
- Maintenir un nombre de dimension limité, afin d'éviter le phénomène de malédiction de la dimensionnalité.

Des techniques comme l'Analyse en Composantes Principales (ACP) permettent de réduire le nombre de dimension.

Dans la suite, on présentera quelques grandes familles de primitives couramment utilisées :

- Les moments invariants [23, 25].
- Les descripteurs de Fourier [46].
- Les descripteurs de Fourier-Mellin [47].
- Les fonctions spéciales [48].

2- Définition d'un groupe

On appelle groupe un ensemble G muni d'une loi vérifiant les propriétés suivantes :

i) l'associativité :

$$\forall g, h, k \in G, \quad g(hk) = (gh)k \quad (7)$$

ii) l'existence d'un élément neutre noté e :

$$\forall g \in G \quad \text{on a} \quad ge = eg = g \quad (8)$$

iii) l'existence des éléments symétriques :

$$\forall g \in G, \exists g^{-1} \in G \quad \text{tel que} \quad gg^{-1} = g^{-1}g = e \quad (9)$$

3- Action d'un groupe sur un ensemble

Soient G un groupe et X un ensemble. On appelle action de G sur X une application :

$$\begin{aligned} G \times X &\rightarrow X \\ (g, x) &\rightarrow g.x \end{aligned} \quad (10)$$

tel que,

$$\text{i) } \forall x \in X, \quad \text{on a} \quad e.x = x \quad (11)$$

$$\text{ii) } \forall g, h \in G, \forall x \in X, \quad g.(h.x) = (gh).x \quad (12)$$

3-1- Action transitive

Si $\forall x, y \in X, \exists g \in G$, tel que $g.x = y$. On dit alors que l'action de G sur X est transitive.

3-2- Orbite

Soit G un groupe opérant sur un ensemble X . Pour tout $x \in X$, l'orbite de x , notée $G.x$, est la partie de X définie par

$$G.x = \{g.x / g \in G\} \quad (13)$$

3-3- Stabilisateur

Pour tout $x \in X$, le stabilisateur de x , noté H_x , est le sous-groupe de G défini par

$$H_x = \{g \in G / g.x = x\} \quad (14)$$

4- Représentations d'un groupe

Soient G un groupe et H un espace vectoriel de dimension finie et T une application de G dans $M(H)$ (ensemble de matrices sur H). On dit que (T, H) est une représentation de G dans H si les propriétés suivantes sont vérifiées :

$$\text{i) } T(e) = I \quad (\text{Opérateur identique sur } H) \quad (15)$$

$$\text{ii) } T(gh) = T(g) \circ T(h) \quad (16)$$

On note $\hat{G}$: l'ensemble des représentations unitaires de G .

5- Action d'un groupe sur un espace

Soit G un groupe et X un espace de points. On dit que G agit sur X s'il existe pour tout $g \in G$ une transformation sur Ω qui à tout point x de X on associe un point de X noté $\Omega(g).x$ tel que

$$\Omega(g_1 g_2) = \Omega(g_1) \circ \Omega(g_2). \quad (17)$$

Ω est appelée l'action de G sur X .

6- Image sur X

Une image sur X est une fonction réelle positive, définie sur X (niveau de gris au point considéré) à support borné, c'est-à-dire nulle en dehors d'un ensemble borné de X .

7- Problème de reconnaissance des formes

Soit X un espace de points sur lequel agit un groupe G via l'action Ω .

• Formulation du problème

Etant donnée une image f sur X , déterminer les grandeurs caractéristiques de f , invariantes sous l'action de G , c'est-à-dire : si f et h sont deux images sur X vérifiant:

$$\exists g \in G, \text{ tel que } \forall x \in X \text{ on a } f(x) = h(\Omega(g).x). \quad (18)$$

Le problème consiste à calculer des grandeurs égales pour f et h , invariantes sous l'action Ω et à déterminer si possible un système complet d'invariants (i.e. permettant de décider si deux images f et h sur X se déduisent l'une de l'autre via l'action de G sur X).

8- Caractérisation des systèmes d'invariants

Nous caractérisons les systèmes d'invariants par des méthodes d'analyse harmonique, à savoir :

- L'analyse de Fourier
- Les fonctions spéciales (les moments invariants, les harmoniques sphériques, la transformation de Walsh)

8-1- Analyse de Fourier et ses variantes

Soient f et h deux fonctions sur le groupe G , telles qu'il existe un élément $a \in G$ tel que $\forall g \in G, f(g) = h(a.g)$. Soit $\lambda \in \hat{G}$ ($\hat{G}$ est le dual de G).

8-1-1- Transformation de Fourier

Pour tout $\lambda \in \hat{G}$ on a :

$$\begin{aligned} \hat{f}(\lambda) &= \int_G f(g) T_\lambda(g^{-1}) P_g(g) dg \\ &= \int_G h(a.g) T_\lambda(g^{-1}) P_g(g) dg \\ &= \int_G h(a.g) T_\lambda^{-1}(a^{-1}.a.g) P_g(a.g) d(a.g) \\ &= \int_G h(g) T_\lambda(a^{-1}.g)^{-1} P_g(g) dg \end{aligned} \quad (19)$$

$$= \int_G h(g) T_\lambda(g^{-1}.a) P_g(g) dg$$

Et comme $T_\lambda(g^{-1}.a) = T_\lambda(g^{-1}).T_\lambda(a)$ (20)

Alors $\hat{f}(\lambda) = \hat{h}(\lambda).T_\lambda(a)$ (21)

L'ensemble $\left\{ \hat{f}(\lambda) = \hat{h}(\lambda).T_\lambda(a), \lambda \in \hat{G} \right\}$ définit un système d'invariants appelés les descripteurs de Fourier.

8-1-2- Transformation de Fourier classique

Pour $G = \mathbb{R}^2$, les descripteurs de Fourier s'obtiennent comme suit :

La transformée de Fourier usuelle s'applique aux fonctions de $L^2(\mathbb{R}^2)$. Sa restriction au cas bidimensionnel est définie à un facteur de normalisation près par

$$\begin{aligned} \hat{f}(u, v) &= \int_{\mathbb{R}^2} f(x, y).e^{-i(ux+vy)} dx dy \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y).e^{-i(ux+vy)} dx dy \end{aligned} \quad (22)$$

où u et v sont les coordonnées de l'image dans le domaine fréquentiel.

8-1-3- Descripteurs de Fourier génériques

Soit $\hat{f}$ la transformée de Fourier 2D discrète :

$$\hat{f}(u, v) = \sum_0^{M-1} \sum_0^{N-1} f(x, y) \exp \left[-2i\pi \left(\frac{ux}{M} - \frac{vy}{N} \right) \right] \quad (23)$$

où M et N sont respectivement la largeur et la hauteur d'une image f . Par un changement de variable (l'utilisation des coordonnées polaires : (ρ, ϕ) sont les coordonnées du spectre de Fourier et (k, θ_l) sont les coordonnées de l'image), on obtient (la transformée de Fourier 2D discrète de l'image passée en coordonnées polaire PF):

$$PF(\rho, \phi) = \sum_k \sum_l f(k, \theta_l). \exp \left[2i\pi \left(\frac{k}{R} \rho + \frac{2\pi l}{T} \phi \right) \right] \quad (24)$$

où $0 \leq k < R$ et $\theta_l = \frac{2l\pi}{T}$ avec $0 \leq l < T$; $0 \leq \rho < R$, $0 \leq \phi < T$ et R et T sont respectivement

la résolution radiale et la résolution angulaire dans le domaine des fréquences.

L'invariance en translation est obtenue en prenant le centre de gravité de l'objet comme origine de l'espace polaire et l'invariance en échelle et en rotation est vérifiée dans l'ensemble des descripteurs réels (Fourier Descriptors : *FD*):

$$FD(f) = \left\{ \frac{|PF(0,0)|}{A}, \frac{|PF(0,1)|}{|PF(0,0)|}, \dots, \frac{|PF(m,n)|}{|PF(0,0)|} \right\} \quad (25)$$

où A est la surface du cercle englobant l'objet d'intérêt et m et n sont respectivement le nombre maximum de fréquences radiales et le nombre de fréquences angulaires sélectionnées.

8-1-4- Descripteurs de Fourier généralisés

Les descripteurs de Fourier généralisés (Generalised Fourier Descriptors : *GFD*) sont construits à partir des actions de groupe et plus particulièrement du groupe des déplacements M_2 . Le groupe M_2 est constitué des translations et des rotations dans le plan. Deux types de descripteurs ont été développés :

- les invariants de type "densité spectrale", $I_1^r(f)$:

$$I_1^r(f) = \int_0^{2\pi} \left| \hat{f}(r, \theta) \right|^2 d\theta \quad (26)$$

- les invariants de type "décalage de phase" :

$$I^{\varepsilon_1, \varepsilon_2}(f) = \int_0^{2\pi} \hat{f}(R_\theta(\varepsilon_1 + \varepsilon_2)) \overline{\hat{f}(R_\theta(\varepsilon_1))} \overline{\hat{f}(R_\theta(\varepsilon_2))} d\theta \quad (27)$$

où f est l'image, $\hat{f}(r, \theta)$ est la transformée de Fourier exprimée en coordonnées polaires dans le plan fréquentiel ε_1 et ε_2 , R_θ une rotation d'angle θ et $\overline{\hat{f}}$ est le conjugué de $\hat{f}$.

En utilisant les invariants spectraux I_1^r , un vecteur de descripteur de Fourier généralisé se note :

$$GFD(f) = \left\{ I_1^0 = \hat{f}(0,0), \frac{I_1^1(f)}{I_1^0(f)}, \dots, \frac{I_1^m(f)}{I_1^0(f)} \right\} \quad (28)$$

où m est le nombre de coefficients calculés.

8-1-5- Les descripteurs de Fourier-Mellin

Soit $f(r, \theta)$ une fonction réelle et positive (la forme) définie en coordonnées polaires. La Transformée de Fourier-Mellin (TFM) M_f de cette fonction est définie par la relation suivante :

$$M_f(q, s) = \int_{r=0}^{\infty} \int_{\theta=0}^{2\pi} r^{s-1} \exp(-iq\theta) f(r, \theta) dr d\theta \quad (29)$$

où $q \in \mathbb{Z}$, $s = \sigma + iv \in \mathbb{C}$ et i désigne l'unité imaginaire. La TFM est donc la combinaison de la représentation par les coefficients de Fourier d'une fonction 2π -périodique (pour la variable angulaire θ) et de la transformée de Mellin (pour la variable radiale r).

Pour $\sigma = 0$, on trouve une définition plus restreinte de la TFM qui trouve son origine dans la théorie des groupes et l'analyse harmonique (la TFM avec $\sigma = 0$ est la TFM de l'équation 29 ce qu'est la transformée de Fourier à la transformée de Laplace d'un signal continu).

La fonction $f(r, \theta)$ peut être reconstruite à partir de la TFM, en utilisant les relations suivantes:

$$\begin{aligned} f(r, \theta) &= \frac{1}{2\pi i} \int_{\sigma - i\infty}^{\sigma + i\infty} \sum_{q \in \mathbb{Z}} r^{-s} \exp(iq\theta) M_f(q, s) ds \\ &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \sum_{q \in \mathbb{Z}} r^{-\sigma - iv} \exp(iq\theta) M_f(q, s = \sigma + iv) dv \end{aligned} \quad (30)$$

Cette relation définit la TFM inverse.

La construction de descripteurs de formes invariants par rotation et homothétie s'appuie sur l'application du théorème du retard [80].

Soit $g(r, \theta) = f(ar, \theta + \beta)$ une version modifiée en orientation et en échelle de $f(r, \theta)$, on a la relation suivante :

$$M_g(q, s) = \alpha^{-s} \exp(iq\beta) M_f(q, s) \quad (31)$$

En prenant le module des termes de cette équation, il est possible de déduire des descripteurs qui sont invariants aux changements de rotation de la forme mais pas aux changements d'échelle.

Pour obtenir un ensemble de primitives invariantes aux changements d'échelle et d'orientation, il est néanmoins possible d'utiliser l'ensemble de descripteurs suivants [49] :

$$I_f(q, s) = M_f(q, s) [M_f(0, \sigma)]_{\sigma}^{-s} [M_f(1, \sigma)]^{-q} |M_f(1, \sigma)|^q \quad (32)$$

Partant de ces invariants, si $g(r, \theta) = f(ar, \theta + \beta)$, il peut être montré facilement que $I_g(q, s) = I_f(q, s)$.

8-2- Fonctions spéciales

Les fonctions spéciales jouent un rôle important dans plusieurs domaines en l'occurrence la reconnaissance des formes, elles peuvent être utilisées pour représenter et modéliser plusieurs phénomènes. Elles représentent un ensemble de polynômes analytiques qui sont

apparus souvent comme solutions de certaines équations différentielles et intégrales. Comme exemples de fonctions spéciales nous citons :

8-2-1- Harmoniques sphériques

Une harmonique sphérique est la restriction d'un polynôme homogène à la sphère $S^2 = \{(x, y, z) \in \mathbb{R}^3 / x^2 + y^2 + z^2 = 1\}$.

Une forme géométrique dans l'espace à trois dimensions est représentée en coordonnées sphériques r, θ et φ où r est la distance à l'origine, $\theta \in [0, \pi]$ et $\varphi \in [0, 2\pi]$ par la figure ci-dessous :

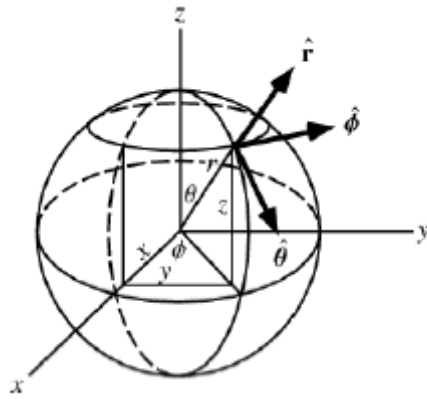


Fig. 7: Système des coordonnées sphériques

Les trois vecteurs principaux en coordonnées cartésiennes $(\vec{i}, \vec{j}, \vec{k})$ sont :

$$\vec{r} = \begin{bmatrix} \cos \varphi \sin \theta \\ \sin \varphi \sin \theta \\ \cos \theta \end{bmatrix}, \quad \vec{\theta} = \begin{bmatrix} \cos \varphi \cos \theta \\ \sin \varphi \cos \theta \\ -\sin \theta \end{bmatrix} \quad \text{et} \quad \vec{\varphi} = \begin{bmatrix} -\sin \varphi \\ \cos \varphi \\ 0 \end{bmatrix} \quad (33)$$

Les harmoniques sphériques sont la partie angulaire de la solution de l'équation de Laplace, $\nabla^2 f = 0$, dans le système de coordonnées sphériques.

L'équation de Laplace en coordonnées sphériques est définie sous la forme suivante:

$$\nabla^2 f = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial f}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial f}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 f}{\partial \varphi^2} = 0 \quad (34)$$

On en déduit qu'une harmonique sphérique est donnée par [50] :

$$Y_l^m(\theta, \varphi) = (-1)^m k(l, m) P_l^m(\cos \theta) e^{im\varphi} \quad (35)$$

Où

$$m \in \mathbb{Z}, |m| \leq l \in \mathbb{N}$$

et

P_l^m : le polynôme de Legendre associé de degré l et d'ordre m .

$$P_{lm}(x) = (-1)^m \frac{(1-x^2)^{m/2}}{2^l l!} \frac{d^{l+m}}{dx^{l+m}} (x^2 - 1)^l \quad (36)$$

et

$k(l, m)$: la fonction de normalisation donnée par

$$k(l, m) = (-1)^m \sqrt{\frac{2l+1(l-m)!}{4\pi(l+m)!}} \quad (37)$$

L'expression analytique des harmoniques sphériques jusqu'à l'ordre 3 peuvent être calculées par les formules suivantes:

$$\begin{aligned} Y_0^0(\theta, \varphi) &= \frac{1}{2} \sqrt{\frac{1}{\pi}} \\ Y_1^{-1}(\theta, \varphi) &= \frac{1}{2} \sqrt{\frac{3}{2\pi}} \sin \theta e^{-i\varphi} \\ Y_1^0(\theta, \varphi) &= \frac{1}{2} \sqrt{\frac{3}{\pi}} \cos \theta \\ Y_1^1(\theta, \varphi) &= \frac{-1}{2} \sqrt{\frac{3}{2\pi}} \sin \theta e^{i\varphi} \\ Y_2^{-2}(\theta, \varphi) &= \frac{1}{4} \sqrt{\frac{15}{2\pi}} \sin^2 \theta e^{-2i\varphi} \\ Y_2^{-1}(\theta, \varphi) &= \frac{1}{2} \sqrt{\frac{15}{2\pi}} \sin \theta \cos \theta e^{-i\varphi} \\ Y_2^0(\theta, \varphi) &= \frac{1}{4} \sqrt{\frac{5}{\pi}} (3 \cos^2 \theta - 1) \\ Y_2^1(\theta, \varphi) &= \frac{-1}{2} \sqrt{\frac{15}{2\pi}} \sin \theta \cos \theta e^{i\varphi} \end{aligned} \quad (38)$$

$$Y_2^2(\theta, \varphi) = \frac{1}{4} \sqrt{\frac{15}{2\pi}} \sin^2 \theta e^{2i\varphi}$$

$$Y_3^0(\theta, \varphi) = \frac{1}{4} \sqrt{\frac{7}{\pi}} (5 \cos^3 \theta - 3 \cos \theta)$$

Les harmoniques sphériques constituent une base orthonormée dans l'espace de Hilbert $L^2(S^2)$.

Pour toute fonction $f \in L^2(S^2)$ on a

$$f = \sum_{l=0}^{\infty} \sum_{m=-l}^l \hat{f}_{l,m} Y_l^m \quad (39)$$

Où

$$\hat{f}_{l,m} = \hat{f}(l, m) = \langle f, Y_l^m \rangle \quad (40)$$

Où $\langle f, Y_l^m \rangle$ est le produit scalaire dans $L^2(S^2)$ défini par :

$$\langle f, Y_l^m \rangle = \int_0^{2\pi} \int_0^\pi f(\theta, \varphi) \overline{Y_l^m} \sin(\theta) d\theta d\varphi \quad (41)$$

Le coefficient $\hat{f}_{l,m}$ est appelé transformée de Fourier sphérique.

L'ensemble $\left\{ \hat{f}_{l,m}, m \in \mathbb{Z}, |m| \leq l \in \mathbb{N} \right\}$ est appelé spectre de f .

8-2-2- Moments invariants

Les invariants par les moments sont proposés par Hu [35] en 1962. Ils sont très utilisés dans le domaine de la reconnaissance, car ils sont indépendants de la représentation d'un caractère (la translation, la rotation et le changement d'échelle). Ils ont été appliqués par Fakir [32] pour la reconnaissance des caractères Arabes imprimés. Ces moments sont calculés sur la forme entière (bordure ainsi que son intérieur).

Soit f une fonction sommable sur $\mathbb{R}^2$. On définit le moment d'ordre (p, q) comme suit :

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x, y) dx dy, \text{ pour } p, q = 0, 1, 2, \dots \quad (42)$$

Les moments centraux peuvent être exprimés par :

$$\mu_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy, \text{ avec } \bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \quad (43)$$

Pour une image numérique, l'équation (38) devient :

$$\mu_{pq} = \sum_{(x,y) \in R} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (44)$$

On peut facilement vérifier que les moments centraux d'ordre $p + q \leq 3$ peuvent être calculés par les formules suivantes :

$$\begin{aligned} \mu_{00} &= m_{00} \\ \mu_{10} &= 0 \\ \mu_{01} &= 0 \\ \mu_{11} &= m_{11} - \bar{y}m_{10} \\ \mu_{20} &= m_{20} - \bar{x}m_{10} \\ \mu_{02} &= m_{02} - \bar{y}m_{01} \\ \mu_{12} &= m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}^2 m_{10} \\ \mu_{21} &= m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}^2 m_{01} \\ \mu_{30} &= m_{30} - 3\bar{x}m_{20} + 2\bar{x}^2 m_{10} \\ \mu_{03} &= m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2 m_{01} \end{aligned} \quad (45)$$

Les moments centraux sont invariants par translation. Ils peuvent être normalisés, pour conserver l'invariance, par changement d'échelle et on obtient les moments centraux normalisés :

$$\alpha_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}, \text{ avec } \gamma = \frac{p+q}{2} + 1, \text{ pour } p+q = 2, 3, \dots \quad (46)$$

Les invariants par les moments ont été obtenus par Hu (1962) [35] et fréquemment utilisés comme caractéristiques pour la reconnaissance des formes. Ils sont calculés sur la forme entière (frontière ainsi que son intérieur).

$$\begin{aligned}
 \varphi_1 &= \alpha_{20} - \alpha_{02} \\
 \varphi_2 &= (\alpha_{20} - \alpha_{02})^2 + 4\alpha_{11}^2 \\
 \varphi_3 &= (\alpha_{30} - \alpha_{12})^2 + (3\alpha_{12} - \alpha_{03})^2 \\
 \varphi_4 &= (\alpha_{30} + \alpha_{12})^2 + (\alpha_{21} + \alpha_{03})^2 \\
 \varphi_5 &= (\alpha_{30} - 3\alpha_{12})(\alpha_{30} + \alpha_{12})[(\alpha_{30} + \alpha_{12})^2 - 3(\alpha_{21} + \alpha_{03})^2] \\
 &+ (3\alpha_{21} - \alpha_{03})(\alpha_{21} + \alpha_{03})[3(\alpha_{30} + \alpha_{12})^2 - (\alpha_{21} + \alpha_{03})^2] \\
 \varphi_6 &= (\alpha_{20} - \alpha_{02})[(\alpha_{30} + \alpha_{12})^2 - (\alpha_{21} + \alpha_{03})^2] \\
 &+ 4\alpha_{11}(\alpha_{30} + \alpha_{12})(\alpha_{21} + \alpha_{03}) \\
 \varphi_7 &= (3\alpha_{21} - \alpha_{03})(\alpha_{30} + \alpha_{12})[(\alpha_{30} + \alpha_{12})^2 - 3(\alpha_{21} + \alpha_{03})^2] \\
 &+ (3\alpha_{12} - \alpha_{03})(\alpha_{21} + \alpha_{03})[3(\alpha_{30} + \alpha_{12})^2 - (\alpha_{21} + \alpha_{03})^2]
 \end{aligned} \tag{47}$$

Les quantités $\varphi_i, (1 \leq i \leq 7)$ (voir Hu [35]) sont invariantes par changement d'échelle, translation et rotation.

8-2-3- Moments invariants et transformation de Fourier

Puisque la transformation de Fourier est une fonction prédéfinie et déjà programmée en plusieurs logiciels (par exemple la fonction `fft` en Matlab), on l'a choisie pour calculer les moments invariants afin de réduire le temps d'exécution et minimiser l'espace mémoire utilisé. Ce qui suit démontre la relation entre les moments invariants et la transformation de Fourier.

Pour $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, sa transformée de Fourier est donnée par :

$$\hat{f}(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{-i2\pi(xu + yv)} dx dy \tag{48}$$

Cette transformation admet l'approximation discrète :

$$\hat{f}(u, v) = \sum_x \sum_y f(x, y) e^{-i2\pi(xu + yv)} \tag{49}$$

On définit aussi le moment d'ordre (p, q) par l'équation (42) dont l'approximation discrète est donnée par l'équation :

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \tag{50}$$

a/ Relation entre les moments et la transformation de Fourier :

Si on calcule

$$\begin{aligned} \frac{\partial^{p+q} \hat{f}(u, v)}{\partial u^p \partial v^q} &= \int_{-\infty-\infty}^{+\infty+\infty} \int_{-\infty-\infty}^{+\infty+\infty} (-i2\pi x)^p (-i2\pi y)^q f(x, y) e^{-i2\pi(xu+yv)} dx dy \\ &= (-i2\pi)^{p+q} \int_{-\infty-\infty}^{+\infty+\infty} \int_{-\infty-\infty}^{+\infty+\infty} x^p y^q f(x, y) e^{-i2\pi(xu+yv)} dx dy \end{aligned} \quad (51)$$

pour $(u, v) = (0, 0)$

on obtiendra

$$\begin{aligned} \frac{\partial^{p+q} \hat{f}(0, 0)}{\partial u^p \partial v^q} &= (-i2\pi)^{p+q} \int_{-\infty-\infty}^{+\infty+\infty} \int_{-\infty-\infty}^{+\infty+\infty} x^p y^q f(x, y) dx dy \\ &= (-i2\pi)^{p+q} m_{pq} \end{aligned} \quad (52)$$

alors

$$m_{pq} = \frac{\frac{\partial^{p+q} \hat{f}(0, 0)}{\partial u^p \partial v^q}}{(-i2\pi)^{p+q}} = Fe[x^p y^q f(x, y)]_{(0,0)} \quad (53)$$

avec Fe c'est la transformée de Fourier

D'où

$$\begin{aligned} \frac{\partial^{p+q} \hat{f}(u, v)}{\partial u^p \partial v^q} &= Fe[(-i2\pi)^{p+q} x^p y^q f(x, y)] \\ &= \sum_x \sum_y (-i2\pi)^{p+q} x^p y^q f(x, y) e^{-i2\pi(xu+yv)} \\ &= (-i2\pi)^{p+q} \sum_x \sum_y x^p y^q f(x, y) e^{-i2\pi(xu+yv)} . \end{aligned} \quad (54)$$

Ainsi on a

$$\begin{aligned} m_{00} &= \hat{f}(0, 0), \\ m_{10} &= \frac{\frac{\partial \hat{f}(0, 0)}{\partial u}}{(-i2\pi)} = Fe[xf(x, y)]_{(0,0)}, \end{aligned}$$

$$\begin{aligned}
 m_{01} &= \frac{\frac{\partial \hat{f}(0,0)}{\partial v}}{(-i2\pi)} = Fe[yf(x, y)]_{(0,0)}, \\
 m_{11} &= \frac{\frac{\partial^2 \hat{f}(0,0)}{\partial u \partial v}}{(-i2\pi)^2} = Fe[xyf(x, y)]_{(0,0)}, \\
 m_{02} &= \frac{\frac{\partial^2 \hat{f}(0,0)}{\partial v^2}}{(-i2\pi)^2} = Fe[y^2 f(x, y)]_{(0,0)}, \\
 m_{12} &= \frac{\frac{\partial^3 \hat{f}(0,0)}{\partial u \partial v^2}}{(-i2\pi)^3} = Fe[xy^2 f(x, y)]_{(0,0)}, \\
 m_{20} &= \frac{\frac{\partial^2 \hat{f}(0,0)}{\partial u^2}}{(-i2\pi)^2} = Fe[x^2 f(x, y)]_{(0,0)}, \\
 m_{21} &= \frac{\frac{\partial^3 \hat{f}(0,0)}{\partial u^2 \partial v}}{(-i2\pi)^3} = Fe[x^2 yf(x, y)]_{(0,0)}, \\
 m_{03} &= \frac{\frac{\partial^3 \hat{f}(0,0)}{\partial v^3}}{(-i2\pi)^3} = Fe[y^3 f(x, y)]_{(0,0)}, \\
 m_{30} &= \frac{\frac{\partial^3 \hat{f}(0,0)}{\partial u^3}}{(-i2\pi)^3} = Fe[x^3 f(x, y)]_{(0,0)}.
 \end{aligned} \tag{55}$$

Alors la conclusion que nous tirons à partir de cette étude théorique, c'est que nous pouvons utiliser la transformation de Fourier et ses dérivées pour calculer les valeurs moyennes pondérées d'une image f .

b/ Centre de gravité d'une image

Le centre de gravité d'une image se calcule comme suit :

$$\langle x \rangle = \frac{m_{10}}{m_{00}} = \frac{\frac{\partial \hat{f}(0,0)}{\partial u}}{(-i2\pi) \hat{f}(0,0)} = \frac{Fe[xf(x, y)]_{(0,0)}}{\hat{f}(0,0)} \tag{56}$$

$$\langle y \rangle = \frac{m_{01}}{m_{00}} = \frac{\frac{\partial \hat{f}(0,0)}{\partial v}}{(-i2\pi) \hat{f}(0,0)} = \frac{Fe[yf(x,y)]_{(0,0)}}{\hat{f}(0,0)}$$

c/ Variance de l'image (l'écart quadratique)

La variance d'une image se calcule comme suit :

$$\langle x^2 \rangle = \frac{m_{20}}{m_{00}} = \frac{\frac{\partial^2 \hat{f}(0,0)}{\partial u^2}}{(-i2\pi)^2 \hat{f}(0,0)} = \frac{Fe[x^2 f(x,y)]_{(0,0)}}{\hat{f}(0,0)} \quad (57)$$

$$\langle y^2 \rangle = \frac{m_{02}}{m_{00}} = \frac{\frac{\partial^2 \hat{f}(0,0)}{\partial v^2}}{(-i2\pi)^2 \hat{f}(0,0)} = \frac{Fe[y^2 f(x,y)]_{(0,0)}}{\hat{f}(0,0)}$$

8-2-4- Moments invariants modifiés

Dans le but de réduire les calculs, on a apporté des améliorations aux calculs classiques des moments invariants. Ces améliorations concernent l'utilisation de la frontière de la forme [51].

Si on utilise uniquement la frontière de la forme, la définition du moment (équation (42)) devient :

$$m_{pq} = \int_C x^p y^q f(x,y) ds \quad , \text{ pour } p, q = 0, 1, 2, 3, \dots \quad (58)$$

avec

$$\int_C : \text{L'intégrale sur la courbe } C \text{ et } ds = \sqrt{((dx)^2 + (dy)^2)} \quad (59)$$

Les moments centraux modifiés sont définis par :

$$\mu_{pq} = \int_C (x - \bar{x})^p (y - \bar{y})^q f(x,y) ds \quad (60)$$

Où

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{et} \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (61)$$

Pour un caractère numérique, l'équation (45) devient :

$$\mu_{pq} = \sum_{(x,y) \in C} (x - \bar{x})^p (y - \bar{y})^q f(x,y) \quad (62)$$

Les moments centraux modifiés sont invariants par translation. Après la normalisation de ces moments, ils deviennent invariants par changement d'échelle et par rotation.

Théorème1 :

Les moments centraux de l'équation (45) sont normalisés en utilisant la formule suivante :

$$\alpha'_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{p+q+1}} \quad (63)$$

Cette équation est invariante par changement d'échelles pour $p, q = 2, 3, \dots$

Preuve :

Soit C une courbe lisse dans le plan, et soit C' la courbe obtenue par changement d'échelle pondérant les coordonnées par un facteur r , alors :

$$\mu'_{pq} = \int_C (x(s'))^p (y(s'))^q ds' = r^{p+q+1} \mu_{pq} \quad (64)$$

Puisque

$$\mu_{00} = \int_C ds = |C| = \text{Longueur de la courbe } C \quad (65)$$

$$\mu'_{00} = \int_C ds' = |C'| = \text{Longueur de la courbe } C' = r|C| \quad (66)$$

Donc, pour $r > 0$ on a :

$$\frac{\mu'_{pq}}{(\mu'_{00})^{p+q+1}} = \frac{r^{p+q+1} \mu_{pq}}{r^{p+q+1} |C|} = \frac{\mu_{pq}}{(\mu_{00})^{p+q+1}} \quad (67)$$

Ainsi l'expression de l'équation (62) est invariante par changement d'échelle.

Théorème2 :

Soit C une courbe lisse dans le plan et C' une courbe obtenue par rotation de C d'un angle θ dans le sens des aiguilles d'une montre, alors :

$$\varphi'_i = \varphi_i \quad \text{pour } 1 \leq i \leq 7 \quad (68)$$

Où φ'_i est définie par l'équation (42) au moyen de α'_{pq} au lieu de α_{pq} pour $p + q = 2, 3, \dots$

Preuve :

$$\mu'_{pq} = \int_C (x(s'))^p (y(s'))^q ds' = \int_C \left\{ [x(s) \cos \theta - y(s) \sin \theta]^p [y(s) \sin \theta + x(s) \cos \theta]^q \right\} ds \quad (69)$$

Notons que

$$ds' = ds \quad (70)$$

A titre d'exemple, nous démontrons que $\varphi_1' = \varphi_1$ et $\varphi_2' = \varphi_2$.

Pour $3 \leq j \leq 7$, $\varphi_j' = \varphi_j$ peut être prouvée de manière analogue après un calcul en utilisant les identités trigonométriques

$$\cos^2 \theta + \sin^2 \theta = 1 \quad (71)$$

et

$$(\cos^3 \theta - 3\cos \theta \sin^2 \theta)^2 + (\sin^3 \theta - 3\cos^2 \theta \sin \theta)^2 = 1 \quad (72)$$

8-2-5- Transformation de Walsh

Abd-Elmageed et al [52] ont développé une méthode qui utilise la transformation de Walsh et qui donne un temps de traitement réduit. Attila et al [53] ont consacré leur travail à la reconnaissance des documents. Dans ce travail, l'algorithme adopté se base sur l'utilisation de la transformation de Walsh.

On définit la transformation de Walsh par [53] :

$$W(u, v) = \iint_{(x, y) \in IR} f(x, y) g(x, y, u, v) dx dy \quad (73)$$

Pour une image numérique la transformation de Walsh est donnée par la formule suivante :

$$W(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) g(x, y, u, v) \quad (74)$$

où

$f(x, y)$: l'intensité du pixel dont les coordonnées sont (x, y)

$u, v = 0, 1, 2, \dots, N - 1$

$N \times N$: La taille de l'image

g : La fonction de Kernel définie par l'équation suivante [53] :

$$g(x, y, u, v) = (1/N) \prod_{i=0}^{n-1} (-1)^{b_i(x)b_{n-i-1}(u) + b_i(y)b_{n-i-1}(v)} \quad (75)$$

Où $b_i(x)$ est le $i^{ème}$ bit dans l'expansion binaire de x (0 ou 1).

La transformation de Walsh est invariante par translation, rotation et changement d'échelle.

9- Conclusion

Dans ce chapitre, on a décrit des descripteurs invariants qui prennent en considération la représentation de la forme en quelques situations (translation, rotation et changement d'échelle), les descripteurs traités sont :

- La transformée de Fourier,
- La transformée de Fourier-Mellin,
- Les harmoniques sphériques,
- Les moments invariants,
- La transformation de Walsh.

1- Introduction

Le prétraitement est une étape importante dans la reconnaissance des formes. L'image acquise à l'ordinateur est affectée par le bruit, ceci dépend du matériel d'acquisition.

La numérisation peut être faite par un scanner plat ou à main. Dans ce travail nous avons utilisé un scanner plat car le scanner à main a typiquement une résolution faible. De plus, le résultat obtenu est affecté par le mouvement de la main crée des fluctuations locales.

Plusieurs processus ont été utilisés dans le prétraitement des caractères: la normalisation, le lissage, la segmentation et la squelettisation...etc

Dans ce chapitre, nous avons procédé à la normalisation de position, le filtrage, la correction d'inclinaison et la segmentation du document en lignes puis en caractères.

2- Normalisation

La normalisation de position est la première fonction à exécuter dans le prétraitement. Elle a pour but d'encadrer l'image d'entrée, c'est-à-dire d'éliminer les zones indésirables afin de réduire le temps de traitement et minimiser l'espace mémoire utilisé. L'opération consiste, premièrement, à détecter les deux lignes L1 et L2 et les deux colonnes C1 et C2 (Fig.8) pour déterminer les marges à supprimer :

- Marge en haut : délimitée par la première ligne de l'image et L1.
- Marge en bas : délimitée par la dernière ligne de l'image et L2.
- Marge à gauche : délimitée par la première colonne de l'image et C1.
- Marge à droite : délimitée par la dernière colonne de l'image et C2.

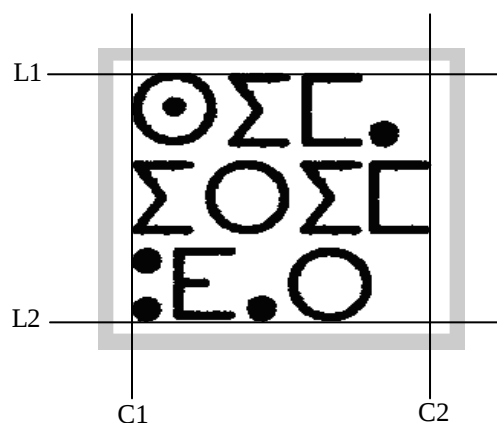


Fig. 8 : Détermination des marges

Deuxièmement, enlever les marges en ne gardant que l'espace délimité par les paramètres L1, L2, C1 et C2, qui correspond à l'image normalisée (Fig.9).



Fig. 9 : (a) Avant normalisation et (b) Après normalisation

3- Filtrage

Le filtrage est une opération fondamentale en traitement d'images. Il permet de réduire le bruit afin d'améliorer la qualité d'image.

Soient : $f(x, y)$ l'image source,

$H(x, y)$ la fonction qui représente le filtre,

$F(x, y)$ l'image filtrée.

L'image filtrée est le produit (convolution) de l'image source par la fonction qui représente le filtre.

$$F(x, y) = (G * f)(x, y) \quad (76)$$

Tel que

$$G(x, y) = H(-x, -y)$$

On en déduit que

$$F(x, y) = \iint f(u, v)G(x - u, y - v)dudv \quad (77)$$

En développant G en série, on obtient

$$G(x - u, y - v) = \sum_{p=0}^{+\infty} \sum_{q=0}^{+\infty} \frac{(-1)^{p+q} u^p v^q}{p!q!} \frac{\partial^{p+q} G(x, y)}{\partial u^p \partial v^q} \quad (78)$$

Donc

$$\begin{aligned}
 F(x, y) &= \iint f(u, v) \sum_{p=0}^{+\infty} \sum_{q=0}^{+\infty} \frac{(-1)^{p+q} u^p v^q}{p! q!} \frac{\partial^{p+q} G(u, v)}{\partial u^p \partial v^q} du dv \\
 &= \sum_{p=0}^{+\infty} \sum_{q=0}^{+\infty} \frac{(-1)^{p+q}}{p! q!} \frac{\partial^{p+q} G(u, v)}{\partial u^p \partial v^q} \iint u^p v^q f(u, v) du dv \\
 &= \sum_{p=0}^{+\infty} \sum_{q=0}^{+\infty} \frac{(-1)^{p+q}}{p! q!} m_{p,q}(f) \frac{\partial^{p+q} G(u, v)}{\partial u^p \partial v^q}
 \end{aligned} \tag{78-bis}$$

En conséquence, on peut calculer l'image filtrée à partir des moments d'une image et ses dérivées.

Aussi, l'image filtrée peut être calculée dans le domaine fréquentiel à l'aide de la formule suivante :

$$\begin{aligned}
 \hat{F}(u, v) &= TF[F(x, y)] \\
 \hat{F}(u, v) &= \sum_{p=0}^{+\infty} \sum_{q=0}^{+\infty} \frac{(-1)^{p+q}}{p! q!} m_{p,q} f \cdot Fe \left[\frac{\partial^{p+q} G(u, v)}{\partial u^p \partial v^q} \right] \\
 &= \sum_{p=0}^{+\infty} \sum_{q=0}^{+\infty} \frac{(-1)^{p+q}}{p! q!} m_{p,q} f \cdot (-i2\pi)^{p+q} u^p v^q \hat{G}(u, v) \\
 &= \hat{G}(u, v) \sum_{p=0}^{+\infty} \sum_{q=0}^{+\infty} \frac{(-1)^{p+q}}{p! q!} m_{p,q} f \cdot (-i2\pi)^{p+q} u^p v^q
 \end{aligned} \tag{79}$$

Et comme

$$\hat{F}(u, v) = \hat{G}(u, v) \hat{f}(u, v) \tag{80}$$

Alors

$$\hat{I}(u, v) = \sum_{p=0}^{+\infty} \sum_{q=0}^{+\infty} \frac{(-1)^{p+q}}{p! q!} (-i2\pi)^{p+q} m_{p,q} f \cdot u^p v^q \tag{81}$$

On obtient ainsi une relation entre la transformée de Fourier d'une image et ses moments.

4- Correction de l'inclinaison

L'inclinaison est considérée comme l'un des problèmes le plus fréquent dans une ROC car elle influence directement l'étape de la segmentation et par conséquent sur la reconnaissance. En effet, l'utilisation d'un scanner peut mener à un non-alignement de l'image du document.

L'angle d'inclinaison θ est l'angle que font les lignes de texte du document avec la direction horizontale (Fig.10).

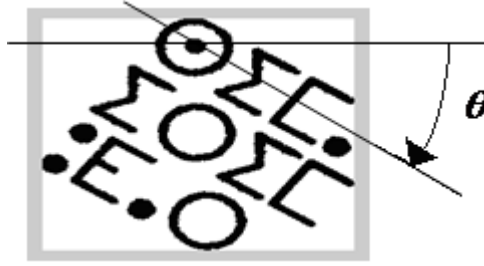


Fig. 10 : Angle d'inclinaison

Pour remédier à ce problème d'inclinaison (Fig.11), il faut tout d'abord détecter l'angle d'inclinaison, ensuite, corriger l'inclinaison en faisant une rotation dans le sens approprié.

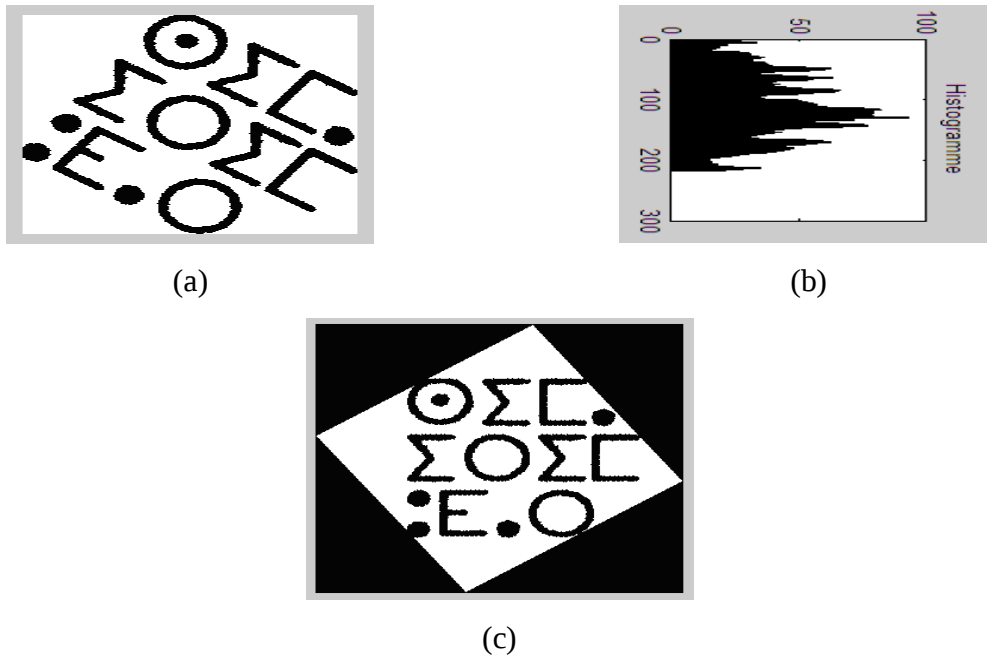


Fig. 11 : (a) Avant la correction d'inclinaison,
(b) Histogramme horizontal, (c) Après la correction d'inclinaison

Il existe une grande variété d'algorithmes pour la correction d'inclinaison basés sur le profil de projection, la transformation de Hough, la corrélation des lignes, etc, [3, 5, 8, 16, 54]. Mais, dans le présent travail, on se limite à utiliser la transformation de Hough pour la correction d'angle d'inclinaison.

La transformation de Hough est une opération importante en traitement d'image. Elle a été inventée en 1962 par Paul Hough [35] pour la détection des lignes droites dans les images numériques et aussi pour la reconnaissance des caractères.

Les propriétés de la transformation de Hough [35] sont :

- Un point dans le plan cartésien correspond à une courbe sinusoïdale dans le plan polaire $(\rho - \theta)$.
- Des points situés sur la même courbe dans le plan polaire correspondent à une famille de lignes passant par un point commun dans le plan $(x - y)$.
- Les points d'une même ligne droite dans le plan $(x - y)$ correspondent à une famille de courbes passant par un point commun dans le plan polaire $(\rho - \theta)$.
- Un point dans le plan polaire correspond à une ligne droite dans le plan cartésien.

5- Segmentation

La segmentation est la fonction qui suit la correction d'inclinaison. Elle consiste à détecter les lignes et les caractères dans l'image. La segmentation du texte consiste à segmenter tout d'abord le texte en lignes, puis les lignes en caractères.

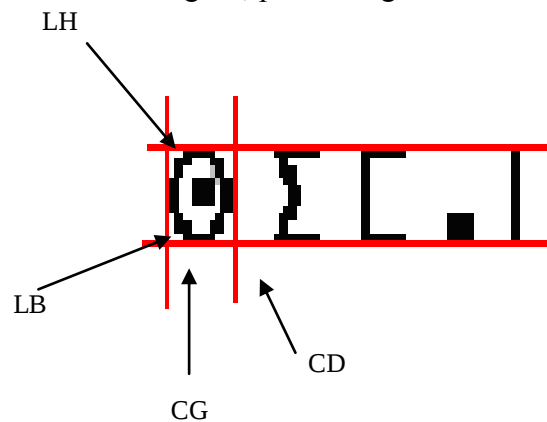


Fig. 12 : Détection des lignes et caractères

La figure (12) représente les quatre paramètres adoptés pour la détection des lignes et des caractères d'un texte écrit en Tifinaghe. La ligne est connue par les deux paramètres LH (Ligne Haute) et LB (Ligne Basse). Dans chaque ligne segmentée, CG (Colonne Gauche) et CD (Colonne Droite) sont utilisés pour déterminer le caractère.

5-1- Segmentation en lignes

Ce type de segmentation a pour but d'identifier les lignes dans un texte. Cela peut être réalisé en cherchant les paramètres LH et LB de chaque ligne (Fig. 13). Par exemple, la recherche des deux paramètres peut être effectuée en se basant sur la projection horizontale [23] (cette méthode sera traitée en détail dans le chapitre suivant).

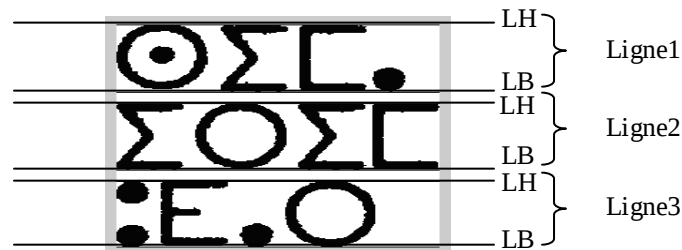


Fig. 13 : Segmentation en lignes

La figure (13) indique les paramètres qui permettent de déterminer les trois lignes de ce texte.

5-2- Segmentation en caractères

Dans chaque ligne segmentée, le caractère est localisé à l'aide des deux paramètres CG et CD qui peuvent être calculés, par exemple, en utilisant la méthode de la projection verticale [24] que nous détaillerons dans le chapitre qui suit. La segmentation en caractères consiste à trouver ces deux paramètres de chaque caractère.

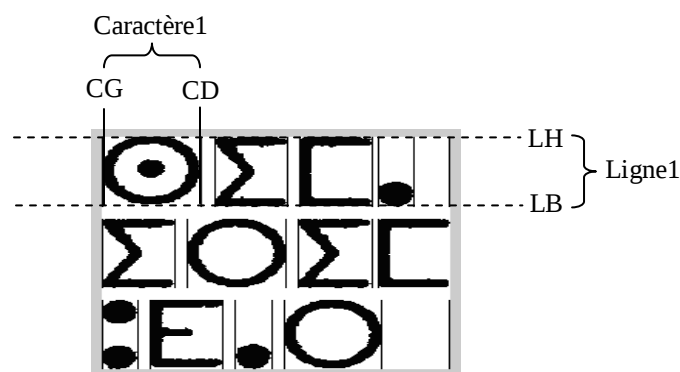


Fig. 14 : Segmentation en caractères

La figure (14) montre les paramètres CG et CD nécessaires pour segmenter le premier caractère de la première ligne.

6- Conclusion

Dans ce chapitre, on a entamé les fonctions de prétraitement : la normalisation, le filtrage, l'inclinaison et la segmentation.

La normalisation a comme objectif d'éliminer les zones indésirables afin de réduire le temps d'exécution.

Le filtrage est basé sur les moments et ses dérivés pour calculer l'image filtrée.

La fonction d'inclinaison a employé la transformation de Hough pour la détection et la correction de l'angle d'inclinaison.

Finalement, la dernière fonction de prétraitement consiste à segmenter le texte en lignes, puis les lignes en caractères.

1- Introduction

La reconnaissance est une phase primordiale dans une ROC, elle est appliquée après le prétraitement et l'extraction d'attributs. La décision trouvée par cette phase juge la robustesse d'un système de reconnaissance. C'est la raison du choix des méthodes efficaces et les plus célèbres dans le domaine de reconnaissance. Parmi ces méthodes, on trouve la méthode structurelle [23, 37, 55, 56, 57, 58] et les réseaux de neurones [10, 21, 25, 59] qu'on a utilisé pour la reconnaissance des caractères Tifinaghe.

Dans ce chapitre, on traite la programmation dynamique, les réseaux de neurones et les mémoires associatives.

2- Notion de structure

La structure doit décrire la composition d'un objet complexe en termes d'objets simples ainsi que les liens entre eux.

Le modèle de structure le plus simple est la chaîne dont le principe consiste à représenter une forme comme une suite d'objets élémentaires. Le codage de Freeman est un exemple de méthode qui peut être utilisé pour décrire une forme par une chaîne selon 4 directions (Fig.15).

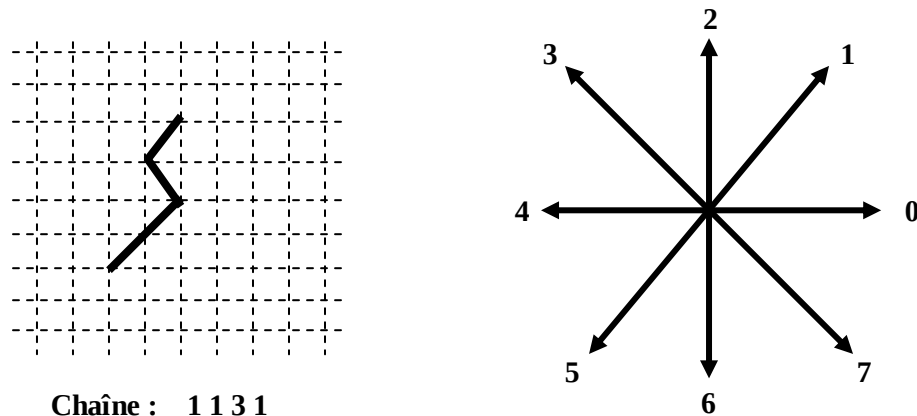


Fig. 15: Codage de Freeman

La chaîne **1131** (Sens 1, Sens 1, Sens 3 et Sens 1) représente le parcours de la forme en utilisant le schéma de déplacement contenant huit sens (0, 1, 2, 3, 4, 5, 6 et 7).

3- Notion de similarité

La plupart des techniques de classification font appel à des mesures de distances, appelé métrique, pour évaluer les degrés de dissemblance ou de ressemblance entre deux formes. Il est à noter que, d'une manière générale, une distance $d : E \times E \rightarrow IR_+$ est une application à valeurs positives ou nulles obéissant aux propriétés suivantes :

$$1) \quad \forall (a, b) \in E^2, a \neq b \Rightarrow d(a, b) > 0 \quad (\text{séparabilité}) \quad (82)$$

$$2) \quad \forall (a, b) \in E^2, d(a, b) = 0 \Leftrightarrow a = b \quad (83)$$

Cela signifie que deux points ont une dissimilarité égale à zéro si et seulement si ils sont confondus.

$$3) \quad \forall (a, b) \in E^2, d(a, b) = d(b, a) \quad (\text{symétrie}) \quad (84)$$

$$4) \quad \forall (a, b, c) \in E^2, d(a, c) \leq d(a, b) + d(b, c) \quad (\text{Inégalité triangulaire}) \quad (85)$$

La propriété 4 signifie que le chemin le plus court d'un point à un autre point est la liaison directe.

Si un indice de dissimilarité possède les propriétés précédentes, on l'appelle généralement une métrique et le couple (E, d) un espace métrique.

4- Mesure de distance

Soient :

E un ensemble de vecteurs

$A = \{a_1, a_2, \dots, a_n\}$ un vecteur de E

$B = \{b_1, b_2, \dots, b_n\}$ un vecteur de E

On peut définir un certain nombre de distances classiques :

- Distance de Hamming :

$$d_H(A, B) = \sum_{i=1}^n |a_i - b_i| \quad (86)$$

- Distance Euclidienne :

$$d_e(A, B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (87)$$

- Distance Rectangulaire :

$$d_r(A, B) = \sum_{i=1}^n |a_i - b_i| \quad (88)$$

- Distance de Minkowski :

$$d_n^q(A, B) = \left(\sum_{i=1}^n |a_i - b_i|^q \right)^{\frac{1}{q}} \quad (89)$$

L'équation (89) représente la définition générale de la distance. En effet, la distance de Hamming est obtenue dans le cas où $q = 1$, et pour $q = 2$ on trouve la distance Euclidienne.

5- Programmation Dynamique

La programmation dynamique est un concept introduit par Bellman [55], dans les années 50, pour résoudre les problèmes d'optimisation. Elle a été appliquée à la reconnaissance de la parole par Sakoe [57] et à la reconnaissance des caractères Latin par Tappert [56].

La programmation dynamique (Fig.16) est à la fois simple et puissante, elle permet la recherche du chemin optimal entre deux points. Elle est fondée sur le principe d'optimalité de Bellman [55]. Ce principe indique que tout chemin optimal est composé de portions de chemins elles-mêmes optimales. En effet, si on considère un chemin optimal entre deux points A et B et un point quelconque C de ce chemin, alors les deux chemins de A à C et de C à B sont aussi optimaux.

Si A est un caractère formé de I vecteurs a_i (rappelons que chaque vecteur représente une colonne de pixels de l'image d'un caractère) et si B est un autre caractère formé de J vecteurs b_j , on peut mesurer la différence entre a_i et b_j par la distance notée $d(a_i, b_j)$:

$$d(a_i, b_j) = \|a_i - b_j\| \quad (90)$$

définie pour chaque point (i, j) du plan ;

Si les vecteurs a_i et b_j sont de dimension P (P représente le nombre de paramètres représentatifs de chaque colonne de pixels de l'image d'un caractère) en utilisant la distance de Minkowsky d'ordre 1 [35], on obtient :

$$d(a_i, b_j) = \sum_{p=1}^P |a_i(p) - b_j(p)| \quad (91)$$

L'ajustement non linéaire des échelles spatiales de A et de B est obtenu en définissant un chemin $C(k)$ dans le plan $[i, j]$ et en transformant i en $n(k)$ et j en $m(k)$:

$$\{C(k) = (n(k), m(k)), k = 1 \text{ à } K\} \quad (92)$$

Où K représente le nombre de segments constituant le chemin parcouru dans le plan $[i, j]$ (Fig.16). Pour représenter la réalité physique, les fonctions $n(k)$ et $m(k)$ doivent être croissantes et doivent respecter certaines conditions de continuité. Ainsi, nous supposons que les seuls chemins valides de déplacement à partir d'un point (i, j) sont les points:

$(i-1, j)$, $(i-1, j-1)$ et $(i, j-1)$ (Fig.16). En supposant que les frontières des caractères sont correctement définies, on prendra $C(1) = (1,1)$ et $C(k) = (I, J)$.

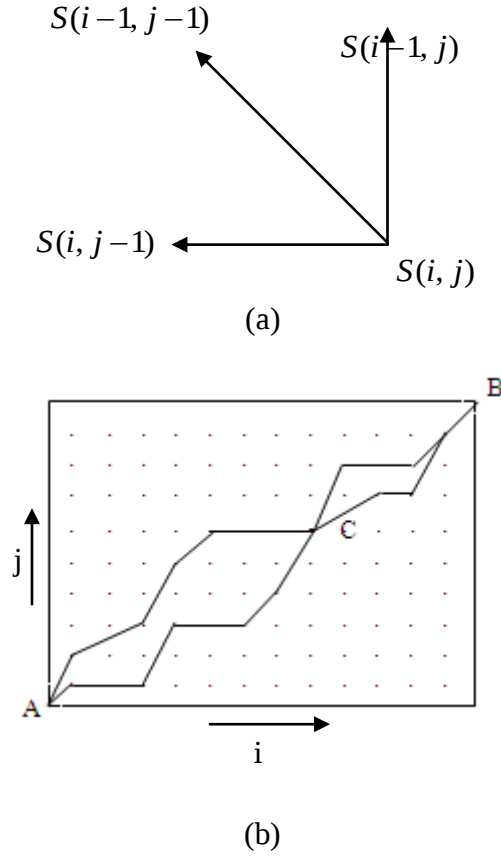


Fig. 16 : (a) Principe de recherche, (b) Principe de la PD

La méthode consiste à choisir parmi tous les chemins physiquement possibles, celui qui passe par les distances $d(a_i, b_j)$ les plus petites, de sorte que la somme des distances le long de ce chemin soit minimale, la dissemblance entre A et B est défini ainsi:

$$D(A, B) = \min \left[\frac{\sum_{k=1}^K d(C(k)) \omega(k)}{N(\omega)} \right] \quad (93)$$

où $\omega(k)$ est un coefficient de pondération appliqué sur le $k^{\text{ième}}$ segment du chemin C et $N(\omega)$ est un coefficient de normalisation qui dépend de la fonction ω .

Pour évaluer $D(A, B)$ définie par l'équation (93), nous avons retenu la fonction de pondération [27] suivante :

$$\omega(k) = n(k) - n(k-1) \quad (94)$$

et le coefficient :

$$N(\omega) = \sum_{k=1}^K \omega(k) = I \quad (95)$$

Ce coefficient est choisi tel que la distance $D(A, B)$ soit indépendante des longueurs de A et B et de la longueur et de la forme de C .

Le problème d'optimisation, décrit par l'équation (93), peut être efficacement résolu par un algorithme de programmation dynamique [35]. Ce dernier est grandement simplifié par le fait que $N(\omega)$ indépendante de C , car le problème se réduit alors à la minimisation du numérateur de (93). Ainsi, avec les chemins valides arrivant au point (i, j) (Fig.16), la fonction de pondération (94) et le coefficient de normalisation (95), on obtient la relation récursive suivante :

$$S(i, j) = d(a_i, b_j) + \min \begin{cases} S(i-1, j) \\ S(i-1, j-1) \\ S(i, j-1) \end{cases} \quad (96)$$

où $S(i, j)$ est la distance cumulée le long du chemin optimal à partir du point $(1,1)$ jusqu'à point (i, j) . $S(i, j)$ est évaluée sur tout le domaine $[1, I] \times [1, J]$ qui est parcouru colonne par colonne et ligne par ligne à partir du point $(1,1)$. On obtient finalement :

$$D(A, B) = \frac{S(I, J)}{I} \quad (97)$$

6- Réseaux de neurones

6-1- Introduction

Le souci d'améliorer l'efficacité des systèmes de reconnaissance est une raison pour introduire de nouvelles techniques de classification. Parmi ces techniques, on trouve les réseaux de neurones.

Les réseaux de neurones est une approche connue par sa robustesse dans le domaine de la reconnaissance des caractères.

Les premiers travaux concernant les réseaux de neurones ont démarré en 1943 par W. McCulloch et W. Pitts [58] qui ont présenté un neurone formel qui est une abstraction du neurone physiologique.

En 1949, D. Hebb [60] présente dans son ouvrage "The Organization of Behavior" une règle d'apprentissage.

En 1960, le premier Perceptron est développé par F. Rosenblatt [58]. C'est un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones : une couche de perception et une couche liée à la prise de décision.

Puis, durant les années 1970 [58] il y eut une remise en cause de l'intérêt des réseaux car les ordinateurs de neurones apprenaient lentement, coûtaient très cher et leurs performances n'étaient pas si impressionnantes. Vers la fin des années 1970, les réseaux de neurones ont connu un nouveau départ.

En 1982, J. Hopfield [61] a donné une étude sur un réseau complètement bouclé.

Actuellement, les réseaux neuronaux sont utilisés dans de nombreux domaines (par exemple : la reconnaissance [21, 27]) à cause de leur propriété et leur capacité d'apprentissage. Ils sont utilisés par Hamza [9] à la reconnaissance des caractères Arabe et Bouikhalene et al [21] à la reconnaissance des caractères Tifinaghe.

6-2- Neurones biologiques

Les neurones biologiques (Fig.17) sont des cellules vivantes spécialisées dans le traitement des signaux électriques. Ils sont reliés entre eux par des liaisons appelées axones. Ces axones vont eux-mêmes jouer un rôle important dans le comportement logique de l'ensemble. Ils conduisent les signaux électriques de la sortie d'un neurone vers l'entrée (synapse) d'un autre neurone.

Les neurones font une sommation des signaux reçus en entrée et en fonction du résultat obtenu vont fournir un courant en sortie.

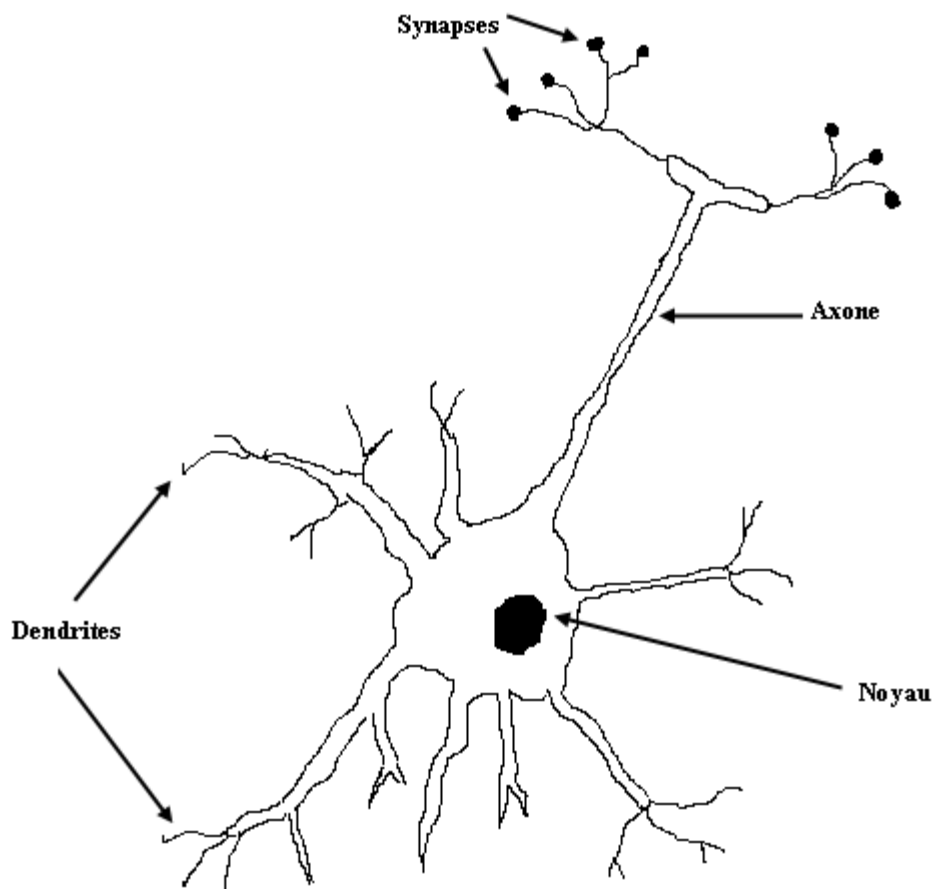


Fig. 17 : Neurone biologique

6-3- Neurone artificiel

Un réseau de neurones est composé d'éléments simples (ou neurones) fonctionnant en parallèle. Ces éléments ont été fortement inspirés par le système nerveux biologique. Il est constitué d'un graphe pondéré orienté dont les nœuds symbolisent les neurones formels. Ces derniers possèdent une fonction d'activation qui permet d'influencer les autres neurones du réseau. Les connexions entre les neurones, que l'on nomme liens synaptiques, propagent l'activité des neurones avec une pondération caractéristique de la connexion.

On peut entraîner un réseau de neurones pour une tâche spécifique (reconnaissance de caractères par exemple) en ajustant les valeurs des connections (ou poids) entre les éléments (neurones).

La figure (Fig.18) présente un modèle de neurone artificiel élaboré par W. McCulloch et W. Pitts [58].

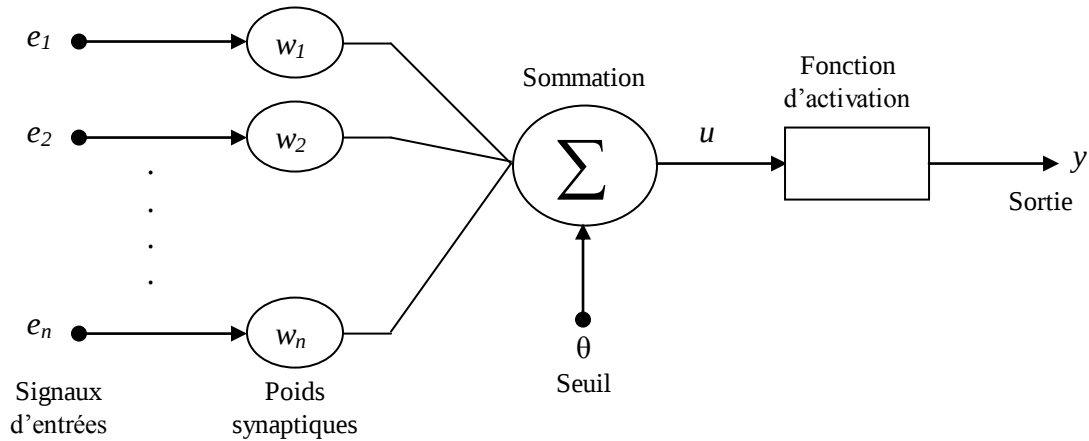


Fig. 18 : Neurone artificiel

Le neurone formel (Fig.18) est une unité élémentaire d'un réseau de neurones. Il effectue la somme pondérée de ses entrées, aussi, la valeur de cette sommation est comparée à un seuil. La sortie du neurone est :

$$u = \sum_{i=1}^n w_i \cdot e_i - \theta \quad (98)$$

$$y = f(u) \quad (99)$$

Tel que

- e_i : élément du vecteur d'entrée,
- $i = 1 \dots n$: les entrées du neurone formel,
- y : la sortie,
- w_i : les paramètres de pondération,
- f : la fonction d'activation.

6-4- Fonctions d'activation

La fonction d'activation ou bien la fonction de transfert est une fonction qui renvoie un réel appartenant généralement à l'intervalle $[0,1]$. Plusieurs fonctions peuvent être utilisées pour l'activation [58] ; les plus célèbres sont : la fonction sigmoïde, la fonction échelon, la fonction Gaussienne et la fonction identité.

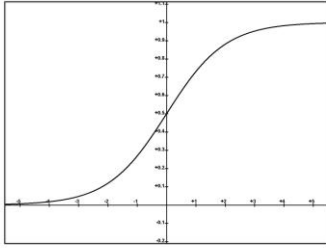
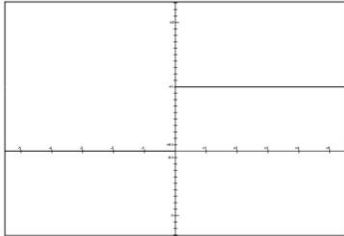
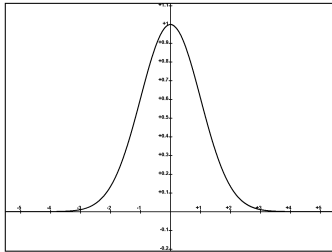
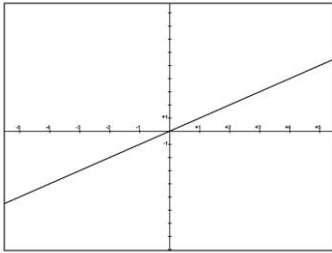
Sigmoïde		$f(u) = \frac{1}{1 + \exp(-u)}$
Echelon		• $f(u) = 0$ si $u < 0$ • $f(u) = 1$ si $u \geq 0$
Gaussienne		$f(u) = \exp\left(\frac{-u^2}{2}\right)$
Identité		$f(u) = u$

Tableau 2 : Fonctions d'activation

6-5- Structure d'un réseau de neurones

L'organisation des neurones définit l'architecture et la topologie du réseau.

6-5-1- Réseau multicouche

La figure (Fig.19) représente un exemple de réseau multicouche contenant des neurones qui sont arrangés par couche [58]. Dans ce réseau, les couches extérieures sont appelées respectivement couches d'entrée et de sortie ; les couches intermédiaires sont appelées couches cachées. Pour la couche d'entrée, elle reçoit le vecteur d'entrée.

Concernant les connexions ; tous les neurones de la couche amont sont connectés à tous les neurones de la couche aval, aussi, il n'y a pas de connexions entre les neurones d'une même couche.

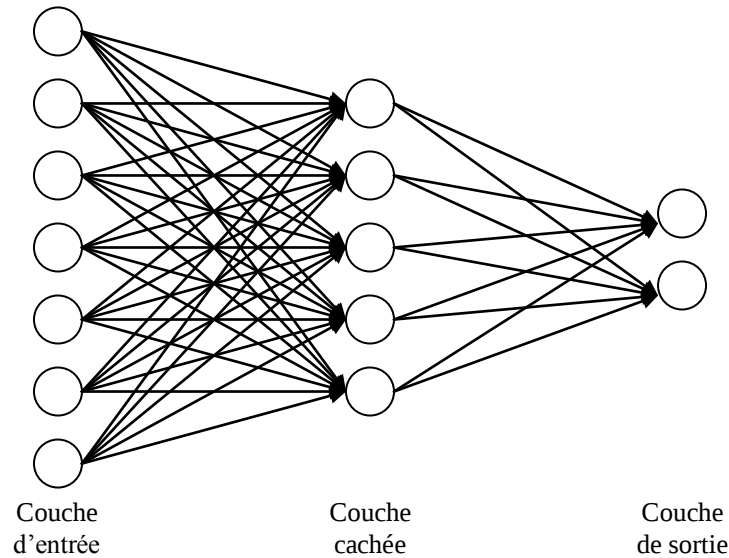


Fig. 19 : Réseau multicouche

6-5-2- Réseau à connexions locales

Ce type de réseau est multicouche [58], mais il se diffère du réseau illustré dans la figure (Fig.19). Cette différence existe dans le nombre de connexions ; c'est-à-dire que le réseau de la figure (Fig.20) contient moins de connexions que le réseau de la figure (Fig.19).

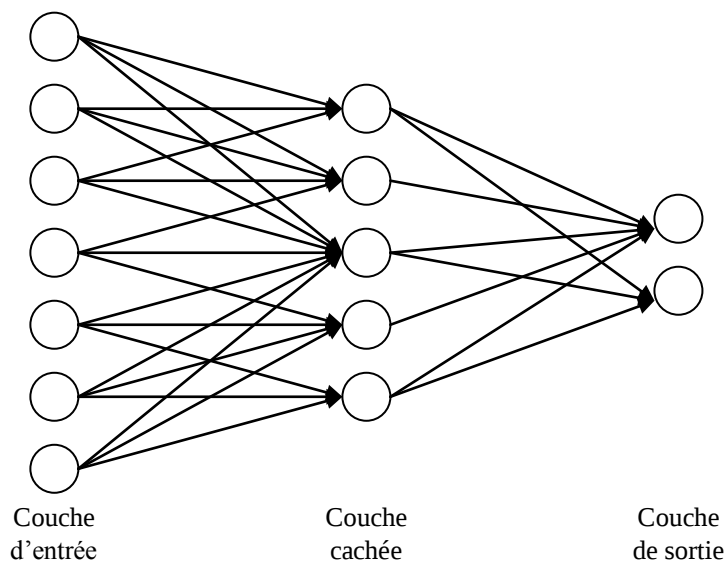


Fig. 20 : Réseau à connexions locales

6-5-3- Réseau à connexions récurrentes

Le réseau de la figure (Fig.21) est caractérisé par des connexions récurrentes qui ramènent l'information en arrière. En effet, il existe des connexions entre des neurones de la même couche, aussi, entre des neurones d'une couche aval et d'une couche amont [58].

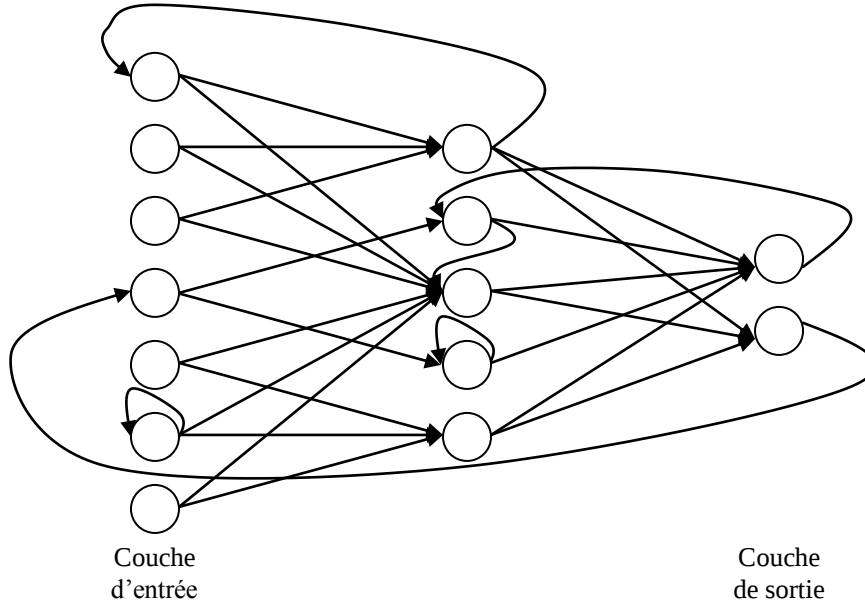


Fig. 21 : Réseau à connexions récurrentes

6-5-4- Réseau à connexions complexes

Dans ce réseau, chaque neurone est connecté à tous les neurones du réseau y compris lui-même [58] (Fig.22).

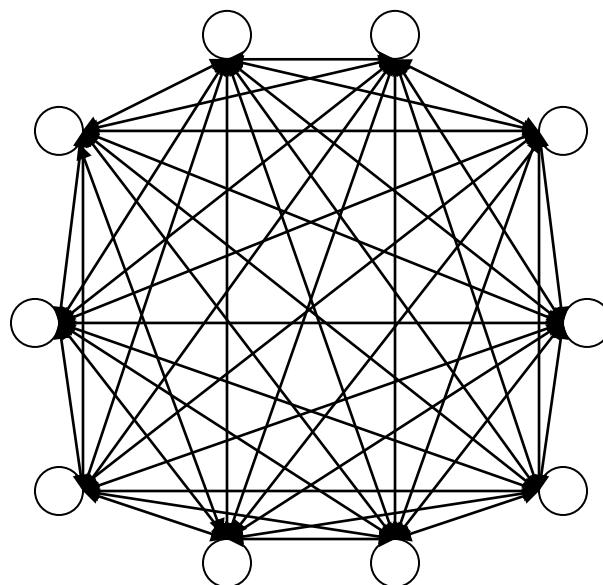


Fig. 22 : Réseau à connexions complexes

6-6- Apprentissage

L'apprentissage est une phase importante qui permet au réseau de neurones de trouver les valeurs de ces poids en ajustant ces coefficients par une règle d'apprentissage.

On distingue deux types d'apprentissage : apprentissage supervisé et apprentissage non supervisé.

6-6-1- Apprentissage supervisé

Dans ce genre d'apprentissage, le réseau est forcé à trouver des sorties bien déterminées. En effet, les entrées et les sorties correspondantes sont fournies au réseau.

A l'aide d'une règle d'apprentissage, le réseau calcule les valeurs de ces poids afin que pour une entrée donnée, la sortie réelle du réseau soit la plus proche de la sortie souhaitée.

6-6-2- Apprentissage non supervisé

L'apprentissage est dit non-supervisé lorsque seuls les valeurs d'entrée sont disponibles. Dans ce cas, les exemples présentés à l'entrée provoquent une auto-adaptation du réseau afin de produire des valeurs de sortie qui soient proches en réponse à des valeurs d'entrées similaires (de même nature).

6-6-3- Exemple d'utilisation d'un réseau multicouche

Parmi les types de réseaux, nous avons un exemple de réseau de neurones multicouche (Fig. 23) qui contient une couche cachée.

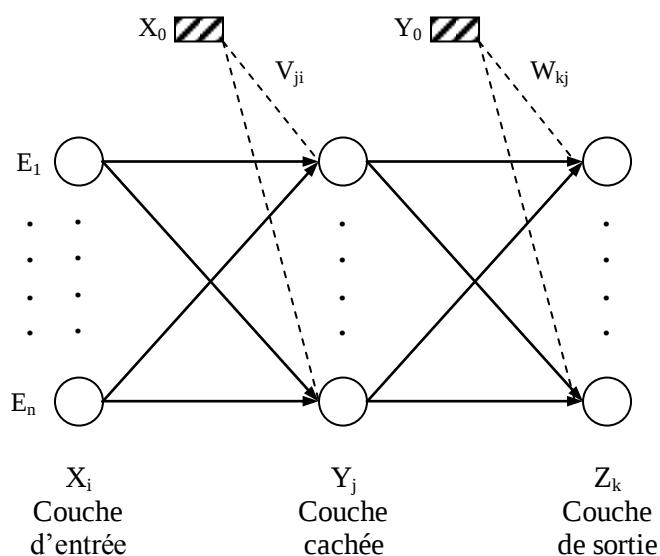


Fig. 23 : Réseau multicouche avec une couche cachée

Cet exemple de réseaux (Fig. 23) comporte :

- Une couche d'entrée à n cellules d'entrées (n éléments du vecteur d'entrée) $E_i = X_i$
- Une couche cachée de m neurones d'activations Y_j
- Une couche de sortie de r neurones (pour la représentation binaire des 33 caractères de Tifinaghe) d'activations Z_k
- $n \times m$ connexions entre la couche d'entrée et la couche cachée, chacune pondérée par V_{ji}
- $m \times r$ connexions entre la couche cachée et la couche de sortie, chacune pondérée par W_{kj}
- V_{ji} et W_{kj} sont des poids synaptiques
- X_0 et Y_0 sont des seuils

En général, l'apprentissage des réseaux de neurones est effectué de telle sorte qu'à chaque entrée particulière présentée au réseau corresponde une cible spécifique. L'ajustement des poids se fait par comparaison entre la réponse du réseau (ou sortie) et la cible, jusqu'à ce que la sortie corresponde (au mieux ?) à la cible. Ce type d'apprentissage est dit supervisé.

Pour notre exemple, le principe d'apprentissage (Fig. 24) repose sur les étapes suivantes :

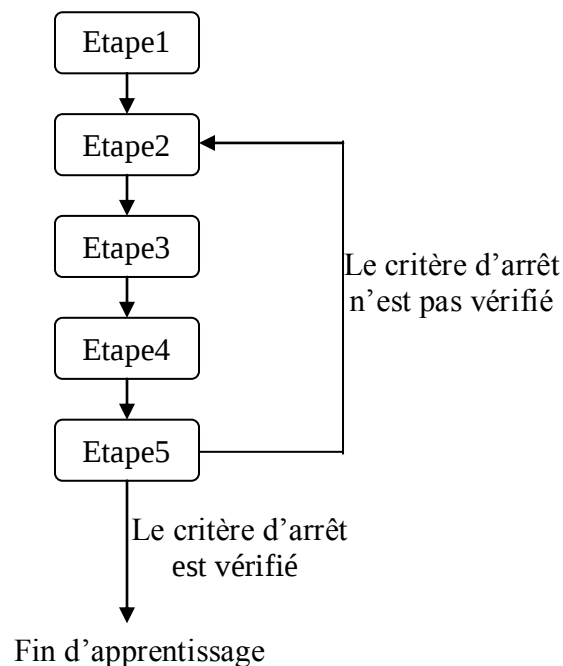


Fig. 24 : Principe d'apprentissage

Etape 1: (Initialisation des poids des connexions)

Initialisation des poids à des petites valeurs et des seuils du réseau.

Etape 2: (Propagation des entrées)

Les entrées E_i sont présentées à la couche d'entrée : $X_i = E_i$

La propagation vers la couche cachée se fait à l'aide de la formule suivante :

$$Y_j = f\left(\sum_{i=1}^n X_i V_{ji} + X_0\right) = \frac{1}{1 + \exp\left(-\left(\sum_{i=1}^n X_i V_{ji} + X_0\right)\right)} \quad (100)$$

Ensuite de la couche cachée vers la couche de sortie, on adopte :

$$Z_k = f\left(\sum_{j=1}^m Y_j W_{kj} + Y_0\right) = \frac{1}{1 + \exp\left(-\left(\sum_{j=1}^m Y_j W_{kj} + Y_0\right)\right)} \quad (101)$$

X_0 et Y_0 sont des seuils

f est la fonction d'activation (fonction sigmoïde) :

$$f(a) = \frac{1}{1 + \exp(-a)} \quad (102)$$

Avec

$$a = \sum_{i=1}^n X_i V_{ji} + X_0 \quad (\text{Le cas de la couche cachée}) \quad (103)$$

Et

$$a = \sum_{j=1}^m Y_j W_{kj} + Y_0 \quad (\text{Le cas de la couche de sortie}) \quad (104)$$

Etape 3: (Rétropropagation de l'erreur)

Au niveau de la couche de sortie, l'erreur entre la sortie désirée S_k et la sortie réelle Z_k est calculée par :

$$E_k = Z_k (1 - Z_k) (S_k - Z_k) \quad (105)$$

L'erreur calculée est propagée sur la couche cachée en utilisant la formule suivante:

$$F_j = Y_j(1 - Y_j) \sum_{k=1}^r W_{kj} \cdot E_k \quad (106)$$

Etape 4: (Correction des poids des connexions)

On corrige les poids de connexions entre la couche d'entrée et la couche cachée par :

$$\Delta V_{ji} = \eta \cdot X_i \cdot F_j \quad \text{et} \quad \Delta Y_0 = \eta \cdot F_j \quad (107)$$

Puis, on modifie les connexions entre la couche cachée et la couche de sortie par :

$$\Delta W_{kj} = \eta \cdot Y_j \cdot E_k \quad \text{et} \quad \Delta X_0 = \eta \cdot E_k \quad (108)$$

Avec η une constante positive appelée taux d'apprentissage (un paramètre à déterminer empiriquement).

Etape 5: (Boucle)

Boucler à l'étape2 jusqu'à l'obtention d'un critère d'arrêt à définir (seuil d'erreur, nombre d'itérations).

Après l'apprentissage et l'exécution d'OCR, on utilise la distance euclidienne pour identifier les caractères de Tifinaghe :

$$d(t_k, o) = \left(\sum_{i=1}^r (t_{ki} - o_i)^2 \right)^{1/2} \quad (109)$$

Avec, t_k la sortie désirée et o la sortie du réseau.

6-7- Mémoires associatives

Le but d'utiliser le concept de la mémoire associative, dans le domaine de la reconnaissance, est de reconnaître une information à partir d'une information partielle ou bruitée. Les mémoires associatives ont été proposées par plusieurs auteurs pour résoudre les problèmes de reconnaissance [60, 61, 62, 63].

Rougier [62] a présenté, en premier lieu, les différents types de mémoires, en suite, l'utilisation de ces mémoires dans le domaine de la robotique autonome. Fukushima [63] a proposé dans son travail un modèle de mémoire associative pour des patterns spatiaux variant dans le temps et sa simulation sur ordinateur. C'est un réseau composé de nombreux neurones qui montre une capacité de mémoire associative similaire à celle du cerveau.

« La mémoire est un processus de stockage et d'exploitation d'une connaissance antérieurement acquise, ce processus s'opérant sur la base d'une modification des propriétés d'un support physique. » [62].

Le processus de mémorisation d'une connaissance (Fig. 25) passe par trois étapes importantes : l'acquisition, le stockage et l'exploitation.

L'acquisition représente la phase d'apprentissage, la mémorisation correspond à la conservation, tandis que l'exploitation signifie la reconnaissance.

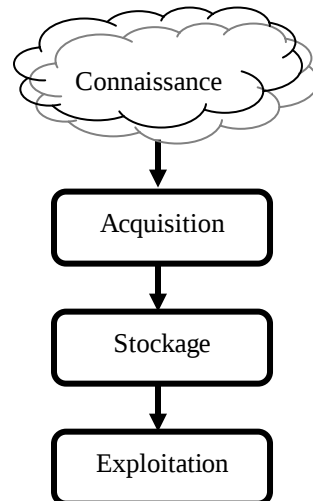


Fig. 25 : Etapes de mémorisation

Dans la notion « mémoire associative » [58], le terme « mémoire » se réfère à la fonction de stockage de l'information et le terme « associative » au mode d'adressage. Cette notion permet de retrouver l'information à partir d'une connaissance partielle de cette information.

6-7-1- Fonctionnement de la mémoire associative

La figure (Fig. 26) illustre un exemple d'une mémoire associative composée d'un ensemble de neurones interconnectés.

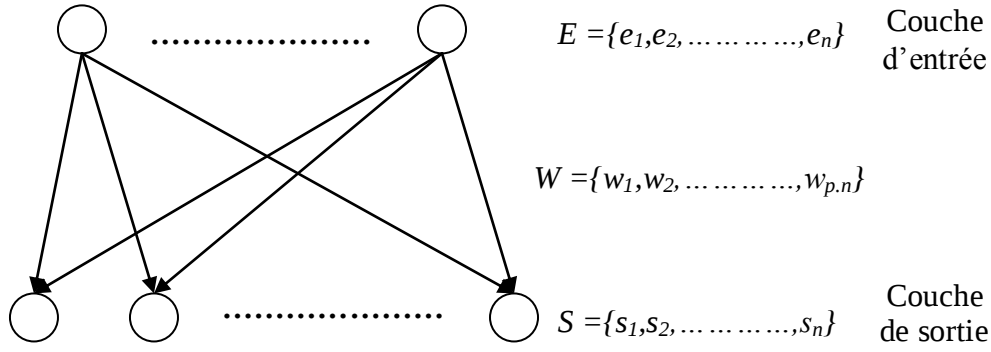


Fig. 26 : Exemple de mémoire associative

La structure de la mémoire associative de la figure (Fig.26) contient :

- n neurones sur la couche d'entrée
- p neurones sur la couche de sortie
- np poids entre la couche d'entrée et la couche de sortie

Le fonctionnement de cette mémoire se déroule comme suit :

- A chaque vecteur E_i de dimension n ($E_{i_1}, E_{i_2}, \dots, E_{i_n}$) est associé un vecteur de sortie S_i de dimension p ($S_{i_1}, S_{i_2}, \dots, S_{i_p}$).
- La relation entre E_i et S_i est linéaire tel que : $S_i = W.E_i$ ou W est une matrice de poids de dimension $n \times p$, ($i = 1, 2, \dots, N$) et N représente le nombre d'éléments de la base d'apprentissage.

L'apprentissage utilisé est de type supervisé. Il adopte la loi de Hebb [58] pour modifier les poids par itérations sur les N éléments de la base d'apprentissage. Pour une entrée E_k , on force la sortie à S_k . A la fin d'apprentissage, on aura : $W = \sum_{k=1}^N S_k.E_k^T$ où E_k^T est le transposé de E_k .

6-7-2- Types de mémoires associatives

Selon le nombre de couches, les mémoires associatives se divisent en deux catégories : hétéro-associative et auto-associative.

○ **Hétéro-associative**

Ce type de mémoire (Fig.27) comporte deux couches : une couche d'entrée et une couche de sortie. Les deux couches sont reliées par des connexions de poids. Les mémoires hétéro-associatives est une généralisation du perceptron, où l'activation de sortie est continue et non plus 0 ou 1.

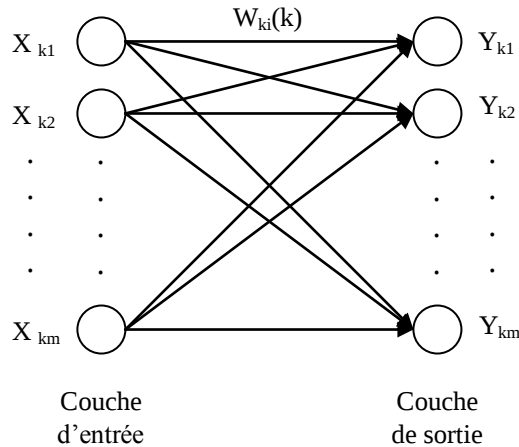


Fig. 27 : Hétéro-associative

○ **Auto-associative**

Ce deuxième genre (Fig.28) ne contient qu'une seule couche, tous les neurones de sortie sont reliés deux à deux. La recherche de réponse peut se faire soit par combinaison linéaire des stimuli stockés, soit de façon linéaire, on parle alors de réseaux de Hopfield [61]. Dans les réseaux de Hopfield, les réponses sont binaires et valent -1 ou 1 et les mises à jour sont synchrones.

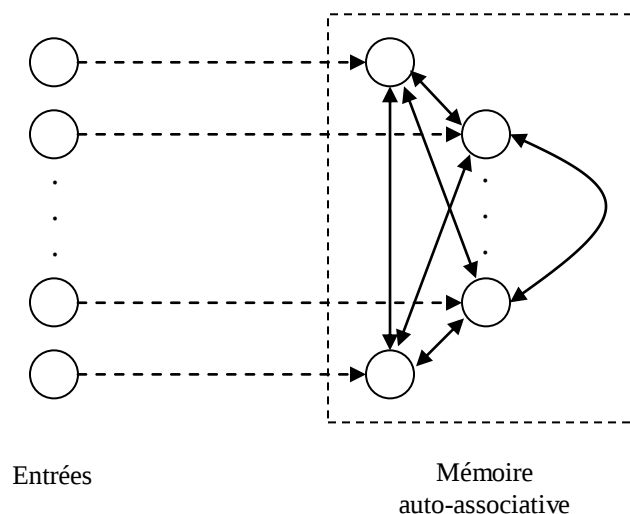


Fig. 28 : Auto-associative

6-7-3- Exemple d'utilisation d'une mémoire hétéro-associative

La figure (Fig.29) représente un exemple de mémoire hétéro-associative.

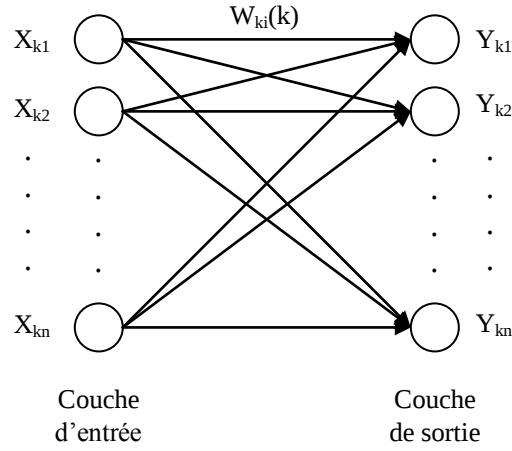


Fig. 29 : Mémoire hétéro-associative

Cet exemple (Fig.29) de mémoire contient :

- X_k : vecteur d'entrée.
- Y_k : vecteur de sortie.
- $W_{ki}(k)$: poids.

Pour se propager vers la couche de sortie, on utilise la formule suivante :

$$Y_{kj} = f\left(\sum_{i=1}^n X_{ki} W_{ji}(k)\right) = \sum_{i=1}^n X_{ki} W_{ji}(k) \quad (110)$$

Tels que :

- $j = 1, 2, \dots, m$ (m : c'est le nombre de neurones de la couche de sortie)
- $k = 1, 2, \dots, r$ (r : c'est le nombre de vecteurs de la couche d'entrée)
- f est la fonction d'activation (fonction identité) :

$$f(a) = a \quad (111)$$

Avec

$$a = \sum_{i=1}^n X_{ki} W_{ji}(k) \quad (112)$$

Au niveau de la couche de sortie, l'erreur entre la sortie désirée T_k et la sortie réelle Y_k est calculée par :

$$E_k = T_k - Y_k \quad (113)$$

La règle de Widrow-Hoff [58] (équation (114)) peut être utilisée pour rapprocher la valeur de Y_k à la valeur T_k , c'est-à-dire minimiser l'erreur entre la sortie désirée T_k et la sortie réelle Y_k

$$W_{ji}(t+1) = W_{ji}(t) + \eta.(T_j - Y_j)X_i \quad (114)$$

Où :

$W_{ji}(t+1)$: poids à l'instant $t+1$

$W_{ji}(t)$: poids à l'instant t

η : gain d'apprentissage, $0 < \eta < 1$

7- Conclusion

Dans ce chapitre on a cité trois approches connues par leurs efficacités qui peuvent être utilisées dans la phase de reconnaissance.

La première approche est la programmation dynamique dont l'objectif est de rechercher le chemin optimal.

La deuxième utilise la notion des réseaux des neurones inspirée du système nerveux biologique.

La dernière méthode concerne les mémoires associatives qui ont une relation étroite avec les réseaux de neurones.

1- Introduction :

L'objectif de ce chapitre est de présenter le système de reconnaissance adopté (Fig.30) pour la reconnaissance des caractères Tifinaghe, ainsi que les résultats trouvés dans chaque phase de ce système. Aussi, il démontre d'autres traitements qui sont utilisés afin de résoudre quelques problèmes rencontrés dans les phases : prétraitement et extraction d'attributs.

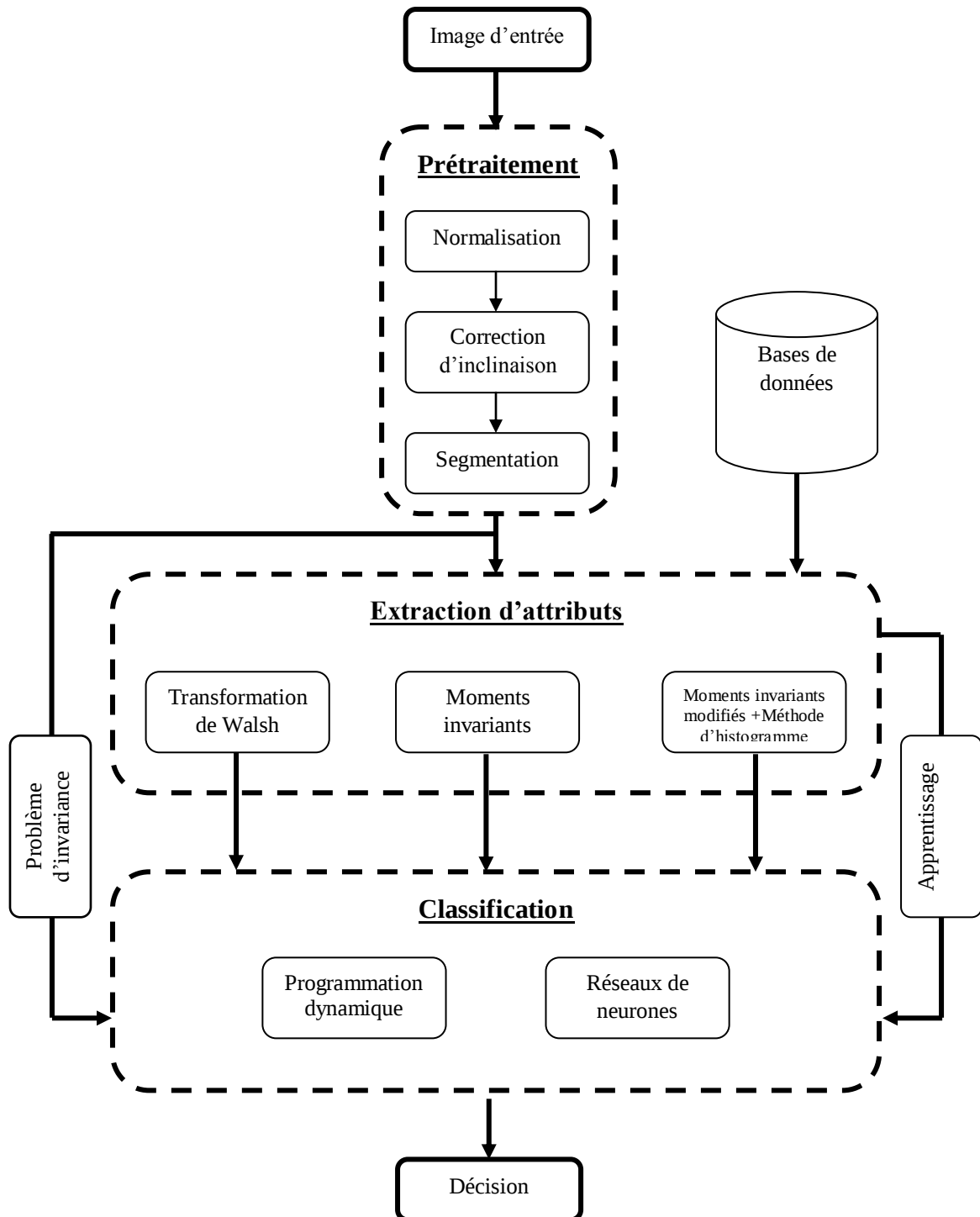


Fig. 30 : Système de reconnaissance

2- Prétraitement

L'image acquise par un scanner est soumise à un prétraitement qui est constitué de trois étapes : normalisation, correction de l'inclinaison et segmentation.

2-1- Normalisation

L'objectif principal de la normalisation est de supprimer les zones inutiles de l'image. Cette opération peut être effectuée à l'aide de deux méthodes :

➤ *Méthode 1*

L'opération de la première méthode consiste à balayer l'image dans quatre directions : de haut en bas et du bas en haut respectivement jusqu'à la rencontre des premiers pixels noirs, de gauche à droite et de droite à gauche respectivement jusqu'à la rencontre des premiers pixels noirs. Après l'obtention des positions des premiers pixels noirs, des lignes sont tracées autour de l'image.

➤ *Méthode 2*

Dans cette méthode, premièrement on calcule les histogrammes horizontal et vertical, ensuite, on balaye l'histogramme horizontal dans les deux directions: de haut en bas et du bas en haut respectivement jusqu'à la rencontre des premiers pixels noirs, après, on parcourt l'histogramme vertical dans les deux directions : de gauche à droite et de droite à gauche respectivement jusqu'à la rencontre des premiers pixels noirs. Finalement, après la détermination des positions des premiers pixels noirs, on élimine les zones indésirables.

La figure (Fig.31) montre les résultats obtenus de la fonction de normalisation en utilisant la deuxième méthode des histogrammes.

2-2- Correction d'inclinaison

La fonction de la correction d'inclinaison passe par un ensemble d'étapes ; l'utilisation de la transformation de Hough pour détecter l'angle d'inclinaison en transformant les pixels noirs de l'image en des courbes sinusoïdales dans le plan polaire, ensuite, en cherchant le maximum de courbes passant par un point commun. Ce point qui correspond à une ligne droite dans le plan cartésien, permet, par conséquent, de déterminer l'angle d'inclinaison.

Après l'obtention de l'angle d'inclinaison, on fait une rotation dans le sens approprié (Fig.32).

2-3- Segmentation

La segmentation consiste à segmenter le texte en lignes puis les lignes en caractères.

2-3-1- Segmentation en lignes

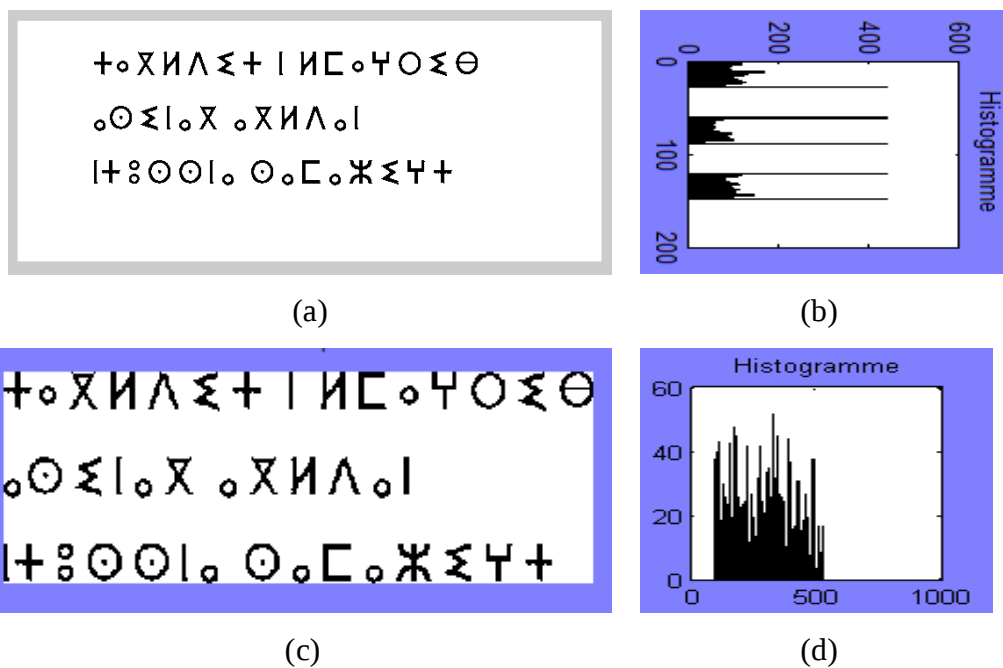
Les lignes d'un texte peuvent être segmentées en utilisant deux méthodes :

➤ Méthode 1

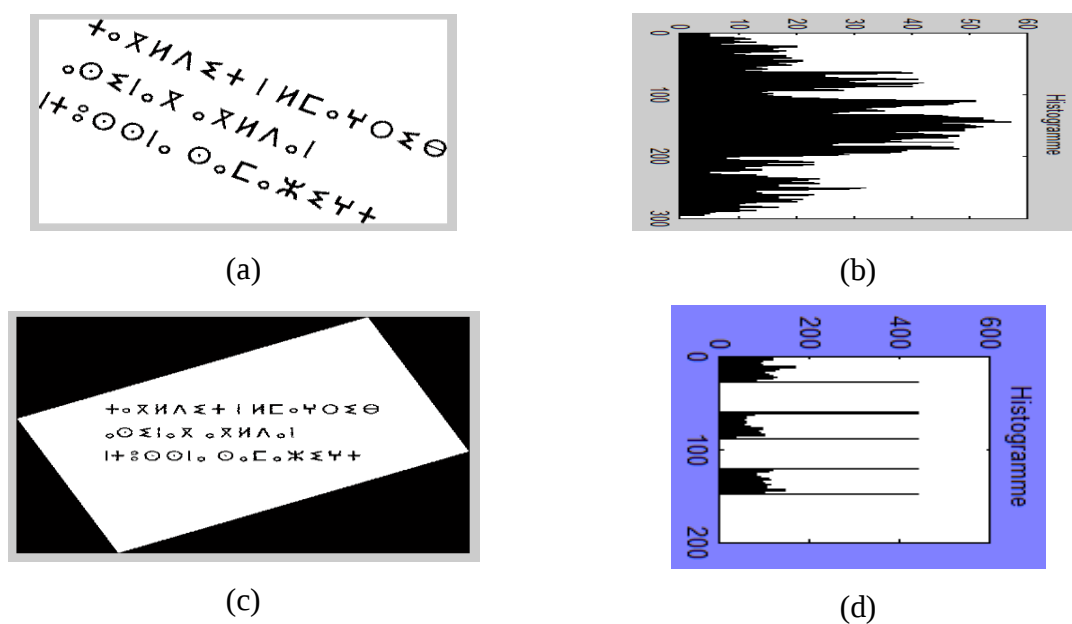
La première consiste à balayer les lignes de l'image de haut en bas jusqu'à la rencontre du premier pixel noir, la ligne contenant ce pixel noir représente le paramètre LH, ensuite, on continue le parcours jusqu'à l'obtention de la première ligne qui ne contient que des pixels blancs. Cette ligne représente le paramètre LB. De la même façon, on poursuit la recherche pour déterminer les autres lignes (Fig.13).

➤ Méthode 2

La deuxième méthode se base sur l'histogramme horizontal pour détecter les lignes du texte. D'abord, nous calculons l'histogramme horizontal de l'image, après, on balaye l'histogramme de haut en bas pour déterminer les paramètres LH et LB (Fig.13).



*Fig. 31 : (a) Avant normalisation, (b) Histogramme horizontal
(c) Après normalisation et (d) Histogramme vertical*



*Fig. 32 : (a) Avant correction, (b) Histogramme horizontal de (a)
(c) Résultat de correction, (d) Histogramme après correction*

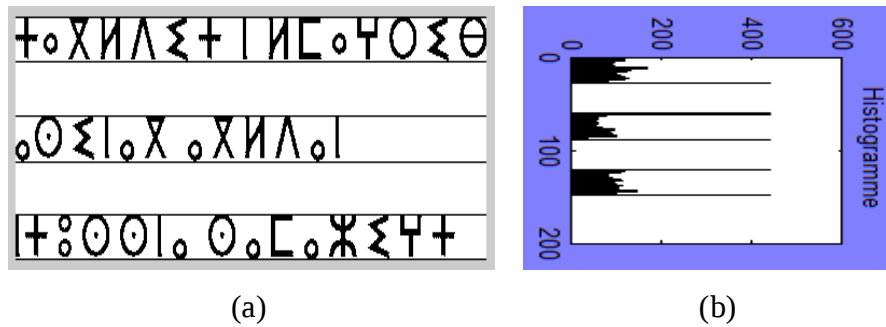


Fig. 33 : (a) Segmentation en lignes et (b) Histogramme horizontal

La figure (Fig.33) indique un texte segmenté en lignes en utilisant la deuxième méthode citée ci-dessus.

2-3-2- Segmentation en caractères

Pour chaque ligne, la segmentation en caractères peut être effectuée par deux méthodes :

➤ Méthode 1

La première a pour principe de parcourir l'image colonne par colonne de gauche à droite jusqu'à la rencontre du premier pixel noir, la colonne contenant ce pixel noir représente le paramètre CG. Ensuite, on continue le parcours jusqu'à l'obtention d'une colonne qui ne contient que des pixels blancs. Cette colonne représente le paramètre CD. De la même façon, les autres caractères sont déterminés.

➤ Méthode 2

La deuxième méthode se base sur l'histogramme vertical pour détecter les caractères. Premièrement, on calcule l'histogramme vertical de l'image, après, on balaye cet histogramme de gauche à droite pour déterminer les paramètres CG et CD.

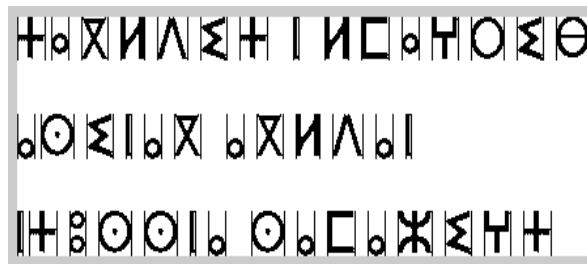


Fig. 34 : Segmentation en caractères

La figure (34) montre un texte segmenté en caractères en se basant sur la méthode d'histogramme.

Lors de la segmentation en caractères, les deux méthodes qu'on peut utiliser rencontrent un problème pour la segmentation de X^u et R^u . Par exemple, elles segmentent le caractère suivant X^u en deux parties, ce qui est faux. Pour résoudre ce problème, on calcule la distance d représentée dans la figure (35). Si $d \leq 5 \text{ pixels}$ (déterminé expérimentalement), on annule la colonne2 et la colonne3 pour obtenir un seul caractère.



Fig. 35: (a) Calcul de la distance d , (b) Colonnes de segmentation

3- Extraction d'attributs

Dans cette phase, on a testé trois approches comme descripteurs invariants : la transformation de Walsh, les moments invariants et les moments invariants modifiés (calculés sur le contour externe du caractère).

3-1- Transformation de Walsh

Le tableau 3 illustre un exemple d'extraction d'attributs, en utilisant la transformation de Walsh, du caractère 16 en quatre représentations.

	Originale	Taille réduite	Miroir	Rotation par 90°
w_i				
w_1	0	0	0	-0.0029
w_2	-0.0029	-0.0029	-0.0029	-0.0059
w_3	-0.0064	-0.0064	-0.0059	-0.0088
w_4	-0.0098	-0.0098	-0.0093	-0.0118
w_5	-0.0132	-0.0132	-0.0127	-0.0137
w_6	-0.0167	-0.0167	-0.0162	-0.0152
w_7	-0.0201	-0.0201	-0.0196	-0.0172

Tableau 3: Transformation de Walsh

3-2- Moments invariants

L'application des moments invariants, sur le caractère 26 représenté en quatre formes, a donné les résultats illustrés dans le tableau 4.

	Originale	Taille réduite	Miroir	Rotation par 90°
$\log(\varphi_i)$				
φ_1	-1.3666	-1.3413	-1.3666	-1.3598
φ_2	-6.6891	-6.5944	-6.6891	-6.7362
φ_3	-7.1587	-7.3184	-7.1587	-7.3915
φ_4	-12.1029	-10.7623	-12.1029	-11.1215
φ_5	-24.2394	-20.1474	-24.2394	-20.4762
φ_6	-16.1004	-14.5595	-16.1004	-14.5128
φ_7	-22.1628	-21.5532	-21.7188	-21.2735

Tableau 4: Moments invariants

3-3- Moments invariants modifiés

L'utilisation de cette méthode pour l'extraction d'attributs se déroule comme suit :

- Extraction du contour (Tableau 6),
- Choix du contour externe fermé,
- Calcul des moments invariants modifiés sur le contour choisi.

Le tableau 5 contient les sept éléments des vecteurs calculés à l'aide des moments invariants modifiés du caractère 29 en quatre représentations.

	Originale	Taille réduite	Miroir	Rotation par 90°
$\log(\varphi_i)$				
φ_1	-0.0229	-0.0311	-0.0229	-0.0229
φ_2	-2.3107	-2.4123	-2.3107	-2.3107
φ_3	-3.4907	-3.5817	-3.4907	-3.4907
φ_4	-5.4983	-5.5664	-5.4983	-5.4983
φ_5	-10.2246	-10.3329	-10.2246	-10.2246
φ_6	-6.8287	-6.9195	-6.8287	-6.8287
φ_7	-10.6272	-10.7364	-10.6272	-10.3445

Tableau 5: Moments invariants modifiés

Numéro du caractère	Caractère	Numéro du caractère	Caractère	Numéro du caractère	Caractère
1		12		23	
2		13		24	
3		14		25	
4		15		26	
5		16		27	
6		17		28	
7		18		29	
8		19		30	
9		20		31	
10		21		32	
11		22		33	

Tableau 6: Contours des caractères Tifinaghe

Les moments invariants modifiés posent un problème pour les caractères Tifinaghe. Ce problème réside dans l'existence des caractères qui ont le même contour externe fermé (Tableau 7) utilisé dans les calculs d'attributs, ce qui provoque des confusions entre les caractères, et par conséquent, il influence sur l'efficacité du système de reconnaissance. Donc, il faut ajouter une autre méthode pour extraire d'autres paramètres permettant la différenciation entre ces caractères. Cette méthode appelée « méthode d'histogramme ».

 ,  et 	 et 
 et 	 et 

Tableau 7: Contours externes

La figure (36) montre quelques caractères Tifinaghe ainsi que leurs contours externes fermés sélectionnés pour calculer les attributs à l'aide des moments invariants modifiés.

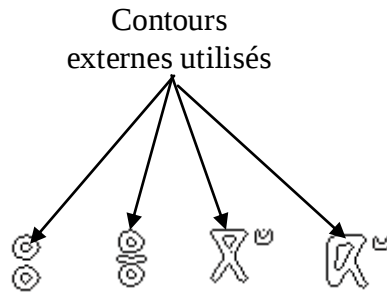


Fig. 36 : Contours externes utilisés pour calculer les moments invariants modifiés

La figure (37) explique le principe adopté pour suivre le chemin du contour tels que :

- (i, j) représente les coordonnées du pixel de départ du contour.
- $\{(i, j-1), (i+1, j-1), (i+1, j), (i+1, j+1), (i, j+1), (i-1, j+1), (i-1, j), (i-1, j-1)\}$ sont les coordonnées des directions de recherche afin de choisir le pixel suivant du contour.

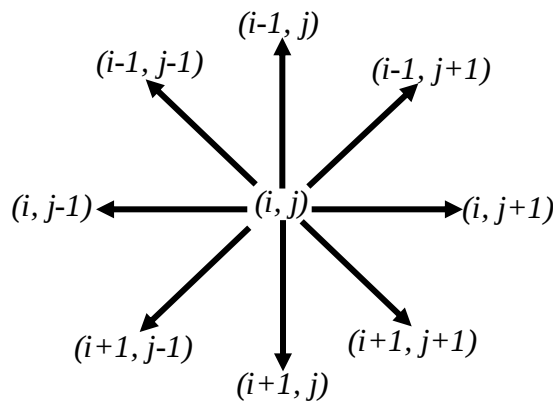


Fig. 37 : Principe utilisé pour suivre le chemin du contour

3-4- Méthode d'histogramme

La méthode d'histogramme est une approche proposée qui permet de résoudre le problème de confusion rencontré lors de l'utilisation des moments invariants modifiés. Cette approche se base sur le calcul des histogrammes horizontal et vertical pour extraire d'autres paramètres (Fig.38).

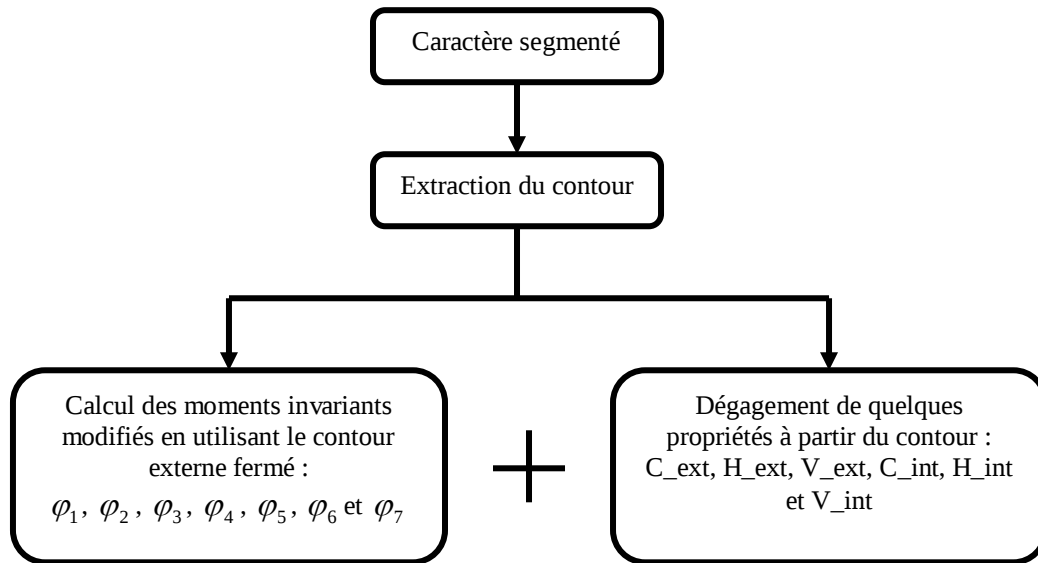


Fig. 38: Organigramme de calcul

Les attributs calculés à l'aide de la méthode d'histogramme sont :

- **C_ext**: Nombre des contours externes
- **H_ext**: Histogramme horizontal utilisé pour calculer le nombre des contours externes
- **V_ext**: Histogramme vertical utilisé pour calculer le nombre des contours externes
- **C_int**: Nombre des contours internes
- **H_int**: Histogramme horizontal utilisé pour calculer le nombre des contours internes
- **V_int**: Histogramme vertical utilisé pour calculer le nombre des contours internes

L'organigramme de la méthode d'histogramme est présenté dans la figure (39).

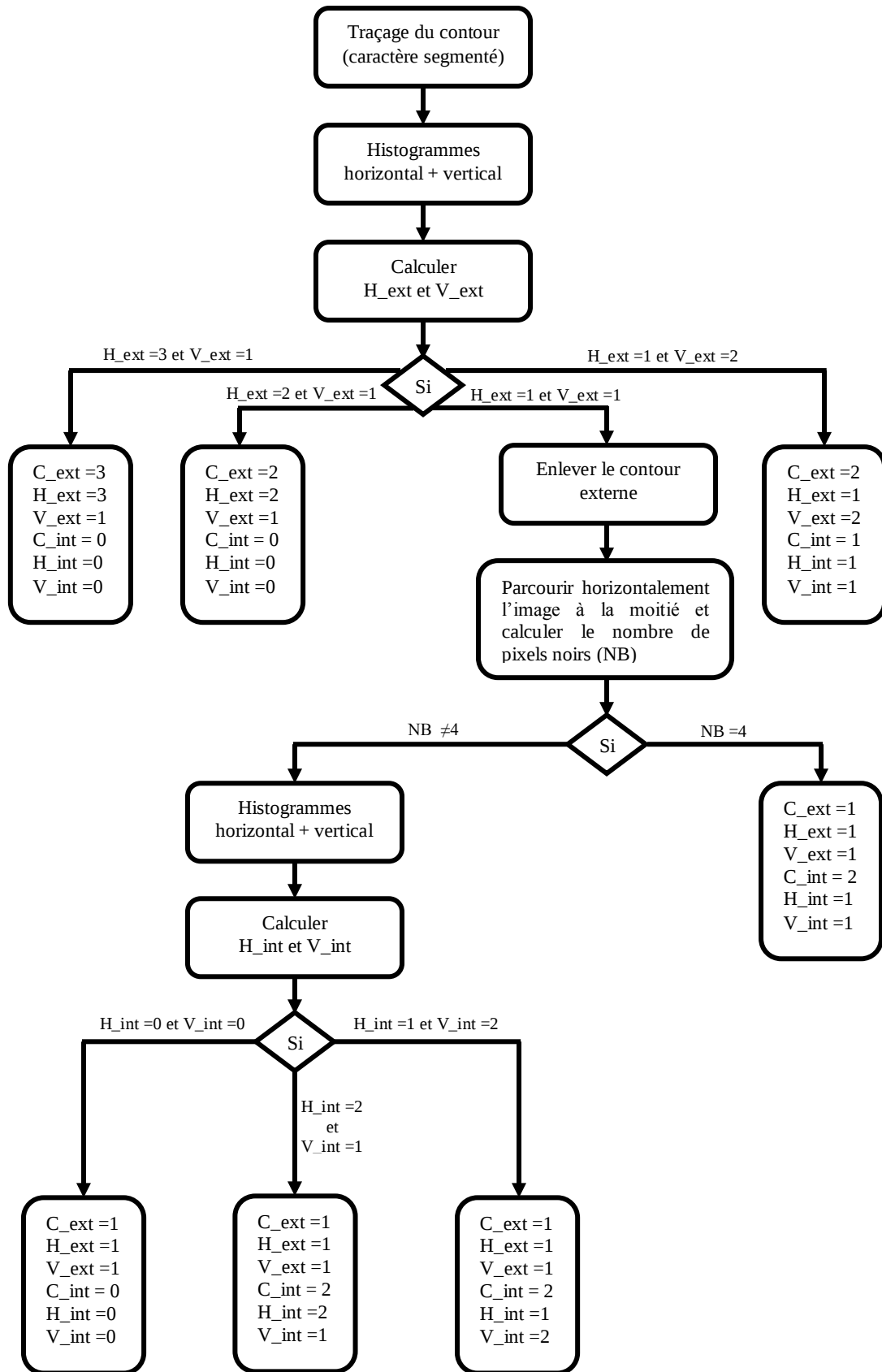


Fig. 39: Organigramme de la méthode d'histogramme

Le tableau 8 montre les propriétés tirées à partir des contours des caractères en utilisant la méthode d'histogramme.

Numéro du caractère	C_ext	H_ext	V_ext	C_int	H_int	V_int
1	1	1	1	1	1	1
2	1	1	1	2	2	1
3	1	1	1	1	1	1
4	2	1	2	1	1	1
5	1	1	1	0	0	0
6	1	1	1	0	0	0
7	3	3	1	0	0	0
8	1	1	1	0	0	0
9	1	1	1	1	1	1
10	2	1	2	1	1	1
11	1	1	1	2	1	2
12	1	1	1	0	0	0
13	1	1	1	0	0	0
14	1	1	1	0	0	0
15	1	1	1	1	1	1
16	1	1	1	0	0	0
17	1	1	1	0	0	0
18	1	1	1	0	0	0
19	1	1	1	0	0	0
20	1	1	1	0	0	0
21	2	2	1	0	0	0
22	1	1	1	1	1	1
23	1	1	1	1	1	1
24	1	1	1	0	0	0
25	1	1	1	2	1	1
26	1	1	1	1	1	1
27	1	1	1	2	2	1
28	1	1	1	0	0	0
29	1	1	1	0	0	0
30	1	1	1	0	0	0
31	1	1	1	0	0	0
32	1	1	1	0	0	0
33	1	1	1	0	0	0

Tableau 8: Propriétés des contours des caractères Tifinaghe

Pour les caractères 7 et 21, on arrête l'exécution du programme au calcul de **C_ext**.

Pour le caractère 25, après la suppression du contour externe fermé, on balaye horizontalement l'image à la moitié et on calcule le nombre de pixels noirs. Si le nombre de pixels égal quatre, alors on déduit que **C_int** égal deux.

Après le calcul de **C_ext**, **H_ext**, **V_ext**, **C_int**, **H_int** et **V_int**, on poursuit les calculs en adoptant les étapes suivantes :

Etape1 :

On affecte la valeur la plus grande de H_{ext} et V_{ext} à C_{ext} .

Etape2 :

Si $H_{ext} > V_{ext}$, alors $H_{ext}=1$ et $V_{ext}=0$,

Si $H_{ext} < V_{ext}$, alors $H_{ext}=0$ et $V_{ext}=1$

Si $H_{ext} = V_{ext}$, alors $H_{ext}=0$ et $V_{ext}=0$

Etape3 :

On attribue la valeur la plus grande de H_{int} et V_{int} à C_{int} .

Etape4 :

Si $H_{int} > V_{int}$, alors $H_{int}=1$ et $V_{int}=0$,

Si $H_{int} < V_{int}$, alors $H_{int}=0$ et $V_{int}=1$

Si $H_{int} = V_{int}$, alors $H_{int}=0$ et $V_{int}=0$

Le tableau 9 indique le tableau résultant.

4- Classification

L'efficacité d'une ROC repose sur la décision signalée par la phase de classification ; c'est la raison du choix de la programmation dynamique et les réseaux de neurones, connus par leur succès, comme approches de classification, ainsi que l'utilisation des descripteurs invariants dans l'extraction d'attributs.

Les descripteurs invariants posent des problèmes lors de la reconnaissance à cause de l'existence des caractères Tifinaghe qui se ressemblent (tableau 10) en quelques situations (translation, rotation et taille).

Pour remédier à ce problème d'invariance, on doit exécuter un autre processus (Fig.40, Fig.41, Fig.42) pour chaque caractère segmenté afin de calculer d'autres variables (SUP, INF, HORIZ et TA) qui vont être intervenir lors de la reconnaissance. Ce processus comporte trois traitements (tableau 11) selon le type de caractères.

Numéro du caractère	C_ext	H_ext	V_ext	C_int	H_int	V_int
1	1	0	0	1	0	0
2	1	0	0	2	1	0
3	1	0	0	1	0	0
4	2	0	1	1	0	0
5	1	0	0	0	0	0
6	1	0	0	0	0	0
7	3	1	0	0	0	0
8	1	0	0	0	0	0
9	1	0	0	1	0	0
10	2	1	0	1	0	0
11	1	0	0	2	0	1
12	1	0	0	0	0	0
13	1	0	0	0	0	0
14	1	0	0	0	0	0
15	1	0	0	1	0	0
16	1	0	0	0	0	0
17	1	0	0	0	0	0
18	1	0	0	0	0	0
19	1	0	0	0	0	0
20	1	0	0	0	0	0
21	2	1	0	0	0	0
22	1	0	0	1	0	0
23	1	0	0	1	0	0
24	1	0	0	0	0	0
25	1	0	0	2	0	0
26	1	0	0	1	0	0
27	1	0	0	2	1	0
28	1	0	0	0	0	0
29	1	0	0	0	0	0
30	1	0	0	0	0	0
31	1	0	0	0	0	0
32	1	0	0	0	0	0
33	1	0	0	0	0	0

Tableau 9: Tableau résultant

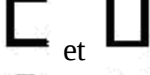
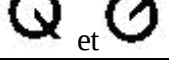
	Taille	Rotation (90°)
Caractères identiques		   

Tableau 10: Caractères identiques

	Caractères
Traitement1	 et   et   et 
Traitement2	 et   et 
Traitement3	 et 

Tableau 11: Traitements 1, 2 et 3

4-1- Traitement 1

Le premier traitement (Fig.40) contient les étapes suivantes:

- Diviser horizontalement l'image en deux parties (SUP et INF).
- Calculer le nombre de pixels noirs dans chaque partie.
- Attribuer 1 au variable SUP et 0 au variable INF si le nombre de pixels noirs dans la partie SUP est supérieur ou égal au nombre de pixels noirs dans la partie INF.
- Attribuer 0 au variable SUP et 1 au variable INF si le nombre de pixels noirs dans la partie SUP est inférieur strictement au nombre de pixels noirs dans la partie INF.

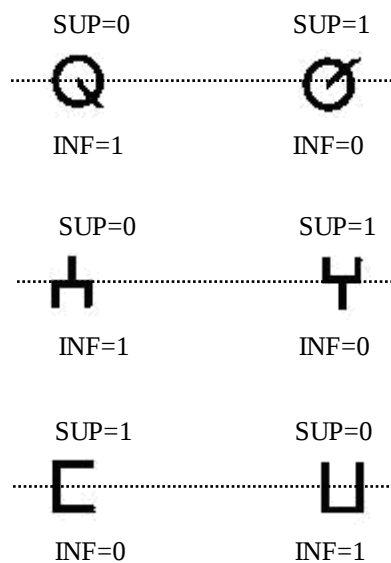


Fig. 40 : Premier traitement

Le tableau suivant indique les valeurs des variables SUP et INF :

<u>Variables</u>	Q	Ø	⌂	¥	□	U
SUP	0	1	0	1	1	0
INF	1	0	1	0	0	1

Tableau 12: Variables SUP et INF

4-2- Traitement 2

Le deuxième traitement (Fig.41) se déroule de la façon suivante :

- Tracer le contour du caractère.
- Enlever le contour externe.
- Calculer l'histogramme horizontal.
- Parcourir l'histogramme, s'il contient deux zones représentant les pixels noirs, alors attribuer 2 au variable HORIZ, et s'il contient une seule zone, alors attribuer 1 au variable HORIZ.

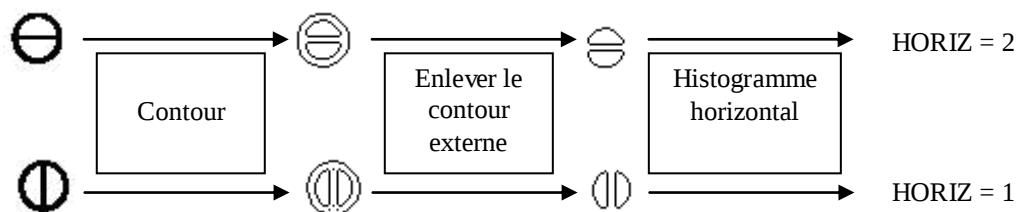


Fig. 41 : Deuxième traitement

4-3- Traitement 3

Le troisième traitement (Fig.42) s'applique comme suit :

- Calculer la longueur L1 du caractère
- Calculer la longueur L2 de la ligne
- Calculer $R = L1 / L2$
- Si $R < 0.5$ alors $TA = 1$, sinon $TA = 0$ (Tableau 13)

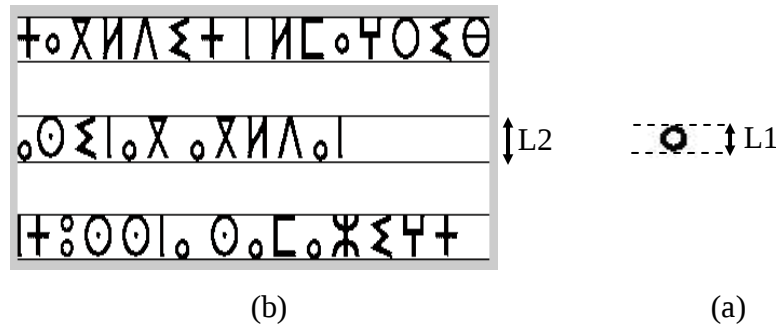


Fig. 42 : (b) Calcul de L2 et (a) Calcul de L1

<u>Variable</u>		
TA	1	0

Tableau 13: Variable TA

4-4- Résultats et remarques

Cette section présente les résultats expérimentaux obtenus par l'utilisation de la transformation de Walsh, les moments invariants et les moments invariants modifiés dans la phase d'extraction d'attributs, ainsi que la programmation dynamique et les réseaux de neurones dans la classification.

Le tableau 14 indique le Taux de Reconnaissance (TR) et le Taux d'Erreur (TE) calculés pour les différentes approches adoptées pour une base de données de 627 caractères avec 594 caractères de test (tableau 17).

Méthodes d'extraction	Programmation dynamique		Réseaux de neurones	
	TR	TE	TR	TE
Méth. 1 : Moments invariants + Traitements (1, 2 et 3)	92.76%	7.24%	93.72%	6.28%
Méth. 2 : Moments invariants modifiés + Méthode d'histogramme	94.44%	5.56%	95.28%	4.72%
Méth.3 : Transformation de Walsh + Traitements (1, 2 et 3)	92.59%	7.41%	94.10%	5.90%

Tableau 14: Taux de reconnaissance et taux d'erreur

Dans le cas des réseaux de neurones, le taux d'apprentissage utilisé est : $\eta = 0.9$.

Un aperçu sur les tableaux 18 et 20 montre les erreurs générées par le système de reconnaissance en cas d'adoption des moments invariants ou bien la transformation de Walsh comme méthodes d'extraction d'attributs.

Les erreurs sont dues principalement aux caractères similaires de Tifinaghe en quelques situations (Taille et Rotation) (tableau 11). Ces dernières ont été évitées grâce à l'ajout des traitements (1, 2 et 3). Les résultats obtenus sont donnés dans les tableaux 15, 19 et 21.

Méthodes d'extraction	Réseaux de neurones	
	TR	TE
Moments invariants	88.38%	11.62%
Moments invariants + Traitements (1, 2 et 3)	93.72%	6.28%
Transformation de Walsh	89.22%	10.78%
Transformation de Walsh + Traitements (1, 2 et 3)	94.10%	5.90%

Tableau 15: *Taux de reconnaissance et taux d'erreur dans le cas d'ajout des traitements (1, 2 et 3)*

5- Conclusion

Au cours de ce chapitre, on a présenté les résultats expérimentaux obtenus par l'application des systèmes de reconnaissance décrits précédemment.

Les résultats montrent que :

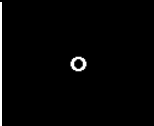
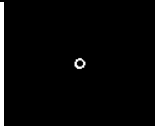
- En adoptant la programmation dynamique, le taux de reconnaissance calculé en utilisant la méthode N°2 est supérieur à ceux trouvés par la méthode N°1 et la méthode N°3. Aussi, le taux d'erreur obtenu par l'application de la méthode N°2 est inférieur à ceux trouvés par la méthode N°1 et la méthode N°3.
- En utilisant les réseaux de neurones, le taux de reconnaissance calculé en appliquant la méthode N°2 est supérieur à ceux calculés par la méthode N°1 et la méthode N°3. Ainsi que le taux d'erreur trouvé par l'utilisation de la méthode N°2 est inférieur à ceux calculés par la méthode N°1 et la méthode N°3.

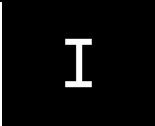
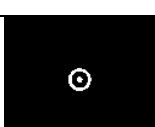
- Les taux de reconnaissance trouvés en adoptant les réseaux de neurones comme méthode de reconnaissance, sont supérieur à ceux calculés dans le cas de la programmation dynamique.

La base de données utilisée est constituée de 627 images [52] dont 33 images représentent la base de référence (tableau 16) et 594 représentent la base de test (tableau 17).

◦	Θ	℄	Λ	E
⌈	ℋ	⌘	⋈	⌘ ^u
⊙	Λ	I	℔	℔ ^u
ℋ	⌈	I	⌘	○
⊙	⊙	⊙	+	E
⌈	℔	⌘	⌘	⌘
⊙	Σ	⊙		

Tableau 16: Base de référence

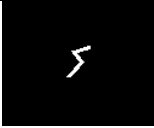
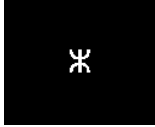
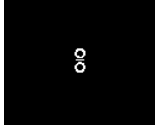
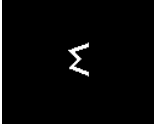
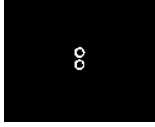
			
			
			
			
			
			
			
			
			

Tableau 17: Base de test

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tableau 18: Matrice de confusion (Moments invariants et R.N)

	35	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	C
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	1	
0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	2	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	12	0	0	3	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	4	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	5	
0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	6	
0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	7	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	8	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	18	0	0	0	0	0	0	0	0	9	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	10	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	11	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	17	0	16	0	0	0	0	0	0	0	0	0	12	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	13	
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	14	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	15	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	19	
0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	21	
0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	22	
0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	
0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	25	
0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	
0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	
0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28	
0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29	
0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	30	
0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	
1	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	

Tableau 19: Matrice de confusion (Moments invariants, Traitements [1, 2 et 3] et R.N)

C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tableau 20: Matrice de confusion (Transformation de Walsh et R.N)

[illegible]

Tableau 21: Matrice de confusion (Transformation de Walsh, Traitements [1, 2 et 3] et R.N)

Dans ce travail, on a abordé le domaine de la reconnaissance des documents Tifinaghe en utilisant deux approches de classification et trois méthodes d'extraction. En général, un système de reconnaissance des caractères Tifinaghe est constitué de trois phases principales :

- Prétraitement
- Extraction d'attributs
- Classification

Après l'acquisition qui consiste à obtenir une image numérisée en utilisant un scanner, on a appliqué le prétraitement qui englobe un ensemble de fonctions : la normalisation, la correction d'inclinaison et la segmentation.

On a commencé par la normalisation afin d'éliminer les zones indésirables dans l'image et de réduire le temps d'exécution.

Ensuite, on a procédé à la correction d'inclinaison en utilisant la transformation de Hough. Finalement, on a adopté la segmentation qui a comme objectif de détecter les lignes dans une image et de segmenter chaque ligne en caractères.

La deuxième phase est l'extraction d'attributs qui consiste à utiliser les descripteurs invariants pour le dégagement des paramètres à partir des caractères segmentés. Les descripteurs testés sont les suivants :

- Les moments invariants
- La transformation de Walsh
- Les moments invariants modifiés

Lors de la classification, on a fait appel à la programmation dynamique et les réseaux de neurones. Pour remédier au problème de confusion entre les caractères similaires par rotation et changement d'échelle, on a fait recours à des traitements qui se basent sur l'histogramme des contours des caractères.

Les résultats obtenus sont :

- En adoptant la programmation dynamique, le taux de reconnaissance calculé en utilisant les moments invariants modifiés (94.44%) est supérieur à ceux trouvés par les moments invariants (92.76%) et la transformation de Walsh (92.59%). Aussi, le taux d'erreur obtenu par l'application des moments invariants modifiés (5.56%) est inférieur à ceux trouvés par les moments invariants (7.24%) et la transformation de Walsh (7.41%).

- En utilisant les réseaux de neurones, le taux de reconnaissance calculé en appliquant les moments invariants modifiés (95.28%) est supérieur à ceux calculés par les moments invariants (93.72%) et la transformation de Walsh (94.10%). Ainsi que le taux d'erreur trouvé par l'utilisation des moments invariants modifiés (4.72%) est inférieur à ceux calculés par les moments invariants (6.28%) et la transformation de Walsh (5.90%).

Malgré le progrès réalisé dans ce travail, concernant les documents Tifinaghe, d'autres problèmes demeurent non résolus. Ces derniers représentent les perspectives à accomplir comme :

- La reconnaissance des caractères manuscrits.
- La reconnaissance des caractères chevauchés.
- La reconnaissance d'un document mixte contenant des images, des textes en Tifinaghe, Latin, Arabe et chiffres.

-
- [1] A. Oulamara et J. Duvernoy, « An application of the Hough transform to automatic recognition of Berber characters », *Signal Processing* », vol.14, no.1, pp.79-90, 1988.
- [2] P.M. Lallican, C. Viarp-Gaudin, S. Knerr : « From off-line to on-line handwriting recognition ». *Proc. 7th workshop on frontiers in handwriting recognition*, pp.303-312, Amsterdam 2000.
- [3] H. Mouchère : « Étude des mécanismes d'adaptation et de rejet pour l'optimisation de classifieurs : Application à la reconnaissance de l'écriture manuscrite en-ligne », Thèse, l'Institut National des Sciences Appliquées de Rennes, 2007.
- [4] M. Blumenstein, C. K. Cheng et X. Y. Liu, 2002, « New Preprocessing Techniques for Handwritten Word Recognition », in *Proceedings of the Second IASTED International Conference on Visualization, Imaging and Image Processing (VIIP 2002)*, ACTA Press, Calgary, pp. 480-484.
- [5] E. Kavallieratou, N. Fakotakis, et G. Kokkinakis, « Skew Angle Estimation In Document Processing Using Cohen's Class Distributions », *PRL(20)*, No. 11-13, November 1999, pp. 1305-1311.
- [6] Wai-Hong Wong, Wan-Chi Siu et Kin-Man Lam, « Generation of moment invariants and their uses for character recognition », *Pattern Recognition Letters* 16 (1995), 115-123 February 1995. © 1995 Elsevier Science.
- [7] A. Bennasri, A. Zahour, B. Taconet : « Extraction des lignes d'un texte manuscrit arabe ». *Vision interface 99*, Trois-Rivières, Canada, 19-21 mai 1999.
- [8] S. H Haitaamar : « Segmentation de textes en caractères pour la reconnaissance optique de l'écriture Arabe ». Thèse de magister, UNIVERSITE EL-HADJ LAKHDHAR BATNA, 2007.
- [9] H. Ali, « A. Back Propagation Neural Network Arabic Characters Classification Module Utilizing Microsoft Word », *Journal of Computer Science* 4 (9): 744-751, 2008.
- [10] S. Ali, K. Al-Omari, P. Sumari, Sadik A. Al-Taweel and Anas J.A. Husain, « Digital Recognition using Neural Network », *Journal of Computer Science* 5 (6): 427-434, 2009.
- [11] F. Slimane, S. Kanoun, J. Hennebert Adel, M. Alimi et R. Ingold, « Modèles de Markov Cachés et Modèle de Longueur pour la Reconnaissance de l'écriture Arabe à Basse Résolution », *MajecSTIC 2009*, Avignon, France, du 16 au 18 novembre 2009.
- [12] A. Sehad, L. Mezai, L. Sekkai et M. Cheriet, « Méthode de détection de l'inclinaison des documents arabes imprimés basée sur la réduction des points et la transformée de Hough », *SETIT 2005 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications* March 27-31, 2005 – TUNISIA.

- [13] R. Al-Hajj, M. Chafic Mokbel et L. Likforman-Sulem, « Reconnaissance de l'écriture arabe cursive : combinaison de classifieurs MMCs à fenêtres orientées », Actes de CIFED'06, pp271-276, Fribourg Suisse, 2006.
- [14] S. Hazmoune, F. Bougamouza et M. Benmohammed, « La reconnaissance automatique de la parole, par combinaison de classifieurs markoviens », JEESI'09, Algerie.
- [15] O. hachour, « Reconnaissance hybride des caractères Arabes imprimés », JEP TALN 2004, Traitement Automatiques de l'Arabe, Fès 20 avril 2004.
- [16] D. Motawa, A. Amin et R. Sabourin, « Segmentation of Arabic Cursive Script », ICDAR'97 Proceedings of the document Analysis and Recognition.
- [17] T.S. El-sheikh et S.G. El-taweel, « Segmentation of handwritten Arabic words », J.K.A.U: Sci. Vol. 5, pp. 109-119 (1413 A.H./ 1993 A.D)
- [18] A. Djematen, B. Taconet., A .Zahour, « Une méthode statistique pour la reconnaissance de caractères berbères manuscrits », Actes de *CIFED'98*, 170-178.
- [19] Y. Es Saady, M. Amrouch, A. Rachidi, M. El Yassa et D. Mammass, « Reconnaissance de Caractères Amazighes Imprimés par le Formalisme des Automates à états finis », SITACAM'09, Agadir-Maroc, 12-13 Décembre 2009.
- [20] M. Fakir, B. Bouikhalene et K. Moro, « Skeletonization Methods Evaluation for the Recognition of Printed Tifinaghe characters », SITACAM'09 Agadir 12-13 Décembre Maroc.
- [21] B. Bouikhalene, M. Fakir, B. El Kessab et S. Safi, « Reconnaissance des Caractères Tifinaghe par l'Utilisation des Réseaux de Neurones Multicouches », SITACAM'09, Agadir-Maroc, 12-13 Décembre 2009.
- [22] Y. Ait Ouguengay, M. Taalabi, « Elaboration d'un réseau de neurones artificiel pour la reconnaissance optique de la graphie amazighe, Phase d'apprentissage », Actes de *SITA'08*, INPT, Maroc, 2008.
- [23] R. El Ayachi, K. Moro, M. Fakir et B. Bouikhalene, « On The Recognition Of Tifinaghe Scripts ». JATIT, vol. 20, No. 2, pp: 61-66, 2010.
- [24] M. Amrouch, A. Rachidi, M. Elyassa, D. Mammass, « Handwritten Amazigh Character Recognition Based On Hidden Markov Models », ICGST-GVIP Journal, Vol.10, Issue 5, pp.11-18, 2010.
- [25] R. El Ayachi, B. Bouikhalene et M. Fakir, « Transformation de Fourier et Moments Invariants Appliqués à la Reconnaissance des Caractères Tifinaghe », SITACAM'11, Agadir-Maroc, 06-07 Mai 2011. pp : 41-54.

- [26] K. Moro, B. El Kessab, M. Fakir, B. Bouikhalene et S. Safi, Contribution à la reconnaissance des caractères Tifinagh par utilisation des réseaux de neurones et la squelettisation, SITACAM'11, Agadir-Maroc, 06-07 Mai 2011. pp : 154-160.
- [27] R. El Ayachi, M. Fakir et B. Bouikhalene, « Recognition of TIFINAGHE Characters Using A multilayer Neural Network », International Journal Of Image Processing (IJIP), vol. 5, Issue2, pp: 109-118, 2011.
- [28] O. Bencharef, M. Fakir, N. Idrissi, B. Bouikhalen et B. Minaoui, « Application de la géométrie riemannienne à la reconnaissance des caractères Tifinaghe », Agadir-Maroc, 06-07 Mai 2011. pp : 179 :188.
- [29] B. El Kessab, C. Daoui, B. Bouikhalene et M. Fakir, « Utilisation des réseaux de neurones et le modèle de Markov pour la reconnaissance des caractères Tifinagh manuscrits », Agadir-Maroc, 06-07 Mai 2011. pp : 31-40.
- [30] M. Fakir, B. Bouikhalene et S. Gounane, « Tifinagh characters recognition using Self Organizing Map and Fuzzy k-Nearest Neighbor », Maroc, 06-07 Mai 2011. pp : 161:169.
- [31] B. Bazdouz, B. Bouikhalene, M. Fakir et R. El Ayachi, «Reconnaissance Automatiques d'un Document Amazighs par les Modèles de Markov cachés », 4^{ème} atelier international sur l'amazighe et les TIC, 2011, IRCAM, Maroc.
- [32] Y. Es Saady, A. Rachidi, M. El Yassa et D. Mammass, « Amazigh Handwritten Character Recognition based on Horizontal and Vertical Centerline of Character », International Journal of Advanced Science and Technology Vol. 33, August, 2011, pp.33-50.
- [33] O. Bencharef, M. Fakir, B. Minaoui et B. Bouikhalene, « Tifinagh Character Recognition Using Geodesic Distances, Decision Trees & Neural Networks », (IJACSA) International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence, 2011, pp. 1-5.
- [34] M. Fakir, O. Bencharef, B. Bouikhalene et B. Minaoui, « Tifinagh Character Recognition Using Riemannian Metric, SVM & Neural Networks », (IJAST) International Journal of Advances in Science and Technology, Vol. 2, No. 6, 2011.
- [35] M. FAKIR : « Reconnaissance des Caractères Arabes Imprimés », Thèse, faculté des sciences semlalia, Maroc, 2001.
- [36] A. W. Senior et A.J. Robinson, « An off-line cursive handwriting recognition system »,on Pattern Analysis and Machine. Intelligence, vol. 20, no. 3, pp. 309-321, 1998
- [37] M. Fakir et al, « Recognition of Arabic printed Scripts by Dynamic Programming Matching Method » IECICE Trans. Inf & Syst, Vol. E76-D, No.2 Feb. 93, pp: 31-37.

- [38] B. Al-Badr et S.A. Mahmoud, « Survey and bibliography of Arabic optical text recognition ». *Signal processing*, vol. 41, pp. 49-77, 1995.
- [39] Casey & Lecolinet : « a survey of methods and strategies in character segmentation ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 7, pp: 690-706 July 1996.
- [40] S. Kermi, « Classifieur neuronal base connaissances, application à la reconnaissance des caractères arabes isolés manuscrits ». Thèse de magister, université Badji Mokhtar, Annaba, Algérie 1999.
- [41] R. O. Dua, « Elements of pattern recognition », dans J.M.Mendel et K.S.Fu (eds), *Adaptive learning and pattern recognition systems*, Academic Press, Ne York, 1970.
- [42] A.J Alnsour et L.M. Alzoubady, « Arabic hand written characters recognized by neocognitron artificial neural network », *J. Pure Appl. Sci.*, 3: 1- 17, 2006.
- [43] J. Anigbogu, « Reconnaissance de textes imprimés multifontes à l'aide de modèles stochastiques et métriques », thèse de doctorat, Université de Nancy I, 1992.
- [44] L. Miclet, « Méthodes structurelles pour la reconnaissance des formes », Eyrolles, 1984.
- [45] M. Amrouch, Y. Es Saady, A. Rachidi, M. Elyassa et D. Mammass, « Printed Amazigh Character Recognition by a Hybrid Approach Based on Hidden Markov Models and The Hough Transform », *ICMCS'09*, Ouarzazate-Maroc, 2009.
- [46] F. Ghorbel, « Application de la Transformée de Fourier Généralisée au Problème de l'Invariance en Reconnaissance de Formes à Niveaux de Gris », *Quatorzième Colloque Gretsi-Juan-Les-Pins*, du 13 au 16 septembre, 1993.
- [47] S. Adam, J.-M. Ogier, C. Cariou, R. Mullot, J. Gardes et Y. Lecourtier, « Utilisation de la transformée de Fourier-Mellin pour la reconnaissance de formes multi-orientées et multi-échelles : application à l'analyse automatique de documents techniques », *Traitement du Signal 2001 – Volume 18 – n° 1*, pp : 17-33, 2001.
- [48] M.-H. Mousa, « Calcul Efficace et Direct des Représentations de Maillages 3D Utilisant les Harmoniques Sphériques », Thèse, Université Claude Bernard – Lyon 1, septembre 2007.
- [49] S. Adam, J.-M. Ogier, C. Cario, R. Mullot, J. Gardes et Y. Lecourtier, « Utilisation de la transformée de Fourier-Mellin pour la reconnaissance de formes multi-orientées et multi-échelles : application à l'analyse automatique des documents techniques », *Traitement du Signal 2001, Volume 18 – n°1*.
- [50] D.W. Ritchie, « Algorithmes Haute-Performance pour la reconnaissance de Formes Moléculaires », *Mémoire scientifique*, Université Henri Poincaré, 2011.

- [51] R. El Ayachi, M. Fakir and B. Bouikhalene « Recognition of Tifinaghe Characters Using Dynamic Programming & Neural Network », chapter in the book "Document Recognition and Understanding", ISBN 978-953-307-995-0, INTECH, 2011.
- [52] W. Abd-Almageed, E. EIKonyaly et S. Saraya, « Point Feature Matching Adopting Walsh transform », SPIE Intelligent Robots and Computer Vision, Pennsylvania, 1997.
- [53] A. Fazekas et A. Hajdu, « Recognizing Typeset Documents using Walsh Transformation », Journal of Computing and Information Technology - CIT 9, 2001, 2, 101–112.
- [54] O. Okun, M. Pietikäinen and J. Sauvola, « Document skew estimation without angle range restriction », IJDAR (1999) 2: 132-144.
- [55] S. Chevalier, E. Geoffrois et F. Prêteux, « A 2D Dynamic Programming Approach for Markov Random Field-based Handwritten Character Recognition », Proceedings IAPR International Conference on Image and Signal Processing (ICISP' 2003), Agadir, Morocco, 2003, p. 617-630.
- [56] C. Tappert, « Cursive Script Recognition by Elastic Matching », *IBM J. Res. Develop.*, Vol. 26, No. 6, 1982, pp. 765-771.
- [57] H. Sakoe et S. Chiba. « Dynamic Programming Algorithm Optimization for Spoken Word Recognition », *IEEE Trans. Acoust., Speech and Signal Processing*, Vol. ASSP-26, No. 1, 1978, pp. 401-408, 1978.
- [58] A. Rebiai, « Une Approche Hybride pour la reconnaissance d'écriture Arabe manuscrite », Thèse de magister, Université de Constantine, Algérie, 2007.
- [59] R. El Ayachi and M. Fakir, « Recognition of Tifinaghe Characters Using Neural Network», 978-1-4244-3757-3/09/\$25.00 ©2009 IEEE.
- [60]. D. Hebb, « the organization of behavior : a neuropsychological theory », 1949.
- [61]. J. Hopfield, « Neural networks and physical systems with emergent collective computational abilities », *Proceedings of the National Academy of Sciences of the USA*, 1982.
- [62] N. Rougier, « Modèles de mémoires pour la navigation autonome », Thèse, Université Henri Poincaré, Nancy, 2000.
- [63] K. Fukushima, « Un Modèle de Mémoire Associative dans le Cerveau », *Kybernetik* 12, 58-63(1973), © by Springer-Verlag 1973.
- [64] A. Rachidi, D. Mammass, « Informatisation de La Langue Amazighe: Méthodes et Mises En Oeuvre », SETIT 2005 3rd International Conference: Sciences of Electronic Technologies of Information and Telecommunications March 27-31, 2005 – Tunisia.

- [65] K. Hallouli, L. Likforman-sulem et M. Sigelle, « Réseaux Bayésiens Dynamiques pour la reconnaissance des caractères imprimés dégradés », GRETSI, Groupe d'Etudes du Traitement du Signal et des Images, 2003.
- [66] M. Outahajala, Y. Benajiba, P. Rosso et L. Zenkour, « POS tagging in Amazighe using tokenization and n-gram character feature set », SITACAM'11 Agadir 06 – 07 Mai 2011 Maroc. pp: 20-30.
- [67] Y. Fujimoto, S. Kadota, S. Hayaehi. M. Yamamoto. S. Yaiima. and M. Yasuda, « Recognition of handprinted characters by nonlinear elastic matching», Proc. 3rd Int. Jt. Conf. Pattern Recognition, pp: 113-118, 1976.
- [68] D. J. Burr, « A Normalizing Transform For Cursive Script Recognition », Proc. 6th Int. J. Conf. on Pattern Recognition Munich (1982), pp. 1027–1030
- [69] B. Chaudhuri, U. Pal et M. Mitra, « Automatic recognition of printed Oriya script », *Sādhanā* Vol. 27, Part 1, February 2002, pp. 23–34. © Printed in India.
- [70] M. Mercimek, K. Gulez et T.V. Mumcu, « Real object recognition using moment invariants », *Sādhanā* Vol. 30, Part 6, December 2005, pp. 765–775. © Printed in India.
- [71] Y. Es Saady, A. Rachidi, M. El Yassa, D. Mammass, « Printed Amazigh Character Recognition by a Syntactic Approach using Finite Automata », ICGST-GVIP Journal, Volume 10, Issue 2, June 2010.
- [72] F. Ghorbel (1993), « Application de la Transformée de Fourier Généralisée au Problème de l'invariance en Reconnaissance de Formes à Niveau de Gris », Quatorzième Colloque GRETSI, JUAN-LES-PINS- du 13 au 16 septembre 1993.
- [73] C.W. Chong, P. Raveendran, et R. Mukundan, « A comparative analysis of algorithms for fast computation of zernike moment », Pattern Recognition, 36:731-742, 2003.
- [74] O. Aubretion, G. Cathebras, B. Lamalle, LFC Lew Yan Voon et C. Lemaitre, « Masques binaires pour reconnaissance de forme », Actes du colloque READ'2005, INT, Evry, France, 1-3 Juin 2005, pg. 83-88.
- [75] G. Abandah et N. Anssari, « Novel Moment Features Extraction for Recognizing Handwritten Arabic Letters », Journal of Computer Science 5 (3): 226-232, 2009.
- [76] H.Z. Shu, H. Zhang, B. J. Chen, P. Haigron, and L.M. Luo, « Fast computation of Tchebichef moments for binary and gray-scale images », published in "IEEE Transactions on Image Processing 2010: epub ahead of print" DOI: 10.1109/TIP.2010.2052276
- [77] Y. Ait ouguengay et Mohamed Taalabi, « Elaboration d'un réseau de neurones artificiels pour la reconnaissance optique de la graphie amazighe : Phase d'apprentissage », 5e

Conférence internationale sur les "Systèmes Intelligents : Théories et Applications", Paris : Europa, cop. 2009 (impr. au Maroc), ISBN 978-2-909285-55-3.

[78]. K. Derdour, « Reconnaissance de formes du chiffre arabe imprimé : Application au code à barre d'un produit », Thèse de magister, Université HADJLAKHDAR –BATNA, Algérie, 2009.

[79] D. W. Ritchie, « Algorithmes Haute-Performance pour la Reconnaissance de Formes Moléculaires », Mémoire Scientifique, École doctorale IAEM Lorraine, UFR STMIA, Université Henri Poincaré, Nancy, 2011.

[80] F. Ghorbel, « A complete invariant description for gray-level images by the harmonic analysis approach », Pattern Recognition Letters 15 (1994) 1043-1051, October 1994.

[81] H. Schaeben et K. Gerald van den Boogaart, « Spherical harmonics in texture analysis », Tectonophysics 370 (2003) 253–268.

[82] S. Bigot-Marchand, « Outils de Traitement d'Images Adaptés au Traitement d'Images Omnidirectionnelles », Thèse, Université de Picardie Jules Verne, Faculté de Mathématiques et d'Informatique, octobre 2008.

[83] M. Valeria Banica, « Equation de Schrödinger en Milieu Inhomogène », Thèse, Université de Paris XI Orsay, Septembre 2003.

[84] Y. Benedic, « Approche Analytique Pour l'Optimisation de Réseaux de Neurones Artificiels », Thèse, Université de Haute-Alsace, décembre 2007.

[85] M. Fakir, M. M. Hassani et C. Sodeyama « On The Recognition Of Arabic Characters Using Hough Transform Technique », Malaysian Journal of Computer Science, Vol. 13 No. 2, December 2000, pp. 39-47.

[86] M. Fakir, O. Bencharefi, B. Bouikhalene and B. Minaoui, « Tifinagh Character Recognition Using Riemannian Metric, SVM & Neural Networks », International Journal of Advances in Science and Technology, Vol. 2, No. 6, 2011, pp. 1-9.

[87] A. Rachidi et D. Mammass, «Informatisation de La Langue Amazighe : Méthodes et Mises En Œuvre », SETIT 2005, 3rd International Conference: Sciences of Electronic Technologies of Information and Telecommunications March 27-31, Tunisia, 2005.

[88] M. Outahajala, Y. Benajiba, P. Rosso et L. Zenkoular, « POS Tagging in Amazighe Using Support Vector Machines and Conditional Random Fields », Natural Language Processing and Information Systems, Lecture Notes in Computer Science, Volume 6716/2011, pp: 238-241, 2011.

- [89] Rachidi, et D. Mammass, « Methods and developments in the creation of a computerised Amazigh script », European Language Resources Association letter's, the ELRA Newsletter, Vol.10 N°1, January-March 2005.
- [90] Rachidi et D. Mammass, « Vers un système de traduction automatique en ligne des documents amazighs fondé sur les graphes UNL », Revue E-TI, ISSN 1114-8802, volume 4, Juin 2007.
- [91] N. Zermi, M. Ramdani et M. Bedda, « Arabic handwriting word recognition based on hybride HMM/ANN approach », International Journal of Soft Computing, Vol. 2, N°. 1, pp. 5-10, 2007.
- [92] A. Benouareth, A. Ennaji et M. Sellami, « Semi-continuous HMMs with explicit state duration for unconstrained Arabic word modeling and recognition », Pattern Recognition Letters, Vol. 29, N°. 12, pp. 742-752, September 2008.
- [93] B. Eastwood, A. Jennings et A. Harvey, « A Feature Based neural Network Segmenter for Handwritten Words », Int. Conf. Computational Intelligence and Multimedia Applications, Gold Coast, Australia, pp. 286-290, 1997.
- [94] Y. Kessentini, T. Paquet et A. Benhamadou, « Off-Line Handwritten Word Recognition Using Multi-Stream Hidden Markov Models », Pattern recognition Letters (PRL), Vol 30, Issue 1, pp. 60-70, January 2010.
- [95] Y. Es Saady, A. Rachidi, M. El Yassa et D. Mammass « Reconnaissance Automatique de l'Écriture Amazighe à base de Ligne Centrale de l'Écriture », 4ème Atelier international sur l'amazighe et les TIC, 2011, IRCAM, Maroc.
- [96] F. Menasri, « Contributions à la reconnaissance de l'écriture arabe manuscrite », Thèse de doctorat, Université paris Descartes, 2008.
- [97] L. Likforman-Sulem, A. Zahour and B.Taconet, « Text line segmentation of historical documents: a survey », IJDAR, vol. 9, no. 2, pages 123–138, 2007.
- [98] A. Vinciarelli, S. Bengio and H. Bunke, « Offline Recognition of Unconstrained Handwritten Texts Using HMMs and Statistical Language Models », IEEE Trans. On PAMI, vol. 26, no. 6, pp. 709–720, 2004.
- [99] H. Xue and V. Govindaraju, « Hidden Markov Models Combining Discrete Symbols and Continuous Attributes in Handwriting Recognition », IEEE Trans. On PAMI, vol. 28, no. 3, pp. 458–462, 2006.
- [100] L.S. Oliveira, M. Morita and R. Sabourin, « Feature Selection for Ensembles Applied to Handwriting Recognition », IJDAR, vol. 18, pages 262–279, 2006.

- [101] W. B. Tao, J. W. Tian and J. Liu,, « Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm », *Pattern Recognition Letters*, vol. 24, no 16, pp. 3069–3078, 2002.
- [102] Y. Es Saady, A. Rachidi, M. El Yassa et D. Mammass, « Une méthode syntaxique pour la reconnaissance de caractères amazighs imprimés », *CARI'08*, Maroc, 27-31 Octobre 2008.
- [103] K.W. Wong, C.S. Leung and S.J. Chang, « Handwritten digit recognition using multi-layer feedforward neural networks with periodic and monotonic activation functions », *ICPR*, vol. 3, pp. 106–109, 2002.
- [104] S. Al-Ma'adeed, « Recognition of off-line handwritten Arabic words using neural network », *Proceeding of GMAI'06*, International Conference on Geometric Modeling and Imaging, pp. 141-114, London, England, July 2006.
- [105] M. Blumenstein, B. Verma and H. Basli, « A novel feature extraction technique for the recognition of segmented handwritten characters », *ICDAR '03*, Edinburgh, Scotland, pp. 137-141, 2003.
- [106] L. Lorigo et V. Govindaraju, « Segmentation and Pre-Recognition of Arabic Handwriting », *ICDAR 05*, Corée du Sud, pp. 605-609, 2005.
- [107] F. Chang, K. H. Liang, T. M. Tan and W. L. Hwang, « Binarization of document images using Hadamard multiresolution analysis », *Proceedings of 5th ICDAR*, Bangalore, India, pp. 157–160, septembre 1999.
- [108] A. Amin, « Recognition of hand-printed characters based on structural description and inductive logic programming », *Pattern Recognition Letters*, Vol. 24, pp. 3187-3196, 2003.
- [109] G. Baptista et K. M. Kulkarni, « A High Accuracy Algorithm for Recognition of handwritten numerals », *Pattern Recognition*, Vol.4, pp.287-291, 1988.
- [110] M. Lemaitre, « Approche markovienne bidimensionnelle d'analyse et de reconnaissance de documents manuscrits », thèse de doctorat, Université Paris 5 René Descartes, 2007.
- [111] Vinciarelli, « A survey on off-line Cursive Word Recognition », *Pattern Recognition*, The journal off pattern recognition society, vol. 35 (7), pp. 1433 - 1446, 2002.
- [112] M. Amrouch Y. Es Saady, A. Rachidi, M. El Yassa et D. Mammass, « Apprentissage Markovien et Neuronal : Cas des Caractères Amazighes Imprimés », *SITACAM'09*, Agadir-Maroc, 12-13 Décembre 2009.
- [113] Y. Es Saady, A. Rachidi, M. El Yassa and D. Mammass, « AMHCD: A Database for Amazigh Handwritten Character Recognition Research », *International Journal of Computer Applications* (0975 – 8887), Volume 27– No.4, August 2011.

- [114] M. Eddahibi, S.Mouhim, C.Cherkaoui et D.Mammass, « Amazighe Transliteration », SITACAM'11 Agadir 06 – 07 Mai 2011, Maroc.
- [115] Jean-Luc Henry, « Une étude sur le choix des caractéristiques pour la représentation de caractères imprimés », Vision Interface '99, Trois-Rivières, Canada, 19-21 May.
- [116] Ji-Hwei Horng and Johnny T. Li, « A dynamic programming approach for fitting digital planar curves with line segments and circular arcs », Journal Patrec Article No. 2686, Pages 01-15, Dispatch 17 October 2000.
- [117] Y. Zhu, L.C. De Silva and C. C. Ko, « Using moment invariants and HMM in facial expression recognition », in Elsevier Pattern Recognition Letters Journal, Vol. 23, No 1, pp: 83-91, January 2002.
- [118] Charles C. Tappert, Ching Y. Suen and Toru Wakahara, « The State of the Art in On-Line Handwriting Recognition », IEEE Transactions On Pattern Analysis And Machine Intelligence. Vol. 12. No 8. August 1990.
- [119] A. Zidouri et M. Sarfraz, « On Optical Character Recognition Of Arabic Text », The 6th Saudi Engineering Conference, KFUPM, Dhahran, December 2002.
- [120] M. Mercimek, K. Gulez and T. Veli Mumcu, « Real object recognition using moment invariants», Sādhanā Vol. 30, Part 6, December 2005, pp. 765–775. © Printed in India.

Le berbère est représenté aujourd'hui par ses variantes parlées par les Berbères. Ces variantes sont présentes depuis le Maroc jusqu'à l'Egypte, en passant par l'Algérie, la Tunisie, le Niger et le Mali. On dénombre une trentaine de variétés.

L'écriture berbère utilise des alphabets appelés Tifinaghe, elle n'est pas absolument unifiée : elle connaît un assez grand nombre de variantes, à travers le temps et l'espace.

1- Variantes

Pour les périodes anciennes, on distingue traditionnellement au moins trois alphabets différents : le libyque occidental, l'oriental, et l'alphabet saharien ou Tifinaghe ancien [35, 43, 44, 45, 64].

Aujourd'hui, il existe également des variantes :

- **Le libyque:** Il s'agit des variétés de Tifinaghe les plus anciennes. Il existe deux formes du libyque, l'occidental utilisé le long de la côte méditerranéenne de la Kabylie jusqu'au Maroc et sans doute aux Îles Canaries. Et la forme orientale a été utilisée dans le Constantinois, en Aurès et en Tunisie.
- **Le Tifinaghe saharien:** Cette variété est également appelée libyco-berbère ou touareg ancien. Elle contient des signes supplémentaires par rapport au libyque, plus particulièrement un trait vertical pour noter la voyelle finale /a/. Cette variété fut utilisée pour transcrire le touareg ancien mais ses inscriptions sont incompréhensibles.
- **Le Tifinaghe touareg:** Il existe au sein du Tifinaghe touareg quelques divergences dans la valeur attribuée aux signes qui correspondent aux variations dialectales touarègues. Si, d'une région à une autre, la forme et le nombre des signes peuvent changer, les textes restent en général mutuellement compréhensibles.
- **Le néotifinaghe:** Le néotifinaghe désigne les systèmes d'écriture développés pour représenter les parlers Amazighes du Maghreb. La première variante fut celle proposée à la fin des années 60 par l'Académie berbère sur la base de lettres Tifinaghes touarègues. Elle est largement diffusée au Maroc et en Algérie surtout en Kabylie. Ce vocable comprend aussi d'autres variantes venues développer ou pour certaines corriger les quelques imperfections du système de l'Académie berbère.

2- Institut Royal de la Culture Amazighe (IRCAM)

Au Maroc, la population qui parle le berbère est répartie entre plusieurs dialectes, Tarifite au nord, Tamazighte dans le Haut et Moyen Atlas et Tachelhite au centre du pays. Les Berbères marocains emploient également les caractères Tifinaghe comme motifs d'ornement dans l'artisanat (tapis, bijoux, poterie...) et pour décorer l'intérieur des maisons traditionnelles.

L'IRCAM (Institut Royal de la Culture Amazighe) a proposé à l'Organisation de Standardisation Internationale (21/06/2004) l'Alphabet Tifinaghe et ce dernier a été confirmé le 05/07/2004 [64]. Cette proposition comprend quatre sous-ensembles de caractères Tifinaghe:

1. le jeu de base de l'IRCAM ;
2. le jeu étendu de l'IRCAM ;
3. d'autres lettres néotifinaghes en usage ;
4. des lettres touarègues modernes dont l'usage est attesté.

L'alphabet de base adopté par l'IRCAM est constitué de trente trois caractères (Tableau.22).

Une inspection approfondie de l'alphabet Tifinaghe montre qu'il y a des caractères qui s'obtiennent à partir d'autres soit par rotation (exemple : 2 et 11, 13 et 24, 19 et 30), soit par changement d'échelle (exemple : 1 et 22).

Numéro du caractère	Caractère	Numéro du caractère	Caractère	Numéro du caractère	Caractère
1	ⵏ	12	ⵊ	23	ⵓ
2	ⵍ	13	ⵋ	24	ⵔ
3	ⵎ	14	ⵌ	25	ⵕ
4	ⵎ̣	15	ⵍ	26	ⵖ
5	ⵏ	16	ⵎ	27	ⵗ
6	ⵐ	17	ⵏ	28	ⵘ
7	ⵑ	18	ⵐ	29	ⵙ
8	ⵒ	19	ⵑ	30	ⵚ
9	ⵓ	20	ⵒ	31	ⵛ
10	ⵓ̣	21	ⵑ	32	ⵜ
11	ⵔ	22	ⵒ	33	ⵝ

Tableau. 22: Alphabet de base Tifinaghe – IRCAM

L'écriture Tifinaghe a une caractéristique spécifique qui se diffère des écritures Latines et Arabes ; elle n'est jamais cursive, ce qui facilite les opérations de segmentation. Mais, l'existence de quelques caractères qui se ressemblent dans quelques situations (bruit, rotation et taille) poserait des problèmes lors de la reconnaissance.

ⵎⵔⵉⵏⵉⵢⵉ ⵎⵓⵎⵓⵎⵓ

ⵓ ⵓⵓⵓⵓ ⵔⵓⵔ ⵔⵓⵓ, ⵓⵓⵓ ⵔⵓⵓ ⵓⵓⵓⵓ.
 ⵓⵔⵓⵓ ⵓⵓⵓ ⵔⵓⵓⵓⵓ: ⵔⵓⵓⵓ ⵓⵔ ⵓⵓ
 ⵔⵓⵓⵓ ⵓⵓ ⵔⵓⵓⵓⵓⵓ, ⵔⵓⵓⵓ ⵓⵓⵓ ⵓⵓⵓⵓ ⵓⵓⵓ
 ⵓⵓⵓⵓⵓⵓ !

ⵓⵓⵓ ⵔⵓⵓⵓⵓ ⵓ. ⵓⵓⵓⵓ ⵔⵓⵓ ⵓⵓⵓⵓⵓ. ⵓⵔⵓⵓ
 ⵔⵓⵓⵓⵓ ⵓⵓⵓⵓ, ⵓⵓⵓⵓⵓⵓ, ⵓⵓⵓⵓⵓ, ⵓⵓⵓⵓⵓ –
 ⵔⵓⵓⵓⵓ ⵓⵔ ⵓⵓⵓⵓⵓ. ⵔⵓⵓⵓⵓⵓ – ⵓⵓⵓⵓⵓ ⵓⵔⵓⵓⵓ
 ⵓⵓⵓⵓ ⵓⵓⵓⵓⵓⵓⵓⵓⵓ, ⵓⵓⵓⵓⵓ ⵓⵓ ⵓⵓⵓⵓⵓⵓ ⵓⵓ ⵓⵓⵓⵓ
 ⵓⵓ ⵓⵓⵓⵓⵓⵓⵓⵓⵓⵓ. ⵓⵓⵓⵓⵓ ⵓⵓⵓⵓⵓ ⵓⵓⵓⵓⵓⵓⵓⵓ
 ⵓⵓⵓ ⵓⵔⵓⵓⵓⵓ, ⵓⵓⵓ ⵓⵓⵓ ⵔⵓⵓⵓⵓⵓ ⵓⵓ ⵓⵔⵓⵓⵓ ⵓⵓ
 ⵓⵓⵓⵓⵓⵓ. ⵓⵓⵓ ⵓⵓⵓ ⵔⵓⵓⵓⵓⵓ ⵓⵓ ⵓⵓⵓⵓⵓⵓⵓⵓ :
 « ⵓⵓⵓⵓⵓⵓⵓ 3251 ».

Fig. 43 : Exemple de texte en Tifinaghe

La liste de l'alphabet Tifinaghe et le plan Unicode associé attribuée par l'ISO [20] est illustré à la figure (Fig.44).

ANNEXE : CARACTERISTIQUES DES CARACTERES TIFINAGHE

Code informatique	Caractère	Code informatique	Caractère	Code informatique	Caractère	Code informatique	Caractère	Code informatique	Caractère
2D30	ⵏ	2D40	ⵐ	2D50	ⵑ	2D60	ⵒ	2D70	
2D31	ⵐ	2D41	ⵑ	2D51	ⵒ	2D61	ⵓ	2D71	
2D32	ⵓ	2D42	ⵔ	2D52	ⵕ	2D62	ⵖ	2D72	
2D33	ⵕ	2D43	ⵖ	2D53	ⵗ	2D63	ⵘ	2D73	
2D34	ⵗ	2D44	ⵘ	2D54	ⵙ	2D64	ⵚ	2D74	
2D35	ⵙ	2D45	ⵚ	2D55	ⵛ	2D65	ⵜ	2D75	
2D36	ⵛ	2D46	ⵜ	2D56	ⵝ	2D66		2D76	
2D37	ⵝ	2D47	ⵞ	2D57	ⵟ	2D67		2D77	
2D38	ⵟ	2D48	ⵠ	2D58	ⵡ	2D68		2D78	
2D39	ⵡ	2D49	ⵢ	2D59	ⵣ	2D69		2D79	
2D3A	ⵢ	2D4A	ⵣ	2D5A	ⵤ	2D6A		2D7A	
2D3B	ⵤ	2D4B	ⵥ	2D5B	ⵦ	2D6B		2D7B	
2D3C	ⵦ	2D4C	ⵧ	2D5C	⵨	2D6C		2D7C	
2D3D	⵨	2D4D	⵩	2D5D	⵪	2D6D		2D7D	
2D3E	⵪	2D4E	⵫	2D5E	⵬	2D6E		2D7E	
2D3F	⵬	2D4F	⵭	2D5F	⵮	2D6F	ⵯ	2D7F	



Tifinaghe – IRCAM de base



Autres lettres Tifinaghe – IRCAM de base, néotifinaghes et lettres Touarègues modernes attestées



Réservé pour un codage ultérieur

Fig. 44: Plan Unicode associée au Tifinaghe