

Transformation de Fourier et moments invariants appliqués à la reconnaissance des caractères Tifinaghe

Rachid El Ayachi

*Equipe de traitement de l'information et de télécommunications TIT, Facultés des Sciences et Techniques, Université Sultan Moulay Slimane, Béni Mellal, Maroc
rachidieea@yahoo.fr*

Mohamed Fakir

*Equipe de traitement de l'information et de télécommunications TIT, Facultés des Sciences et Techniques, Université Sultan Moulay Slimane, Béni Mellal, Maroc
fakfad@yahoo.fr*

Belaid Bouikhalene

*Equipe de traitement de l'information et de télécommunications TIT, Faculté polydisciplinaire, Université Sultan Moulay Slimane, Béni Mellal, Maroc
bbouikhalene@yahoo.fr*

Résumé

La Reconnaissance des Caractères Optiques OCR est un outil qui vise à donner la possibilité aux ordinateurs de lire les caractères sans intervention humaine. Le problème de l'OCR est celui de faire reconnaître les caractères indépendamment de leur position, rotation et taille. Pour remédier à ce problème, on utilise les descripteurs invariants. Le système développé dans ce travail utilise la transformation de Fourier et les moments invariants pour extraire les attributs, et la programmation dynamique et les réseaux de neurone pour la classification. La reconnaissance est effectuée sur les caractères Tifinaghe imprimés.

Abstract

Optical Character Recognition OCR is a tool that aims to provide opportunities for computers to read characters without human intervention. The objective of OCR is characterization of a character by invariant descriptors in translation, rotation and scaling.

In this paper, the OCR developed use invariant moments and Fourier transform in extraction phase. In the recognition phase, dynamic programming and neural network are adopted. All tests are applied on Tifinaghe printed characters.

Mots-clés

Reconnaissance des Caractères Optiques, transformation de Fourier, moment invariant, programmation dynamique, réseaux de neurones

Keywords

Requirement optical character Recognition OCR, Fourier transform, invariant moment, Dynamic programming, neuronal Network

1. Introduction

La Reconnaissance de Caractères Optiques, ROC (*Optical Character Recognition* OCR) (Fakir, 2001), (Fakir *et al.*, 2000), (El Ayachi *et al.*, 2010), (Amrouch *et al.*, 2010), (Es Saady *et al.*, 2010), (El Ayachi *et al.*, 2011b) est un sujet de recherche courant dans le domaine de la reconnaissance de forme.

Les applications de l'OCR sont nombreuses et s'appliquent à des domaines aussi variés que le guidage automatique de véhicule, la reconnaissance d'objet, la numérisation d'ouvrage, les transactions bancaires.

Dans cet article, l'objectif de l'OCR (Figure 1) est de reconnaître les caractères Tifinaghe (Blumenstein *et al.*, 2002), (Bencharef *et al.*, 2011), (Es Saady *et al.*, 2011), (El Ayachi *et al.*, 2011c) (Figure 2). Le système adopté comporte plusieurs phases : prétraitement, extraction d'attributs et reconnaissance.

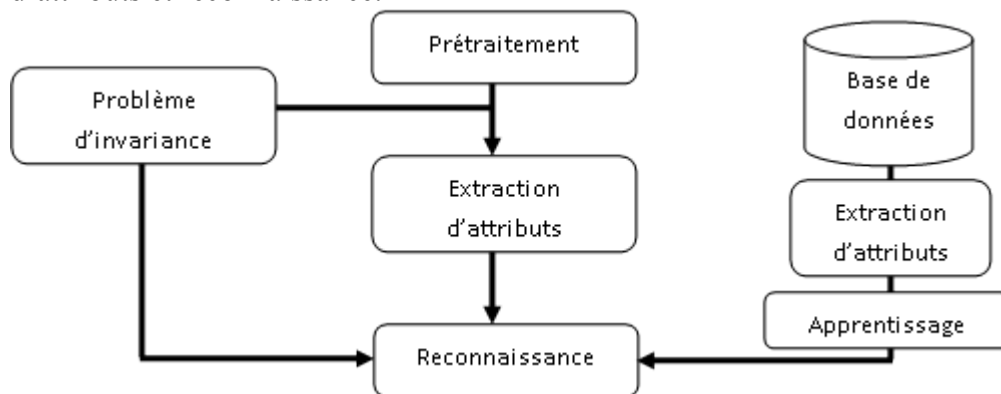


Figure 1. l'OCR

La structure de l'article est organisée de la façon suivante : la section 2 s'intéresse à la phase de prétraitement qui contient plusieurs opérations ; la section 3 représente la phase d'extraction d'attributs en utilisant la transformation de Fourier et les moments invariants, puis la section 4 traite la phase de la reconnaissance à l'aide des réseaux de neurones et la programmation dynamique. La section 5 résout le problème d'invariance alors que la section 6 illustre les résultats obtenus. La section 7 donne les différentes conclusions tirées à partir de ce travail.



Figure 2. Les caractères de Tifinaghe – IRCAM

2. Pré-traitement

Après l'acquisition de l'image, le système de reconnaissance commence par la phase de prétraitement comportant les fonctions suivantes : la normalisation, la correction de l'inclinaison et la segmentation.

2.1. Normalisation

Pour éliminer les zones inutiles dans une image, la méthode d'histogramme (El Ayachi *et al.*, 2010) est utilisée. Le principe repose sur le calcul des histogrammes horizontal et vertical, ensuite, le parcours des deux histogrammes pour détecter les deux lignes L1 et L2 et les deux colonnes C1 et C2 qui permettent de déterminer les marges à supprimer (Figure 3).



Figure 3. Détermination des marges

2.2. Correction d'inclinaison

L'inclinaison (Kavallieratou *et al.*, 1999), (El Ayachi *et al.*, 2011b) (Fig.4) est un phénomène qui influence négativement sur la phase de segmentation et par conséquent sur le système de reconnaissance, il est donc indispensable d'utiliser une méthode efficace pour la correction de l'inclinaison.

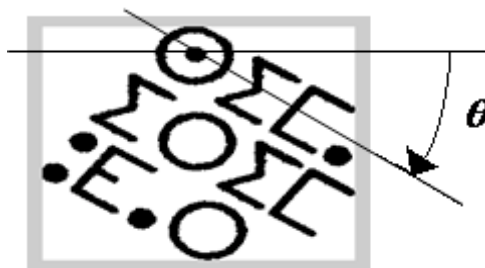


Figure 4. Angle d'inclinaison

La correction d'inclinaison passe par deux étapes : la détection de l'angle d'inclinaison et la rotation dans le sens approprié.

Pour détecter l'angle d'inclinaison, nous avons adopté la transformation de Hough (Fakir *et al.*, 2000), (El Ayachi *et al.*, 2010) qui est connue par robustesse dans ce type de problèmes.

2.3. Segmentation

La fonction de segmentation consiste premièrement à détecter les lignes dans une image contenant un texte en Tifinaghe, et secondement, à segmenter ces lignes en caractères.

Aussi, dans cette fonction de segmentation, nous avons utilisé la méthode d'histogramme (El Ayachi *et al.*, 2010) : le balayage de l'histogramme horizontal de haut en bas pour déterminer les lignes (à l'aide des paramètres LH et LB), le parcours de l'histogramme vertical de gauche à droite afin de segmenter les lignes en caractères (en utilisant les paramètres CG et CD).

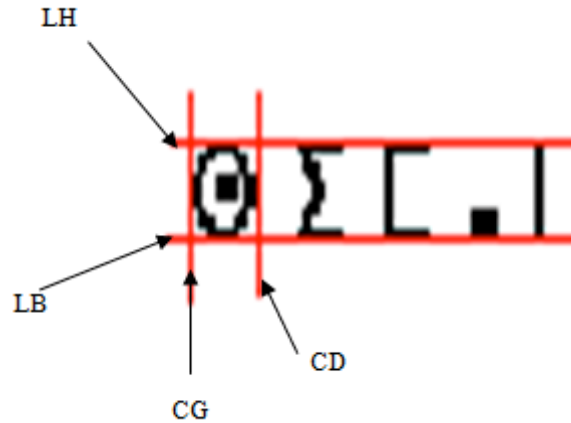


Figure 5. Détection des lignes et caractères

Cette section a traité les différentes fonctions de prétraitement à savoir : la normalisation, la correction d'inclinaison et la segmentation. La section suivante s'intéresse à la phase d'extraction d'attributs.

3. Extraction d'attributs

L'extraction d'attributs est la deuxième phase à appliquer dans un OCR, elle joue un rôle primordial dans la reconnaissance, car elle doit prendre en considération la représentation du caractère dans diverses situations telles que : la translation, la rotation et l'échelle. C'est la raison de l'utilisation des moments invariants (Fakir, 2001), (El Ayachi *et al.*, 2010), (El Ayachi *et al.*, 2011b) et de la transformation de Fourier (Ghorbel, 1993), (El Ayachi *et al.*, 2011a) dans le système traité dans ce travail.

3.1. Moments invariants

Soit f une fonction définie par : $f(x, y) = 1$ sur une région R fermée et délimitée et $f(x, y) = 0$ ailleurs.

On définit le moment d'ordre (p, q) comme suit :

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

Les moments centraux peuvent être exprimés par :

$$\mu_{pq} = \iint_R (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy, \text{ avec } \bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \quad (2)$$

Pour une image numérique, l'équation (2) devient :

$$\mu_{pq} = \sum_{(x,y) \in R} \sum (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (3)$$

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

$$\mu_{00} = m_{00}$$

$$\mu_{10} = 0$$

$$\begin{aligned}
 \mu_{01} &= 0 \\
 \mu_{11} &= m_{11} - \bar{y}m_{10} \\
 \mu_{20} &= m_{20} - \bar{x}m_{10} \\
 \mu_{02} &= m_{02} - \bar{y}m_{01} \\
 \mu_{12} &= m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}^2 m_{10} \\
 \mu_{21} &= m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}^2 m_{01} \\
 \mu_{30} &= m_{30} - 3\bar{x}m_{20} + 2\bar{x}^2 m_{10} \\
 \mu_{03} &= m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2 m_{01}
 \end{aligned} \tag{4}$$

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 :

$$\alpha_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}, \text{ avec } \gamma = \frac{p+q}{2} + 1, \text{ pour } p+q = 2,3,\dots \tag{5}$$

Les invariants du moment suivants ont été obtenus par Hu (1962) et fréquemment utilisés comme caractéristiques pour la reconnaissance des formes.

$$\begin{aligned}
 \varphi_1 &= \alpha_{20} - \alpha_{02} \\
 \varphi_2 &= (\alpha_{20} - \alpha_{02})^2 + 4\alpha_{11}^2 \\
 \varphi_3 &= (\alpha_{30} - \alpha_{12})^2 + (3\alpha_{12} - \alpha_{03})^2 \\
 \varphi_4 &= (\alpha_{30} + \alpha_{12})^2 + (\alpha_{21} + \alpha_{03})^2 \\
 \varphi_5 &= (\alpha_{30} - 3\alpha_{12})(\alpha_{30} + \alpha_{12})[(\alpha_{30} + \alpha_{12})^2 - 3(\alpha_{21} + \alpha_{03})^2] \\
 &+ (3\alpha_{21} - \alpha_{03})(\alpha_{21} + \alpha_{03})[3(\alpha_{30} + \alpha_{12})^2 - (\alpha_{21} + \alpha_{03})^2] \\
 \varphi_6 &= (\alpha_{20} - \alpha_{02})[(\alpha_{30} + \alpha_{12})^2 - (\alpha_{21} + \alpha_{03})^2] \\
 &+ 4\alpha_{11}(\alpha_{30} + \alpha_{12})(\alpha_{21} + \alpha_{03}) \\
 \varphi_7 &= (3\alpha_{21} - \alpha_{03})(\alpha_{30} + \alpha_{12})[(\alpha_{30} + \alpha_{12})^2 - 3(\alpha_{21} + \alpha_{03})^2] \\
 &+ (3\alpha_{12} - \alpha_{03})(\alpha_{21} + \alpha_{03})[3(\alpha_{30} + \alpha_{12})^2 - (\alpha_{21} + \alpha_{03})^2]
 \end{aligned} \tag{6}$$

Hu a montré que ces quantités $\varphi_i, (1 \leq i \leq 7)$ sont invariantes par changement d'échelle, translation et rotation.

3.2. Transformations de Fourier et moments invariants (étude théorique)

Pour $f : IR^2 \rightarrow IR$, sa transformée de Fourier est donnée par :

$$\widehat{f}(u, v) = \int_{-\infty-\infty}^{+\infty+\infty} \int_{-\infty-\infty}^{+\infty+\infty} f(x, y) e^{-i2\pi(xu+yv)} dx dy \quad (7)$$

Cette transformation admet l'approximation discrète :

$$\widehat{f}(u, v) = \sum_x \sum_y f(x, y) e^{-i2\pi(xu+yv)} \quad (8)$$

On définit aussi le moment d'ordre (p, q) par l'équation (1) dont l'approximation discrète est donnée par l'équation :

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad (9)$$

La relation entre les moments et la transformation de Fourier :

Si on calcule

$$\begin{aligned} \frac{\partial^{p+q} \widehat{f}(u, v)}{\partial u^p \partial v^q} &= \int_{-\infty-\infty}^{+\infty+\infty} \int_{-\infty-\infty}^{+\infty+\infty} (-i2\pi x)^p (-i2\pi y)^q f(x, y) e^{-i2\pi(xu+yv)} dx dy \\ &= (-i2\pi)^{p+q} \int_{-\infty-\infty}^{+\infty+\infty} \int_{-\infty-\infty}^{+\infty+\infty} x^p y^q f(x, y) e^{-i2\pi(xu+yv)} dx dy \end{aligned} \quad (10)$$

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

On obtient

$$\begin{aligned} \frac{\partial^{p+q} \widehat{f}(0, 0)}{\partial u^p \partial v^q} &= (-i2\pi)^{p+q} \int_{-\infty-\infty}^{+\infty+\infty} \int_{-\infty-\infty}^{+\infty+\infty} x^p y^q f(x, y) dx dy \\ &= (-i2\pi)^{p+q} m_{pq} \end{aligned} \quad (11)$$

Alors,

$$m_{pq} = \frac{\partial^{p+q} \widehat{f}(0, 0)}{(-i2\pi)^{p+q}} = Fe \left[x^p y^q f(x, y) \right]_{(0,0)} \quad (12)$$

D'où,

$$\begin{aligned} \frac{\partial^{p+q} \widehat{f}(u, v)}{\partial u^p \partial v^q} &= Fe \left[(-i2\pi)^{p+q} x^p y^q f(x, y) \right] \\ &= \sum_x \sum_y (-i2\pi)^{p+q} x^p y^q f(x, y) e^{-i2\pi(xu+yv)} \\ &= (-i2\pi)^{p+q} \sum_x \sum_y x^p y^q f(x, y) e^{-i2\pi(xu+yv)} \end{aligned} \quad (13)$$

On a :

$$\begin{aligned} m_{00} &= \widehat{f}(0, 0) \\ m_{10} &= \frac{\partial \widehat{f}(0, 0)}{\partial u} = Fe \left[x f(x, y) \right]_{(0,0)} \end{aligned}$$

$$\begin{aligned}
m_{01} &= \frac{\partial \widehat{f}(0,0)}{(-i2\pi)} = Fe[yf(x, y)]_{(0,0)} \\
m_{11} &= \frac{\partial^2 \widehat{f}(0,0)}{(-i2\pi)^2} = Fe[xyf(x, y)]_{(0,0)} \\
m_{02} &= \frac{\partial^2 \widehat{f}(0,0)}{(-i2\pi)^2} = Fe[y^2 f(x, y)]_{(0,0)} \\
m_{12} &= \frac{\partial^3 \widehat{f}(0,0)}{(-i2\pi)^3} = Fe[xy^2 f(x, y)]_{(0,0)} \\
m_{20} &= \frac{\partial^2 \widehat{f}(0,0)}{(-i2\pi)^2} = Fe[x^2 f(x, y)]_{(0,0)} \\
m_{21} &= \frac{\partial^3 \widehat{f}(0,0)}{(-i2\pi)^3} = Fe[x^2 yf(x, y)]_{(0,0)} \\
m_{03} &= \frac{\partial^3 \widehat{f}(0,0)}{(-i2\pi)^3} = Fe[y^3 f(x, y)]_{(0,0)} \\
m_{30} &= \frac{\partial^3 \widehat{f}(0,0)}{(-i2\pi)^3} = Fe[x^3 f(x, y)]_{(0,0)}
\end{aligned} \tag{14}$$

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

Dans cette section, nous avons détaillé une étude théorique qui démontre la possibilité d'utiliser la transformation de Fourier et ses dérivées pour calculer les moments invariants d'un caractère segmenté. La prochaine section présente les méthodes de classification.

4. Reconnaissance

La reconnaissance (Al-Hajj *et al.*, 2006), (Outahajala *et al.*, 2011 ; Kessentini *et al.*, 2010), (Ali, 2008), (Ali *et al.*, 2009) est la tâche la plus délicate dans un OCR, car la réussite de ce système de reconnaissance repose sur la décision résultante. Il est donc nécessaire d'utiliser un processus de classification efficace qui a un taux de reconnaissance élevé et par conséquent un taux d'erreur diminué.

Dans cette étape, nous avons utilisé les deux méthodes suivantes : le réseau de neurones (Bouikhalene *et al.*, 2009), (El Ayachi *et al.*, 2010), (El Ayachi *et al.*, 2011b), (El Ayachi *et al.*, 2011c) et la programmation dynamique (El Ayachi *et al.*, 2010), (Chevalier *et al.*, 2003)

pour la reconnaissance des caractères Tifinaghe, parce qu'elles sont connues par leur succès dans le domaine de la reconnaissance des caractères.

4.1. Réseau de neurones

La figure 6 représente l'exemple de réseau de neurones utilisé, à savoir un réseau multi-couches qui contient une couche cachée.

Cet exemple de réseaux comporte :

- Une couche d'entrée à 7 cellules d'entrées (vecteur du moment invariant)
 $E_i = X_i$
- Une couche cachée de 3 neurones d'activations Y_j
- Une couche de sortie de 6 neurones d'activations Z_k
- 7×3 connexions entre la couche d'entrée et la couche cachée, chacune pondérée par V_{ji}
- 3×6 connexions entre la couche cachée et la couche de sortie, chacune pondérée par W_{kj}
- X_0 et Y_0 sont des scalaires

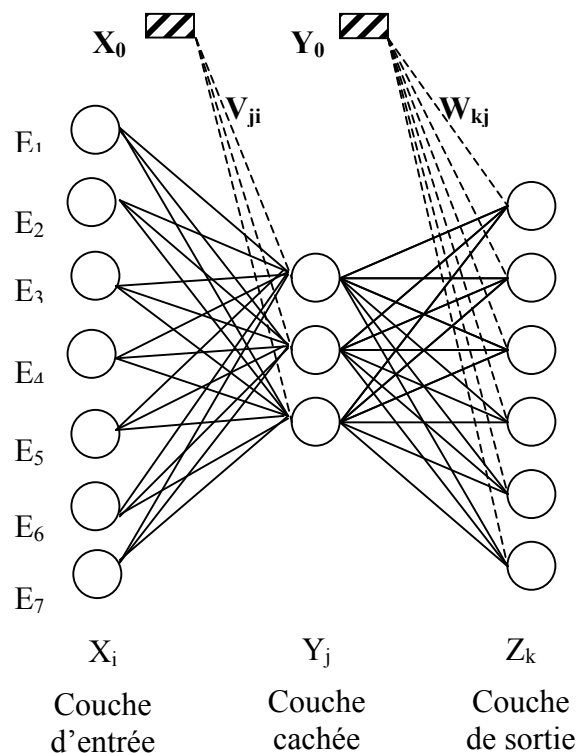


Figure 6. Réseau de neurones

Le principe de fonctionnement du réseau de neurones (Figure 6) repose sur cinq étapes.

Étape 1. Initialisation des poids des connexions.

Les poids sont pris aléatoirement.

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 :

$$Y_j = f\left(\sum_{i=1}^7 X_i V_{ji} + X_0\right) \quad (15)$$

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

$$Z_k = f\left(\sum_{j=1}^3 Y_j W_{kj} + Y_0\right) \quad (16)$$

X_0 et Y_0 sont des scalaires

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

$$f(a) = \frac{1}{1 + \exp(-a)} \quad (17)$$

Etape 3. Rétro-propagation 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 :

$$E_k = Z_k (1 - Z_k) (S_k - Z_k) \quad (18)$$

L'erreur calculée est propagée sur la couche cachée en utilisant la formule 19.

$$F_j = Y_j (1 - Y_j) \sum_{k=1}^6 W_{kj} \cdot E_k \quad (19)$$

Etape 4. Correction des poids des connexions.

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

$$\Delta V_{ji} = \eta \cdot X_i \cdot F_j \quad \text{et} \quad \Delta Y_0 = \eta \cdot F_j \quad (20)$$

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

$$\Delta W_{kj} = \eta \cdot Y_j \cdot E_k \quad \text{et} \quad \Delta X_0 = \eta \cdot E_k \quad (21)$$

η Un paramètre à déterminer empiriquement.

Etape 5. Boucle.

Boucler à l'étape 2 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 :

$$d(t_k, o) = \left(\sum_{i=1}^6 (t_{ki} - o_i)^2 \right)^{1/2} \quad (22)$$

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

4.2. Programmation dynamique

La programmation dynamique est la deuxième méthode utilisée dans la reconnaissance. Elle est organisée en trois étapes.

Etape 1. Calculer la matrice d entre le vecteur du caractère segmenté V_{car} et chaque vecteur des caractères Tifinaghe de référence V_{ref}

La matrice est donnée par :

$$d(x, y) = |V_{car}(x) - V_{ref}(y)| \quad (23)$$

Avec $x, y = 1, 2, \dots, 7$

Etape 2. Calculer le chemin optimal à partir du point (1,1) jusqu'à point (x, y) en utilisant la formule récursive suivante :

$$S(x, y) = d(x, y) + \min \begin{cases} S(x-1, y) \\ S(x-1, y-1) \\ S(x, y-1) \end{cases} \quad (24)$$

Avec $S(x, y)$ est la distance cumulée le long du chemin optimal à partir du point (1,1) jusqu'au point (x, y) .

$S(x, y)$ est évaluée dans l'espace $[1, 7] * [7, 1]$ qui est parcouru colonne par colonne et ligne par ligne à partir du point (1,1).

Etape 3, Calculer l'indice de dissemblance :

$$D(V_{car}, V_{ref}) = \frac{S(7, 7)}{7} \quad (25)$$

5. Problème d'invariance

Un OCR efficace est un système de reconnaissance qui prend en considération l'invariance du caractère en quelques situations (translation, rotation et taille). Cependant, pour les caractères Tifinaghe, cette caractéristique d'invariance pose un problème dans la phase de reconnaissance, car il y en a des caractères qui se ressemblent (\ominus et \oplus , \mathcal{Q} et \mathcal{O} , \mathcal{H} et \mathcal{Y}). Donc, il faut ajouter un autre traitement pour régler ce problème. Ce qui est l'objectif de cette section, ou bien, dans la phase d'extraction d'attributs, il faut utiliser une autre méthode variante.

Pour cela, nous allons enregistrer le caractère segmenté puis exécuter le processus pour calculer d'autres variables (SUP, INF et HORIZ) qui vont intervenir lors de la reconnaissance. Ce processus comporte deux traitements : le premier pour les caractères (\mathcal{Q} et \mathcal{O}) et (\mathcal{H} et \mathcal{Y}) et le deuxième pour les caractères (\ominus et \oplus).

Le premier traitement (Figure 7) comprend les étapes suivantes :

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

- Attribuer 0 à la variable SUP et 1 à la variable INF si le nombre de pixels noirs dans la partie SUP est inférieur au nombre de pixels noirs dans la partie INF.

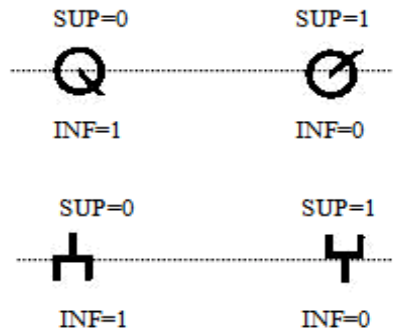


Figure 7. Premier traitement

Le tableau 1 indique les valeurs des variables SUP et INF.

Tableau 1. Les variables SUP et INF

<i>Variables</i>	Q	Ø	H	Y
<i>SUP</i>	0	1	0	1
<i>INF</i>	1	0	1	0

Le deuxième traitement (Figure 8) 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 à la variable HORIZ, et s'il contient une seule zone, alors attribuer 1 à la variable HORIZ.

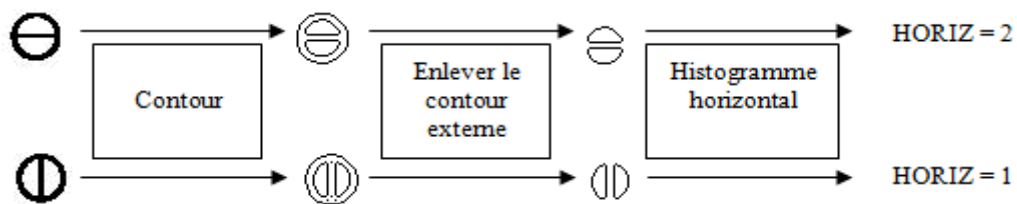


Figure 8. Deuxième traitement

Le tableau 2 indique les valeurs de la variable HORIZ.

Tableau 2. La variable HORIZ

Variable	⊖	⊕
HORIZ	2	1

Cette section a présenté des traitements adoptés pour éviter le problème de ressemblance des caractères Tifinaghe. La section suivante montre les résultats obtenus en utilisant le système de reconnaissance développé.

6. Résultats

L'image de la figure (Figure 9) représente un exemple de texte en Tifinaghe.

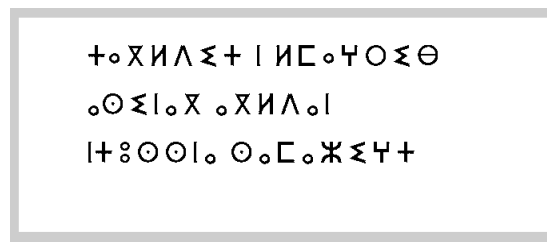


Figure 9. Exemple de texte en Tifinaghe

Après l'application de la phase de prétraitement à la figure 9, nous obtenons les figures 10 et 11.

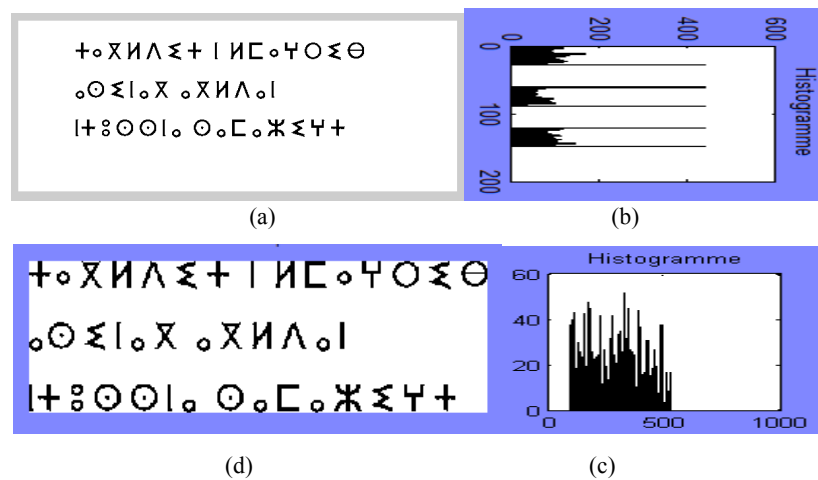


Figure 10. (a) Avant normalisation, (b) Histogramme horis.

(c) Histogramme vertical. (d) Après normalisation

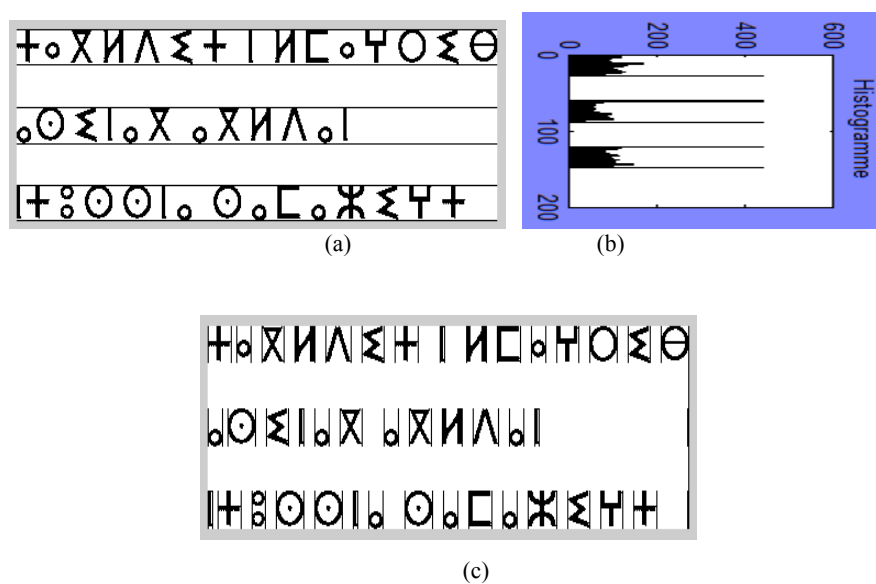


Figure 11. (a) Segmentation en lignes, (b) Histogramme horizontal, (c) Segmentation en caractères

La méthode d'histogramme rencontre un problème dans le cas de la segmentation des caractères (X^u et R^u), elle divise chacun de ces caractères en deux, ce qui est faux, et par conséquent, elle pose une difficulté lors de la reconnaissance. Pour remédier à ce problème, nous faisons recours à la distance d représentée dans la figure (Fig.12). Tout d'abord, nous calculons la distance d , et ensuite, si $d \leq 5 \text{ pixels}$ nous annulons les colonnes 2 et 3 afin d'obtenir un seul caractère.

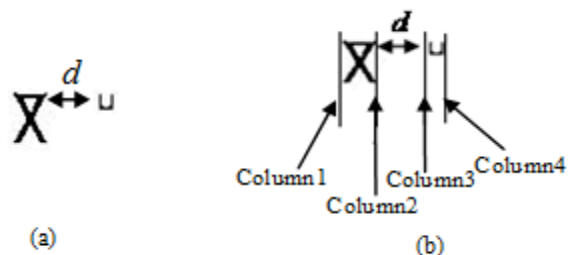


Figure 12. (a) Distance d , (b) Colonnes de segmentation

Dans le cas d'un texte incliné, la fonction de la correction d'inclinaison permet de résoudre le problème, ce qui est illustré en figure 13.

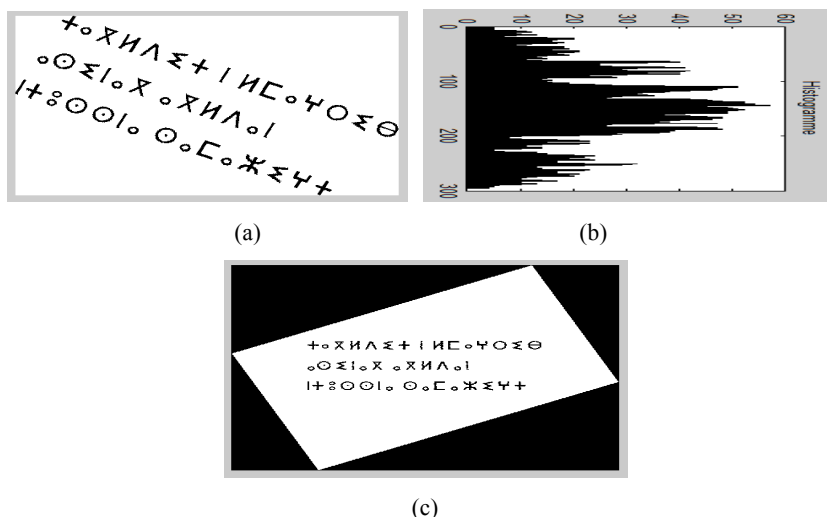


Figure 13. (a) Inclinaison, (b) Histogramme horizontal, (c) Correction d'inclinaison

La comparaison effectuée entre les deux approches appliquées dans la phase de reconnaissance (la programmation dynamique et le réseau de neurones), en se fondant sur des attributs extraits à l'aide de la transformation de Fourier et les moments invariants, est illustrée dans le tableau 3.

Tableau 3. Taux de reconnaissance et taux d'erreur

	Transformation de Fourier et moments invariants	
	Taux de reconnaissance	Taux d'erreur
Programmation dynamique	92.76%	7.24%
Réseau de neurones	93.27%	6.73%

Les résultats du tableau 3 montrent que :

- Le taux de reconnaissance calculé en utilisant le réseau de neurones est supérieur au taux de reconnaissance trouvé par la programmation dynamique.
- Le taux d'erreur du réseau de neurones est inférieur au taux d'erreur de la programmation dynamique.

Dans ce système, la source des erreurs repose sur deux points :

- Problème d'invariance (\ominus et \oplus , \mathcal{Q} et \emptyset , \mathcal{H} et \mathcal{Y}) ie. l'existence de quelques caractères qui sont identiques en quelques situations.
- Le vecteur caractéristique extrait contient uniquement sept éléments (ce qui est insuffisant pour une reconnaissance efficace).

Pour remédier à ce problème source d'erreurs, les solutions que l'on peut proposer sont :

- Dans le cas d'invariance, la correction peut se faire à l'aide de l'utilisation de la méthode détaillée dans la partie (5) de cet article, ou bien, par l'adoption d'une méthode (phase d'extraction) qui n'a pas de problème d'invariance.
- Dans le cas du nombre d'éléments du vecteur caractéristique, on peut utiliser des méthodes qui produisent des vecteurs dont le nombre d'éléments est supérieur à sept, ou bien, regrouper plusieurs méthodes pour obtenir un seul vecteur dont le nombre d'éléments est supérieur à sept.

La base de données (Ait Ouguengay et Taalabi, 2009) utilisée est constituée de 627 images (les caractères imprimés de Tifinaghe). La configuration de l'ordinateur portable utilisée est :

Processor: Intel (R) Core (TM) 2 Duo, CPU T5870 @ 2.00 Ghz, RAM: 2.00 Go.

7. Conclusion

Dans cette recherche, nous avons traité un exemple de système de reconnaissance des caractères Tifinaghe. Ce système est composé d'un ensemble de phases : prétraitement, extraction d'attributs et reconnaissance.

La phase de prétraitement a regroupé plusieurs fonctions à appliquer sur l'image d'entrée ; la fonction de normalisation permettant l'élimination des zones indésirables afin de réduire le temps d'exécution et minimiser l'espace mémoire utilisé, la fonction de correction d'inclinaison qui facilite l'étape suivante concernant la segmentation en lignes et en caractères.

Dans l'extraction d'attributs, nous avons utilisé une méthode qui garde l'invariance du caractère en quelques situations (translation, rotation et taille), appelée moments invariants, ces moments invariants sont calculés à l'aide de la transformation de Fourier. Cette invariance a posé un problème lors de la phase de reconnaissance, ce problème est dû à l'existence des caractères de Tifinaghe identiques (s'ils ont subi une rotation par exemple). Il est résolu par l'exécution d'un traitement en parallèle avec le système de reconnaissance.

La reconnaissance est la phase la plus délicate dans un OCR, car l'efficacité de ce système se base sur la classification trouvée. C'est la raison du le choix du réseau de neurones et programmation dynamique comme deux méthodes de reconnaissance grâce à leurs succès dans ce type de domaine de recherche.

Les résultats obtenus ont montré la réussite de ce système traité dans la reconnaissance des caractères Tifinaghe, c'est-à-dire le succès de l'utilisation de la transformation de Fourier dans le calcul des moments invariants.

Références

- Ait Ouguengay, Y., Taalabi, M. (2009). 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 : Europia, cop. 2009 (impr. au Maroc), ISBN 978-2-909285-55-3.
- Al-Hajj, R., Chafic Mokbel, M. et Likforman-Sulem, L. (2006). Reconnaissance de l'écriture arabe cursive : combinaison de classifieurs MMCs à fenêtres orientées. *Actes de CIFED'06*, Fribourg, Suisse, 271-276.
- Ali, H. (2008). A Back Propagation Neural Network Arabic Characters Classification Module Utilizing Microsoft Word. *Journal of Computer Science 4 (9)*, 744-751.
- Ali, S., Al-Omari, K., Sumari, P., Al-Taweel Sadik, A. et Husain Anas, J.A. (2009). Digital Recognition using Neural Network. *Journal of Computer Science 5 (6)*, 427-434.
- Amrouch, M., Rachidi, A., Elyassa, M. et Mammass, D. (2010). Handwritten Amazigh Character Recognition Based On Hidden Markov Models. *ICGST-GVIP Journal, Vol.10, Issue 5*, 11-18.
- Bencharef, O., Fakir, M., Minaoui, B. et Bouikhalene, B. (2011). Tifinagh Character Recognition Using Geodesic Distances, Decision Trees & Neural Networks. (*IJACSA*) *International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence*, 1-5.
- Blumenstein, M., C. K. Cheng et Liu, X. Y. (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, 480-484.
- Bouikhalene, B., Fakir, M. et Moro, K., (2009). Reconnaissance des Caractères Tifinaghe par l'Utilisation des Réseaux de Neurones Multicouches. *SITACAM'09*, Agadir, Maroc, 12-13 Décembre.
- Chevalier, S., Geoffrois, E., et Prêteux, F. (2003). 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, Maroc, 617-630.
- El Ayachi, R., Bouikhalene, B. et Fakir, M. (2011). Transformation de Fourier et Moments Invariants Appliqués à la Reconnaissance des Caractères Tifinaghe. *SITACAM'11*, Agadir, Maroc, 6-7 Mai, 41-54.
- El Ayachi, R., Fakir, M. et Bouikhalene, B. (2011). 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.
- El Ayachi, R., Fakir, M. et Bouikhalene, B. (2011). Recognition of TIFINAGHE Characters Using A multilayer Neural Network. *International Journal Of Image Processing (IJIP)*, vol. 5, Issue2, 109-118.
- El Ayachi, R., Moro, K., Fakir, M. et Bouikhalene, B. (2010). On The Recognition Of Tifinaghe Scripts. *JATIT*, vol. 20, No.2, 61-66.
- Es Saady, Y., Rachidi, A., El Yassa, M. et Mammass, D. (2011). Amazigh Handwritten Character Recognition based on Horizontal and Vertical Centerline of Character. *International Journal of Advanced Science and Technology Vol. 33*, Août, 33-50.
- Es Saady, Y., Rachidi, A., El Yassa, M. et Mammass, D. (2010). Printed Amazigh Character Recognition by a Syntactic Approach using Finite Automata. *ICGST-GVIP Journal, Volume 10, Issue 2*, Juin.
- Fakir, M. (2001). *Reconnaissance des Caractères Arabes Imprimés*. Thèse de doctorat, Faculté des Sciences, Semailia, Maroc.
- Fakir, M., Sodeyama, C. et Hassani, M. M. (2000). On the recognition of Arabic characters using Hough transform technique. *Malysian Journal of Computer Science Vol. 13, No.2*, Décembre, 39-47.
- Ghorbel, F. (1993). Application de la Transformée de Fourier Généralisée au Problème de l'invariance en Reconnaissance de Formes à Niveau de Gris. *Quatorzième Colloque GRESTI*, Juan-les Pins, 13 - 16 Septembre.
- Kavallieratou, E., Fakotakis, N. et Kokkinakis, G. (1999). Skew Angle Estimation In Document Processing Using Cohen's Class Distributions. *PRL(20)*, No. 11-13, Novembre, 1305-1311.
- Kessentini, Y., Paquet, T. et Benhamadou, A. (2010). Off-Line Handwritten Word Recognition Using Multi-Stream Hidden Markov Models. *Pattern recognition Letters (PRL)*, Vol 30, Issue 1, Janvier, 60-70.

Outahajala, M., Benajiba, Y., Rosso, P. et Zenkouar, L. (2011). 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*, 238-241.