

Systeme de reconnaissance de chiffres manuscrits hors lignes

Recognition System of Unconstrained Handwritten Numerals

par Nicolas FERAY et Denis DE BRUCQ

PSI - La3I - Itepeca, UFR des Sciences et des Techniques BPI 118

76 821 Mont Saint Aignan Cedex

Téléphone : 35.14.65.87 - FAX : 35.14.63.49 E-mail : Feray@la3i.univ-rouen.fr - Gdr.Tdsi 134 CNRS

résumé et mots clés

Nous présentons dans cet article un système de reconnaissance de chiffres manuscrits hors lignes, en décrivant toutes les étapes essentielles à son élaboration. Nous abordons d'abord la première phase de tout système de reconnaissance : l'extraction de primitives. Une représentation structurée construite à partir du squelette du chiffre est utilisée pour extraire rapidement un jeu de 55 primitives binaires. Nous précisons ensuite une méthode qui permet de déterminer l'information transmise par une primitive sur le problème de la reconnaissance des chiffres manuscrits hors lignes. L'information transmise par chaque primitive fournit un critère permettant de générer un arbre de décision binaire de manière complètement automatique. Ce critère est utilisé pour sélectionner au niveau de chaque nœud de l'arbre la primitive la plus informative sur le problème de reconnaissance associé au nœud en cours de traitement. Le classifieur obtenu n'utilise pas la totalité des 55 primitives binaires mais uniquement celles qui ont été retenues durant la phase d'identification de l'arbre de décision. Nous présentons ensuite un critère de rejet original qui permet d'augmenter les performances du système de reconnaissance de manière significative. Nous décrivons finalement la base de données de chiffres manuscrits américains qui sert à tester le classifieur. Nous donnons les résultats obtenus.

Squelettisation, Primitive, Théorie de l'information, Critère de rejet, Reconnaissance de chiffres.

abstract and key words

In this paper, we present a recognition system of unconstrained handwritten numerals. We describe all essential stages to its elaboration. We approach the first phase of all recognition system : the extraction of the primitives. A structure that use the skeleton of the numeral is used to extract rapidly 55 binary primitives. We specify a method that allows to determine the transmitted information about the primitives on the problem of the recognition of unconstrained handwritten numerals. Information transmitted by each primitive providing a criterion allowing to generate a binary decision tree. This criterion is used to select in each node the best primitive. The obtained classifier does not use the totality of 55 binary primitives but solely those that have been retained during the phase of identification of the decision tree. We present an original reject criterion that allows to increase performances of the recognition system. Finally, We describe the database of American handwriting numerals that serves to test the classifier. We demonstrate the performance of our system with this database.

Skeleton, Primitive, Information theory, Reject criterion, Handwritten numeral recognition.

1. introduction

Depuis plusieurs années, de nombreux travaux de recherche ont porté sur la reconnaissance des chiffres manuscrits. Deux grandes classes de systèmes sont aujourd'hui à l'étude : les systèmes pour

les applications bancaires ou postales, qualifiés de hors lignes ou statiques et les systèmes destinés à la bureautique, qualifiés de en lignes ou dynamiques ou temps réel (voir [1] et respectivement [2] pour des exemples).

Les travaux que nous présentons portent sur le premier problème qui reste aujourd'hui un thème de recherche ouvert. En effet, bien que le nombre de classes naturelles soit très réduit (chiffres '0'

à '9'), on trouve à l'intérieur de chacune d'entre elles, une très grande variabilité de l'écriture, [3]. De plus, les conditions souvent relativement précaires dans lesquelles sont écrits les chiffres (chèques écrits rapidement sur un coin de table) et la variabilité du matériel utilisé (utilisation de divers stylos, de différentes qualités de papier) tendent à complexifier la reconnaissance.

Dans ces conditions, l'extraction de caractéristiques est une étape essentielle et délicate dans la construction d'un système de reconnaissance. Cette phase est donc présente dans de nombreux systèmes de reconnaissance présentés dans la littérature. On trouve principalement deux catégories d'approche : l'analyse globale et l'analyse structurelle. La première approche s'attache à décrire les formes de manière globale, [4], [5], et elle est en général utilisée avec des méthodes statistiques de classification, [6]. Notre approche se situe dans la deuxième catégorie qui capture des primitives décrivant les principaux aspects géométriques de la forme. Ces primitives sont usuellement extraites du squelette ou du contour des formes, [7].

Nous présentons une décomposition simple du chiffre manuscrit sur plusieurs plans informatifs structurés, chargés de représenter complètement le squelette du chiffre manuscrit. Le squelette est donc l'élément déterminant en terme de robustesse de la représentation structurelle du chiffre. Notre algorithme de squelettisation et ses propriétés sont présentés.

Les différents plans informatifs de la structure précédente facilitent l'accès à l'information contenue dans le squelette. Ils sont donc utilisés par les extracteurs de primitives pour générer une description des chiffres manuscrits. Cette description est formée de 55 primitives binaires rapides à extraire. Les primitives retenues et le nombre de ces dernières ont été choisis de manière à ce que l'information transmise par le mot constitué à partir de la concaténation des 55 primitives binaires soit maximale, [9]. Le fait que l'information transmise, calculée sur un échantillon que l'on espère le plus représentatif du problème, soit maximale, est une condition nécessaire pour résoudre le problème.

Le classifieur, qui est un arbre de décision binaire, est ensuite présenté. Une telle démarche permet une phase de reconnaissance, réalisée à l'aide de questions sur des caractéristiques locales de l'image, très rapide. Cette approche est délicate en reconnaissance de chiffres manuscrits hors lignes à cause de la difficulté à construire un arbre, tâche qui ne peut être effectuée manuellement, étant donnée la multiplicité des règles attachées à chacune des classes. L'information transmise par une primitive fournit un critère de choix efficace [8], [9], qui est utilisé pour automatiser la fabrication de l'arbre de décision binaire lors de la phase d'apprentissage. Les nœuds de l'arbre sont identifiés de manière récursive. A chaque nœud, est associée la primitive qui maximise la quantité d'information transmise sur la sous-partie de la base d'apprentissage relative au nœud. D'autres méthodes utilisent un critère de type entropie pour construire un arbre de décision binaire. Dans [10], l'objectif est de construire un arbre de décision en réduisant l'entropie de Shannon H à tous les niveaux de l'arbre (ensemble des nœuds situés à la même profondeur dans l'arbre).

La technique développée dans [11] est très proche de la notre puisque les primitives, au niveau de chaque nœud, sont choisies pour réduire le plus possible, l'incertitude (mesurée par l'entropie) sur l'hypothèse vraie. L'approche de CART, [12], utilise un groupe relativement réduit de primitives à chaque nœud et chaque test est optimisé en fonction d'un seuil.

Notre approche s'appuie sur la proposition du paragraphe 4.1.1 qui formalise, d'un point de vue mathématique, l'intuition physique. Les distributions conditionnelles des caractéristiques binaires et la distribution marginale sur les chiffres sont estimées au niveau de chaque nœud sur une sous-partie de la base d'apprentissage pour calculer l'information transmise par chaque primitive. Le processus est adaptatif car la sous-partie de la base d'apprentissage et les informations transmises par les primitives au niveau d'un nœud dépendent des valeurs des primitives observées au niveau de tous les nœuds précédents. Ce système conduit à éclater chaque classe en sous-classes, correspondant à des variabilités de l'écriture spécifiques, représentatives des différentes manières de tracer le chiffre (variabilité d'écriture d'un scripteur à un autre) et aussi des différents outils ayant servi à tracer le chiffre (fermeture des boucles lors de l'utilisation de stylos à mine épaisse).

Nous précisons la capacité de cette méthode à réduire l'ensemble des caractéristiques à observer, pour ne retenir que les plus informatives, ce qui permet d'arriver à un arbre de décision de taille raisonnable.

Nous présentons ensuite un critère de rejet, simple à implémenter, qui permet d'améliorer de manière significative les performances du classifieur en terme de fiabilité.

Le classifieur sera testé, dans la dernière partie, sur la base de données de chiffres manuscrits américains NIST (National Institute and Standards of Technology).

2. représentation de haut niveau d'un chiffre manuscrit

Nous avons dû mettre en œuvre un certain nombre de prétraitements, ayant pour objectif de passer d'une représentation du chiffre sous forme d'image (difficile à exploiter) à une représentation de haut niveau du chiffre. Cette représentation devra permettre d'accéder facilement et rapidement aux caractéristiques informatives du chiffre en vue d'effectuer son identification.

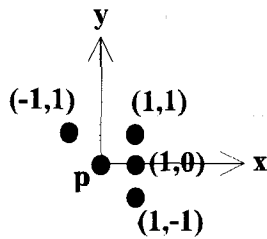
2.1. squelettisation

Cette première étape est souvent utilisée pour simplifier les formes à reconnaître, [13]. Elle permet notamment de s'affranchir du

problème de l'épaisseur du trait qui est extrêmement variable au niveau de l'écriture manuscrite hors ligne. En amont de cette étape, nous effectuons un lissage de l'image brute opérant sur un voisinage de 8 points. Cette étape permet de limiter l'apparition de trous à l'intérieur des chiffres et de lisser leurs contours extérieurs. L'algorithme de squelettisation présenté est une amélioration de l'algorithme de Marthon, [14], [15]. Contrairement à ce dernier, il permet d'obtenir un squelette d'épaisseur 1 pixel. Il est fondé sur le calcul, en chaque point $p(i, j)$ de l'image, d'une fonction simple $F(i, j)$ opérant sur un voisinage de 8 points. $F(i, j)$ est la longueur de la somme des vecteurs joignant le point $p(i, j)$ à tous ses voisins.

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$F(i, j) = |X| + |Y| = 3$$



Observons que $0 \leq F(i, j) \leq 4$. Intuitivement, les points intérieurs à la forme sont associés à une valeur de $F(i, j)$ petite et seront conservés. Les points extérieurs sont associés à une valeur de $F(i, j)$ grande et seront supprimés. La valeur de $F(i, j)$ est liée au degré d'intériorité dans la forme du pixel de coordonnées (i, j) . L'algorithme proposé est itératif et il supprime, à chaque étape, des points inessentiels (points n'appartenant pas au squelette). Les points supprimés sont les points de coordonnées (i, j) tels que $F(i, j) = 4$ et certains points tels que $F(i, j) = 3$. L'élimination de tous les points tels que $F(i, j) = 3$ entraîne en général une déconnexion du squelette, ce qu'il faut éviter. Aussi les cas $F(i, j) = 3$ sont analysés cas par cas. Donnons des exemples :

	$ X = 2, Y = 1, F(i, j) = 3$: le point est gardé.
	$ X = 3, Y = 0, F(i, j) = 3$: le point est enlevé.
	$ X = 1, Y = 2, F(i, j) = 3$: le point est enlevé.

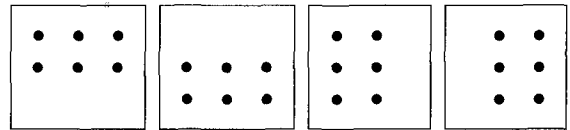
C'est la connaissance de $|X|, |Y|$ et de $nvois$ (nombre de voisins du point sur un 8-voisinage) qui permet de déterminer si le point

pour lequel $F(i, j) = 3$ est enlevé ou non.

$$\text{si } nvois = 2 \text{ ou } (nvois = 3 \text{ et } \sup(|X|, |Y|) = 3)$$

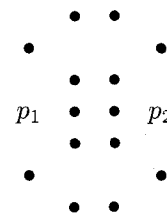
$$\text{alors le point doit être éliminé.} \quad (1)$$

Une dernière règle doit être ajoutée pour les cas où $F(i, j) = 3$. C'est le cas des configurations doubles suivantes.



Ces cas correspondent à $nvois = 5$ et $\sup(|X|, |Y|) = 3$

Ces configurations sont détectées et, en phase de suppression, l'élimination de ces points est faite uniquement si l'on est dans un cas de configuration ligne double. L'intérêt de cette démarche est illustré par l'exemple suivant :

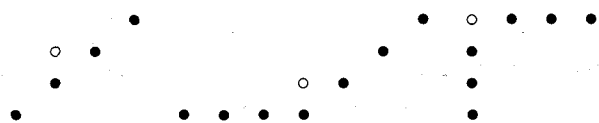


Les points p_1 et p_2 ont tous les deux une configuration ligne double, mais un seul des deux doit être éliminé pour qu'il n'y ait pas de disconnexion de la forme. En phase de suppression, le point p_1 sera supprimé car c'est un cas de configuration ligne double (calculé au moment de la suppression) et le point p_2 sera conservé car la suppression du point p_1 aura entraîné la disparition de la configuration ligne double pour le point p_2 .

L'algorithme procède de façon itérative par paire d'étapes : la première calcule, en chaque point $p(i, j)$ de la forme, la fonction $F(i, j)$. La deuxième étape supprime les points inessentiels tels que $F(i, j) = 4$ et procède à l'élimination des points associés à $F(i, j) = 3$ lorsque la condition (1) est remplie ou s'il y a détection d'une ligne double. Ces deux étapes sont réitérées jusqu'à ce qu'il n'y ait plus de point inessentiel.

Finalement, afin que l'épaisseur du squelette soit strictement d'épaisseur 1, l'algorithme se termine par un traitement itératif, [15], qui élimine les points ayant la configuration suivante :

$$(|X| = 1 \text{ et } |Y| = 1 \text{ et } nvois = 2) \text{ ou } (|X| = 0 \text{ et } |Y| = 2 \text{ et } nvois = 3) \text{ ou } (|X| = 2 \text{ et } |Y| = 0 \text{ et } nvois = 3)$$



Exemples de configurations typiques
○ : points éliminés en fin de squelettisation

Cette méthode permet d'obtenir une représentation structurale robuste des chiffres manuscrits. Ainsi :

- L'algorithme a été appliqué sur des chiffres manuscrits de la base NIST (figure 1). Le bruit de contour n'introduit pas d'artefact dans le squelette, les extrémités des branches ne sont pas raccourcies, les courbures des formes sont bien représentées. Ces points sont indispensables pour l'extraction de primitives structurelles robustes.

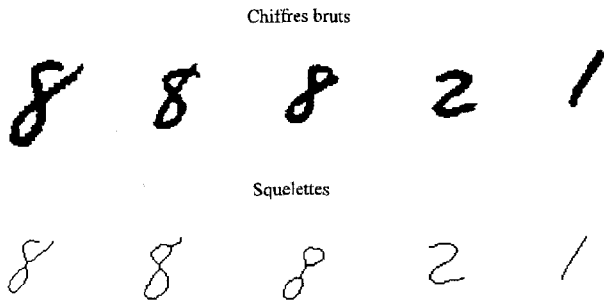
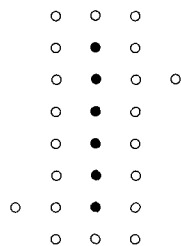


Figure 1. - Squelettisation.

- L'épaisseur du squelette est toujours de 1.
- La méthode offre une faible sensibilité au bruit.



• points du squelette

- Le squelette conserve les propriétés métriques de la forme comme le nombre de parties, le nombre de trous et la connexité.
- L'algorithme utilise un critère simple d'élimination des points. Ceci permettra une implémentation câblée aisée. Ce point est intéressant car l'algorithme est itératif. Le temps moyen de squelettisation estimé sur 200 chiffres de la base NIST (images de 128×128 pixels) est de 8 centièmes de seconde sur un compatible PC DX2 66 MHz.

2.2. description du chiffre à partir du squelette

Nous sommes passés d'une représentation du chiffre sous forme d'image à une représentation de ce dernier par son squelette. L'objectif est d'obtenir maintenant, [16], une représentation structurée de plus haut niveau en utilisant son squelette (figure 2).

L'objectif est de structurer l'information afin de rendre l'utilisation du squelette et donc la phase d'extraction des primitives, plus

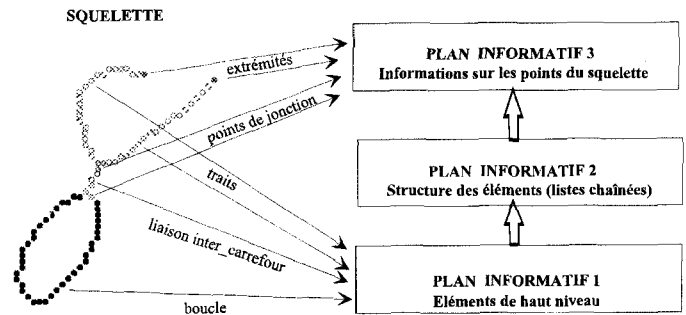


Figure 2. - Structuration du squelette .

aisée et plus rapide. D'autre part, la totalité des plans informatifs n'engendrant pas de perte d'information par rapport au squelette initial, cette représentation laisse une grande liberté à de futurs traitements. Une chaîne de reconnaissance est souvent amenée à évoluer par ajout ou modification d'extracteurs de primitives.

Le premier plan informatif est constitué à partir de la décomposition du squelette du chiffre en éléments de haut niveau facilement manipulables (trait, liaison inter-carrefour, boucle).

Ce plan donne accès à un deuxième plan informatif où l'on trouve la description de chaque élément de haut niveau sous la forme d'une liste des points le constituant. L'ordre des points dans la liste chaînée est donné par un suivi de trait et n'est donc pas aléatoire. Ce plan permet notamment l'extraction de primitives décrivant les ouvertures présentes au niveau du chiffre.

Le deuxième plan permet l'accès au troisième et dernier plan informatif, où l'on trouve la description des points du squelette (coordonnées, type du point, nombre de voisins sur un 8-voisinage). Ce plan permet l'extraction de primitives représentatives de points caractéristiques du squelette (points extrémités, points de jonction) ainsi que toutes les opérations de localisation d'un élément (élément dans la partie basse, haute, etc du chiffre). Il permet aussi une opération de projection du squelette qui a pour but la détection de traits horizontaux significatifs au niveau du chiffre.

Identification des points du squelette.

Les points du squelette sont classés en trois catégories suivant le nombre de voisins sur un 8-voisinage :

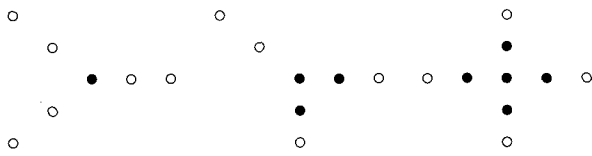
- les points tels que $nvois = 1$, ce sont des extrémités.
- les points tels que $nvois = 2$, ce sont des points situés sur un élément de haut niveau.
- les points tels que $nvois \geq 3$, ce sont des points de jonction.

Ce paramètre sera disponible dans le troisième plan informatif au niveau de la description de chaque point.

Identification des carrefours.

Cette étape est essentielle pour l'extraction d'éléments de plus haut niveau. Un carrefour est défini comme un ensemble connexe de points de jonction . Nous effectuons donc une extraction des

points de jonction connexes, [17].



Exemples de carrefours
 ● : point de jonction, $nvois \geq 3$
 ○ : $nvois < 3$

Chaque carrefour (masse connexe) est défini par un identificateur ID – carrefour :

$$0 < ID - \text{carrefour} \leq \text{nombre total de carrefours}$$

Chaque point du troisième plan informatif dispose dans sa description d'un paramètre ID – carrefour. Celui-ci permet de savoir à quel carrefour le point appartient. Si le point n'appartient pas à un carrefour, le paramètre ID – carrefour est égal à zéro.

Extraction des traits du squelette

Un trait du squelette est un élément qui a pour origine un point extrémité et pour fin un point extrémité ou un point de jonction. Pour extraire les traits du squelette, nous partons des points extrémités pour effectuer un suivi de trait jusqu'à ce que l'on arrive sur un point extrémité ou un point de jonction.

Extraction des liaisons inter-carrefour.

Une liaison inter-carrefour est un élément qui a pour origine un point de jonction et pour fin un point de jonction. Pour extraire les liaisons inter-carrefour, nous partons des points de jonction pour effectuer un suivi de trait jusqu'à ce que l'on arrive sur un autre point de jonction.

Extraction des boucles.

L'algorithme d'extraction des boucles consiste à assembler les liaisons inter-carrefour entre elles pour former des parcours fermés (recherche de cycles dans un graphe non orienté où les carrefours sont les sommets des liaisons inter-carrefour). L'assemblage se fait uniquement grâce à la connaissance, pour chaque liaison inter-carrefour, du premier point et du dernier point de la liste chaînée ainsi que du paramètre ID – carrefour associé à ces points dans le troisième plan informatif. Bien que l'algorithme utilisé soit combinatoire, l'extraction est très rapide car elle travaille sur des éléments de haut niveau, les liaisons inter-carrefour, faciles à manipuler et en nombre très réduit (en règle générale, le nombre de liaisons inter-carrefour n'est jamais supérieur à quatre).

Nous avons, par ces différentes opérations, décomposé le squelette en différents éléments de haut niveau. Chaque élément est décrit par une liste chaînée des points le constituant. Il n'y a eu aucune perte d'information, mais uniquement une structuration du squelette. Chaque élément de haut niveau est :

- Facile à manipuler grâce à la structure qui le représente (liste chaînée des points le constituant).
- Structuré : l'ordre des points de la liste chaînée est représentatif du suivi de trait.
- Riche en information : connaissance de la nature des points le constituant, de leurs coordonnées dans l'image initiale.
- Facilement interrogeable par des dispositifs situés en aval pour des traitements spécifiques.

3. extraction de caractéristiques en vue de la reconnaissance

Les primitives que nous présentons exploitent pleinement la structure présentée précédemment pour arriver à une extraction rapide. Le choix des primitives s'est fait après une étude bibliographique, [18], [19], [20], qui nous a permis de déterminer celles qui semblaient à la fois les plus informatives et les plus simples. Pour chaque famille de primitives, nous précisons les plans informatifs du squelette structuré utilisés.

Localisation des extrémités (plans informatifs 1 et 3 utilisés)

Les extrémités et les carrefours constituent des informations très utiles pour la reconnaissance, car elles permettent de décrire localement la structure d'une forme, [21]. Les extrémités se trouvent à la fin des éléments de haut niveau appelés traits. Il est courant, en reconnaissance de chiffre manuscrit, de considérer le nombre d'extrémités. Expérimentalement, nous avons pu constater que cette primitive n'est pas robuste lorsque les chiffres sont très bruités à cause de l'apparition de barbules. Nous préférons donc utiliser les positions approchées de ces dernières. Nous introduisons des primitives binaires $E_i, i \in \{1, 2, \dots, 9\}$ indiquant la présence ou l'absence d'extrémité dans des zones du rectangle englobant le chiffre (figure 3).

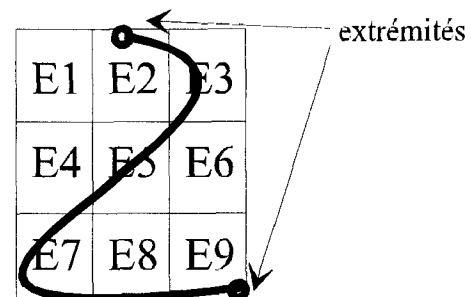


Figure 3. – Primitives extrémités.

Localisation et dénombrement des carrefours (plans informatifs 1 et 3 utilisés)

Comme pour les extrémités, nous localisons les carrefours. Nous obtenons ainsi neuf primitives binaires (Ec1, Ec2, Ec3, Ec4, Ec5, Ec6, Ec7, Ec8, Ec9). Nous ajoutons quatre autres primitives binaires, voir tableau 1, chargées de spécifier le nombre de carrefours (certains chiffres comme le 8 peuvent avoir dans une même zone deux carrefours).

Tableau 1. – Primitives nombre de carrefours

0 carrefour	Ca0
1 carrefour	Ca1
2 carrefours	Ca2
3 carrefours	Ca3
plus de 3 carrefours	Ca>3

Détection et localisation des boucles (plans informatifs 1, 2 et 3 utilisés)

Les boucles et les ouvertures mettent en évidence les propriétés topologiques et géométriques des chiffres tout en présentant une grande tolérance aux distorsions et aux variations de style. Nous avons facilement accès aux boucles grâce au premier plan informatif. L'étape suivante consiste à caractériser le type des boucles (voir tableau 2) grâce à la description de ces dernières dans les plans 2 et 3.

Tableau 2. – Primitives boucles

présence boucle	O
boucle totale	Ot
boucle basse	Ob
boucle haute	Oh
boucle double	Od

Nous approchons l'aire d'une boucle par l'aire A_o du rectangle dans lequel elle est inscrite. Soit $I(Ix, Iy)$ le « centre » de la boucle (figure 4).

Soit A_c , l'aire du rectangle dans lequel le chiffre est inscrit et $y = \frac{(y1+y2)}{2}$ l'équation de la droite horizontale qui coupe le chiffre dans sa moitié.

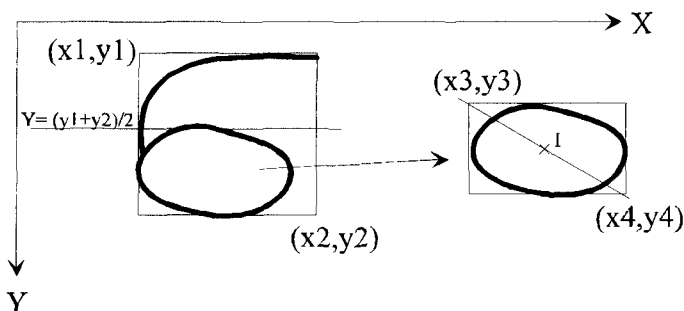


Figure 4. – Primitives boucles.

1) Si $A_o < S_{min} \times A_c$, la boucle est considérée comme trop petite et est ignorée.

2) Sinon, si $A_o > S_{max} \times A_c$, l'aire de la boucle est très importante vis-à-vis de A_c : elle est considérée comme totale (cas du chiffre zéro).

3) Sinon, si $Iy > \frac{(y1+y2)}{2}$, la boucle est basse (exemple du chiffre 6).

4) Sinon, si $Iy < \frac{(y1+y2)}{2}$, la boucle est haute (exemple du chiffre 9).

5) S'il y a une boucle haute et une boucle basse alors la boucle est double (exemple du chiffre 8).

Les seuils sont obtenus en faisant varier la valeur de S_{min} et S_{max} par pas de 0,1 ($0 \leq S_{min} \leq 1$ et $0 \leq S_{max} \leq 1$) et en retenant le couple qui maximise la quantité d'information transmise (définie au paragraphe 4.1.1) par le mot constitué de la concaténation des cinq primitives binaires décrivant les boucles présentes au niveau d'un chiffre. Ce procédé est décrit en détail dans [9].

Détection de primitives décrivant le tracé du chiffre (plans informatifs 1,2 et 3 utilisés).

La première phase consiste en une polygonalisation rapide, [22], des différents éléments du premier plan informatif. Ainsi, à chaque trait du squelette structuré, est associée une suite de segments caractérisés par leur direction de Freeman. Cette suite de directions de Freeman donne une description approchée de la nature du tracé d'un élément trait. Seuls les éléments traits sont polygonalisés car l'objectif est de détecter des arcs concaves ou convexes qui n'apparaissent pas de manière significative sur les liaisons inter-carrefour. Les suites de directions de Freeman associées aux traits sont ensuite analysées par des automates finis pour détecter toutes les ouvertures présentes au niveau du chiffre. L'automate non déterministe présenté figure 5 détecte les ouvertures O7, O8, O9.

Pour l'implémentation informatique, nous nous ramenons à un automate déterministe, [23]. Les ouvertures sont ensuite localisées avec le procédé utilisé pour les boucles.

Détection des traits horizontaux (plan informatif 3 utilisé)

Un histogramme de projection du squelette est construit de la façon suivante. L'axe des abscisses représente les lignes de l'image et l'axe des ordonnées représente le nombre de pixels noirs rencontrés sur chaque ligne. Tous les endroits où le nombre de pixels noirs est supérieur à la plus grande largeur du chiffre multipliée par un seuil S_p ($0 < S_p < 1$), sont détectés pour déterminer trois types de traits horizontaux, voir tableau 3.

Nous projetons le squelette plutôt que le chiffre brut pour éviter les problèmes liés aux sur-épaisseurs locales du tracé du chiffre (fréquentes sur le manuscrit) qui perturbent l'extraction des traits horizontaux. De plus, la projection qui est obtenue en un seul balayage du troisième plan informatif, manipule un minimum de points par rapport aux projections classiques qui utilisent l'image brute. L'utilisation du squelette conduit à détecter des

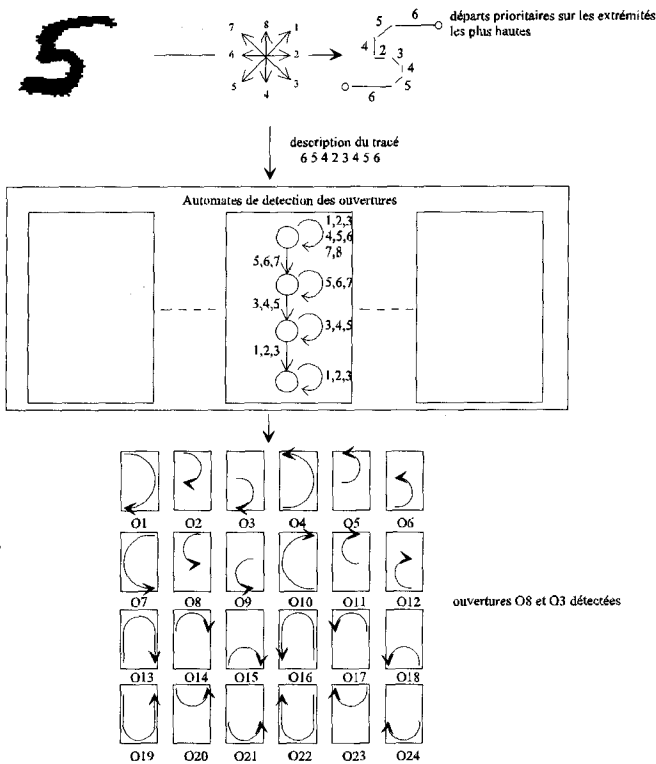


Figure 5. – Primitives ouvertures.

Tableau 3. – Primitives projections

Type du trait	Configuration
Trait horizontal dans la partie haute du chiffre	Th
Trait horizontal dans la partie médiane du chiffre	Tm
Trait horizontal dans la partie basse du chiffre	Tb

traits horizontaux plus ou moins marqués suivant la valeur de S_p . Le choix de la valeur du seuil S_p est fait sur le même principe que celui employé pour déterminer les seuils associés aux boucles.

4. construction d'un classifieur

La phase de reconnaissance consiste à interpréter les données obtenues dans l'étape précédente et à décider à quelle classe appartient le chiffre analysé. Cette décision d'appartenance du chiffre à une classe va être effectuée par un arbre binaire chargé de faire la mise en adéquation des propriétés de l'objet aux propriétés de la classe. Ce classifieur sera obtenu à la suite d'une phase d'apprentissage entièrement automatique.

4.1. construction de l'arbre binaire à partir des quantités d'information transmise par les caractéristiques extraites des chiffres

Un arbre binaire est composé d'un ensemble de nœuds, chaque nœud pouvant activer l'un de ses deux nœuds fils. Un nœud de l'arbre est chargé, pour identifier le chiffre, de poser une question. Suivant la réponse (oui ou non), le parcours dans l'arbre se poursuit vers l'un des nœuds fils où une nouvelle question est posée. Le processus démarre au niveau de la racine de l'arbre et se réitère jusqu'à l'arrivée sur un nœud terminal (nœud n'ayant pas de fils) qui va permettre l'identification du chiffre. L'apprentissage consiste donc à découvrir, en partant de la racine, la question qu'il faut poser puis à créer deux nœuds fils et enfin à réitérer le processus au niveau de ces derniers. Ce processus appliqué à un problème complexe comme la reconnaissance de chiffre manuscrit hors ligne, doit être automatisé et sous-entend donc de disposer d'un critère qui va permettre de poser la bonne question au niveau d'un nœud.

Nous avons choisi un critère qui va permettre de calculer, au niveau d'un nœud, la quantité d'information transmise [8], [9] par les primitives sur le problème d'identification.

L'information transmise, comme nous allons l'utiliser, sera donc une fonction d'évaluation des primitives présentées précédemment sur un problème de reconnaissance défini dans un contexte très précis et de plus en plus spécifique (nœud de l'arbre). L'utilisation d'une information transmise est intuitive. Roche, [8], a montré que dans tous les problèmes d'applications où l'homme a une certaine idée des informations utiles transmises, la quantité d'information transmise, $I(Z; Y_n)$ dans notre cas, correspond à l'intuition.

4.1.1. calcul de la quantité d'information transmise par les primitives.

Soit une base de données de chiffres étiquetés, chaque chiffre pouvant être identifié grâce à son étiquette. Les chiffres à lire étant aléatoires, nous introduisons une variable aléatoire Z d'un espace de probabilité (Ω, A, P) dans $\underline{Z} = \{0, 1, \dots, 9\}$. Pour chaque épreuve ω , le chiffre $Z(\omega) = i$ représente le chiffre aléatoire inconnu que le scripteur a voulu écrire. Ce dernier a tracé une forme notée $X(\omega)$.

Nous supposons que le problème de classification est bien posé. De façon plus précise, nous supposons qu'il est possible de retrouver le chiffre à partir de la forme observée; les images sont suffisamment fines pour que toute forme $X(\omega)$ corresponde à un seul chiffre i ; deux chiffres appartenant à des classes différentes ne peuvent avoir la même forme. Ainsi, au chiffre i est associé un ensemble F_i , comprenant toutes les formes possibles de tracé

de i . Nous admettons ainsi que :

$$\begin{aligned} A_i &= \{\omega \in \Omega; Z(\omega) = i\} \\ &= \{\omega \in \Omega; X(\omega) \in F_i\} \end{aligned}$$

Nous avons introduit 55 variables aléatoires $Y_n, n \in \{1, \dots, 55\}$; $Y_n(\omega) = y_n$ où $y_n \in \{0, 1\}$ représente la valeur de la primitive Y_n extraite pour l'épreuve ω c'est-à-dire à partir de la forme $X(\omega)$. Ainsi, connaissant les valeurs d'une séquence de primitives obtenues à partir d'une forme donnée, il s'agit de retrouver le chiffre. Usuellement, la reconnaissance s'appuie sur la distribution *a posteriori*, conditionnellement aux valeurs des primitives observées.

Nous n'allons pas calculer l'ensemble de toutes les primitives. Nous allons en choisir un nombre, le plus restreint possible. Le choix des primitives est effectué sur la base de l'information que donne une primitive Y_n sur la variable aléatoire Z . Un rappel rapide des notions d'information et d'information conditionnelle est donné afin que le texte soit immédiatement compréhensible par tous les lecteurs.

Nous allons maintenant évaluer numériquement l'incertitude sur l'expérience de reconnaissance pour chacune des 55 primitives. La mesure d'incertitude, *a priori*, est égale à, [24] :

$$H(Z) = -P(A_0) \log_2 P(A_0) - \dots - P(A_p) \log_2 P(A_p)$$

Cette quantité est l'entropie de la variable aléatoire Z et son unité est le bit. Pour le cas de la reconnaissance de chiffres manuscrits hors lignes et pour des événements équiprobables, nous avons $p = 9$ et $H(Z) = 3,321 \text{ bits}$.

Les probabilités *a posteriori* vérifient la formule de Bayes :

$$P(A_i/y_n) = \frac{P(A_i)P(y_n/A_i)}{\sum_{j \in \mathcal{Z}} P(A_j)P(y_n/A_j)}$$

Lors de la reconnaissance, chaque variable aléatoire Y_n réalise une valeur y_n qui conditionne la variable aléatoire Z , d'où les incertitudes *a posteriori* :

$$\begin{aligned} H(Z/y_n) &= -P(A_0/y_n) \log_2 P(A_0/y_n) \\ &\quad - \dots - P(A_9/y_n) \log_2 P(A_9/y_n) \end{aligned}$$

En faisant les sommations sur les valeurs possibles des primitives, on obtient :

$$H(Z/Y_n) = - \sum_{y_n=0,1} P(Y = y_n) \sum_{i \in \mathcal{Z}} P(A_i/y_n) \log_2 P(A_i/y_n)$$

Il est à espérer que l'observation Y_n est utile et que l'incertitude *a posteriori* est plus petite que l'incertitude *a priori*. Nous introduisons la quantité d'information transmise par chacune des 55 primitives :

$$I(Z; Y_n) = H(Z) - H(Z/Y_n)$$

A la suite de ces résultats classiques, nous fournissons quelques propriétés originales et utiles quant au choix d'une séquence de primitives.

Observons que l'ensemble d'apprentissage est la base B de données fournies. Cet ensemble peut être considéré comme un échantillon aléatoire, extrait de la population totale Ω . La reconnaissance doit être effectuée sur cette population totale. Le problème se pose de savoir si cet échantillon B est représentatif des propriétés statistiques de la population totale Ω .

Par ailleurs, considérons un procédé de découpage de B en deux classes C_0 et C_1 telles que $C_0 \cup C_1 = B$. C'est la situation après l'observation de la première primitive et successivement ensuite lorsque chaque séquence de primitives est augmentée d'une unité. Chacune des deux classes peut être considérée comme un échantillon de la base.

Ainsi, le problème se pose d'évaluer la quantité d'information transmise sur Z par une primitive Y quand on l'utilise dans une population Ω ou dans une partie B de cette population. Des exemples simples de construction d'espaces de probabilité (Ω, A, P) , construction laissée aux lecteurs, montrent que diverses situations contradictoires sont possibles.

Si la variable Y est indépendante de la variable Z dans chacune des deux classes C_0 et C_1 , alors l'information fournie par Y sur Z est nulle dans chacune des deux classes bien que l'information fournie par Y sur Z dans la population $C_0 \cup C_1$ n'ait aucune raison d'être nulle. Ainsi, dans ce cas particulier, le découpage en sous-classes éliminerait des primitives utiles sur la population totale.

Tout au contraire, il est aisé de construire un exemple où, dans chacune des deux classes C_0 et C_1 , la variable Y détermine la variable Z . Dans ce cas, l'information calculée dans chacune des deux classes C_0 et C_1 est maximum et il faut procéder au découpage. Pour faire un parallèle, prenons la couleur rouge. Prise seule, la couleur rouge serait notre primitive de base. Elle ne semble pas contenir énormément d'information si l'on veut reconnaître des fleurs (globalement elle est peu informative). Si l'on se ramène maintenant à un contexte particulier (une partie de la population totale) la couleur rouge permet de différencier une marguerite d'une rose ou une tulipe d'une jonquille, etc... Il suffit pour se ramener à ces contextes bien spécifiques de disposer de primitives aptes à les décrire. Certains lecteurs aimeront mieux une formulation mathématique de ces résultats.

Proposition :

Soit (Ω, A, P) un espace de probabilité, soit Z et Y deux variables binaires et $\xi \hat{=} \{C_0, C_1\}$ une partition de Ω .

a) Si $Z \perp\!\!\!\perp Y/\xi$ alors $I(Z, Y/\xi) = 0$.

b) Si $Z = f(y; C_i) \forall i \in \{0, 1\}$, avec f fonction à valeurs binaires, alors $H(Z/Y, \xi) = 0$.

Il existe des exemples pour la propriété a) avec

$$0 = I(Z; Y/\xi) < I(Z; Y)$$

et des exemples pour la propriété b) avec

$$H(Z/\xi) = I(Z; Y/\xi) > I(Z; Y).$$

En conclusion de ces propriétés, nous avons déduit que le choix de la meilleure primitive est un problème local à l'ensemble sur lequel se pose la classification. Etant donnée une séquence de primitives, dans chacune des sous-populations définies par les valeurs prises par les primitives de la séquence, le problème de classification se pose à l'identique et il faut prendre la primitive Y fournissant l'information la plus grande sur Z .

Notre mécanisme de classification va s'attacher à dégager des contextes de plus en plus précis, jusqu'à ce que les contextes soient si précis (regroupement d'une variabilité de l'écriture spécifique) qu'ils permettent de formuler une hypothèse de reconnaissance.

Il est essentiel que la base d'apprentissage B et que la population totale Ω (ou la population test) aient, à chaque niveau de découpage, les mêmes répartitions statistiques. Intuitivement, en cas de différences notables, il faut espérer une robustesse de l'utilité des primitives vis-à-vis des fluctuations statistiques entre l'échantillon B et la population Ω . Une condition nécessaire pour que B soit représentative de Ω est que, quel que soit la primitive Y considérée, nous ayons :

$$I(Z; Y/B) = I(Z; Y/\Omega)$$

4.1.2. algorithme d'identification de l'arbre binaire.

Soit une base de données d'apprentissage qui va permettre d'identifier l'arbre de décision binaire. L'identification va commencer en partant de la racine (nœud numéro 0, noté N_0) de l'arbre en considérant l'ensemble de la base d'apprentissage notée B_0 . Nous allons donc calculer pour chacune des 55 primitives la quantité $I(Z; Y_n/B_0)$. Une correspondance entre n et les primitives binaires présentées précédemment est donnée en annexe. L'information transmise par chacune des primitives au niveau de la racine de l'arbre est aussi donnée.

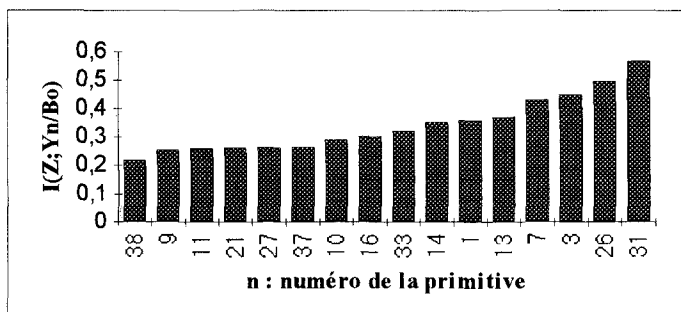


Figure 6. – Classement des 16 primitives les plus informatives.

La primitive numéro $m = 31$ telle que :

$$I(Z; Y_{m=31}/B_0) = \max(I(Z; Y_n/B_0), n \in \{1, 2, \dots, 55\})$$

est retenue pour poser la première question qui est donc « la primitive $m = 31$ est-elle présente? » et qui a pour réponse 0 ou 1 (respectivement oui ou non). Cette observation est réalisée sur tous les chiffres de la base de données d'apprentissage associée au nœud. Ceci permet de générer deux bases de données B_1 et B_2 .

$$\begin{aligned} B_1 &= \{\omega ; Y_m(\omega) = 1\} \\ B_2 &= \{\omega ; Y_m(\omega) = 0\} \\ B_0 &= B_1 \cup B_2 \end{aligned}$$

Deux nœuds fils, N_1 et N_2 , sont alors créés et à chaque nœud fils sont attachées respectivement la base B_1 et la base B_2 .

Le processus est réitéré sur chacun des nœuds nouvellement créés en considérant, pour le calcul des informations transmises par les primitives, la sous-base attachée au nœud en cours de traitement, soit B_1 , soit B_2 . Un nœud N_i de base B_i ne crée plus de fils et devient terminal lorsque :

$$\forall n \in \{1, 2, \dots, 55\}, I(Z; Y_n/B_i) = 0$$

La base B_i attachée à ce nœud permet alors de définir à quel chiffre le nœud N_i doit être associé. Le meilleur cas de figure arrive lorsque tous les chiffres de la base de données attachée au nœud terminal ont la même valeur (le nœud est pur). Dans le cas où la base B_i contient des représentants de plusieurs chiffres, la reconnaissance n'est pas complète et il faut introduire de nouvelles primitives spécifiques à la levée d'ambiguïtés persistantes, [25]. Remarquons que ce dernier point peut être prévu avant de construire l'arbre de décision. En effet, dans ces conditions, l'information transmise par le code constitué à partir de la concaténation de toutes les primitives binaires (calculée sur la base d'apprentissage) ne peut être maximum.

$$I(Z ; Y_1 Y_2 \dots Y_n / B_0) < H(Z) \text{ avec } n = 55 \text{ dans notre cas.}$$

Malgré tout, il est tout de même intéressant de construire l'arbre de décision binaire lorsque l'information transmise est proche de $H(Z)$ car l'analyse des bases de données attachées aux nœuds impurs permet de découvrir les primitives spécifiques qu'il faut rajouter.

En phase d'apprentissage, le problème de reconnaissance est donc, dans un premier temps (racine de l'arbre), considéré dans sa globalité (informations transmises calculées sur une variabilité de l'écriture extrême, donc sur la totalité de la base d'apprentissage) et au fur et à mesure que nous allons descendre dans l'arbre de décision binaire, les bases de données attachées aux nœuds de l'arbre vont être réduites et homogénéisées. Les variabilités de l'écriture à l'intérieur de ces bases de données vont devenir plus réduites et aussi plus spécifiques. C'est l'utilisation en priorité des primitives les plus informatives (les plus robustes aux variations de style) qui diminue la complexité du problème de reconnaissance, lors de la descente dans l'arbre, en partitionnant la base de données initiale. Il sera alors possible, sur ces différentes parties, d'obtenir des primitives ayant une quantité d'information non négligeable, alors que, sur le problème global, elles étaient peu informatives.

4.2. mécanisme de rejet

L'arbre de décision observe un certain nombre de primitives dans un ordre bien spécifique afin de déterminer le chiffre inconnu, décision qui se fait au niveau des nœuds terminaux. Le problème de l'arbre de décision binaire est qu'il est susceptible de faire une erreur au niveau de chaque nœud (primitive observée provenant d'un bruit, exemple d'une barbule entraînant l'apparition d'une primitive extrémité non significative). Ceci se traduit alors par l'arrivée sur un nœud terminal qui attribue une mauvaise valeur au chiffre inconnu et donc provoque une erreur. Au niveau des nœuds terminaux, nous avons inclus un mécanisme capable d'entraîner le rejet du chiffre s'il existe des primitives ayant une valeur aberrante au vue du chiffre proposé. Par exemple, il serait aberrant de conclure à la reconnaissance du chiffre 3 si le chiffre analysé possède une boucle totale ($O_t = 1$). Nous avons donc associé à chaque nœud terminal des primitives binaires qui ne doivent pas être observées (primitives binaires égales à 0) ou bien qui doivent être observées (primitives binaires égales à 1) pour conclure à la reconnaissance.

Soit T l'ensemble des nœuds terminaux. A chaque nœud terminal $N_q \in T$ sont associés une base de données B_q et un ensemble O_q contenant la liste des primitives à observer pour passer du nœud racine N_0 au nœud terminal N_q . B_q et O_q permettent de définir deux ensembles :

$$R_1(q) \hat{=} \{Y_n \notin O_q, n \in \{1, 2, \dots, 55\}; \forall \omega \in B_q, Y_n(\omega) = 1\}$$

$$R_0(q) \hat{=} \{Y_n \notin O_q, n \in \{1, 2, \dots, 55\}; \forall \omega \in B_q, Y_n(\omega) = 0\}$$

Les ensembles $R_1(q)$ et $R_0(q)$ constituent la liste des primitives qui doivent (ensemble $R_1(q)$) ou qui ne doivent pas (ensemble $R_0(q)$) être observées au niveau du chiffre inconnu pour valider l'hypothèse de reconnaissance au niveau du nœud N_q . Quel que soit q tel que $N_q \in T$, les ensembles $R_1(q)$ et $R_0(q)$ sont générés automatiquement lors de la phase d'identification de l'arbre de décision binaire à partir des bases de données associées aux nœuds terminaux.

Le critère de rejet proposé est original car il n'introduit pas de calcul de seuil. Il travaille uniquement sur des observations simples « le chiffre doit être rejeté car j'observe ou je n'observe pas telles et telles primitives ».

L'arbre de décision binaire observe donc un certain nombre de primitives au niveau d'un chiffre inconnu qui l'amène à formuler une hypothèse sur l'identité du chiffre inconnu. A chaque hypothèse prise conditionnellement à un nœud terminal, nous lui associons une partie déterministe sous la forme de primitives devant avoir des valeurs bien spécifiques pour que l'hypothèse de reconnaissance soit validée. L'intérêt du critère est que, plus la taille de la base de données au niveau d'un nœud terminal sera faible (variabilité de l'écriture très spécifique, chiffre très bruité), plus la liste de primitives constituant la partie déterministe de l'hypothèse de reconnaissance sera importante et donc plus

nous serons exigeants en phase de reconnaissance pour valider l'hypothèse de reconnaissance.

5. résultats

Nous travaillons sur des chiffres stockés sur disque sous la forme d'images binaires de 128×128 pixels. Ces chiffres sont extraits de la base de données américaine NIST. Il faut noter que les chiffres à notre disposition n'ont subi aucun pré-traitement avant d'être mis sous forme d'images. Par conséquent, ces dernières peuvent présenter les défauts suivants, (figure 7) :

- Images contenant du bruit
- Présence de chiffres fragmentés
- Formes provenant des cas de chevauchement de caractères : certaines images contiennent un chiffre et une partie d'un autre
- Taille des chiffres variable.

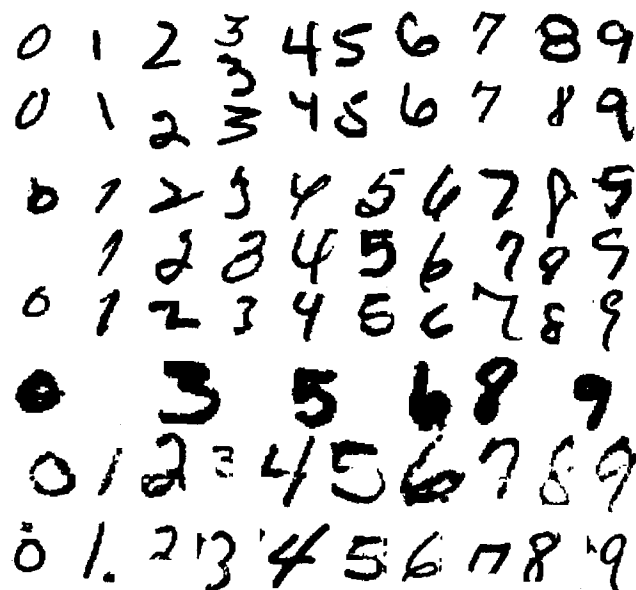


Figure 7. – Exemples de chiffres de la base NIST.

Nous avons constitué trois bases d'apprentissage : une de 1000 chiffres, une de 3500 chiffres et une de 6000 chiffres. D'autre part, une base de test de 6300 chiffres a été constituée. La base de test et les bases d'apprentissage sont sans recouvrement. De plus, il faut noter que les bases d'apprentissage et de test n'ont pas été modifiées pour éliminer certains chiffres non significatifs. Le processus d'apprentissage est donc entièrement automatique et n'introduit à aucun moment des facteurs humains qui sont souvent subjectifs.

Les résultats de reconnaissance sont présentés dans les tableaux 4 et 5.

**Tableau 4. – Reconnaissance sans critère de rejet.
Reconnaissance sur 6300 chiffres.
Base d'apprentissage et de test sans recouvrement**

Effectif de la base d'apprentissage	Reconnaissance	Erreur	Nombre de nœuds de l'arbre
$T = 1000$	86,98%	13,02%	140
$T = 3500$	89,44%	10,56%	598
$T = 6000$	90,32%	9,68%	1 036

**Tableau 5. – Reconnaissance avec critère de rejet.
Reconnaissance sur 6300 chiffres.
Base d'apprentissage et de test sans recouvrement**

Effectif de la base d'apprentissage	Reconnaissance	Rejet	Erreur
$T = 1000$	76,1%	20,36%	3,54%
$T = 3500$	81,8%	15,3%	2,9%
$T = 6000$	83,21%	14,19%	2,6%

Le tableau 5 montre l'intérêt du critère de rejet. En effet, sans ce dernier, nous constatons un bon taux de reconnaissance mais une erreur trop importante. L'utilisation du critère de rejet permet de diminuer l'erreur dans des proportions très importantes tout en gardant un taux de reconnaissance correct. Le rejet permet donc d'augmenter l'efficacité e du classifieur, voir tableau 6.

Tableau 6. – Influence du rejet sur l'efficacité

$T = 6000$	$e = \frac{\text{Bonne reconnaissance}}{\text{Bonne reconnaissance} + \text{erreur}} \times 100$
Sans rejet	90,41
Avec rejet	96,97

L'influence de la taille de la base d'apprentissage sur le classifieur permet de formuler deux idées :

- L'augmentation de la taille de la base d'apprentissage permet de diminuer l'erreur tout en augmentant le taux de reconnaissance. Le processus d'apprentissage est donc efficace.
- Avec un jeu de primitives binaires relativement simple et en nombre réduit et une base d'apprentissage de taille modeste ($T = 1000$), il a été possible d'obtenir de bons résultats sur une base de test de taille beaucoup plus importante (ces conditions sont rarement prises dans le domaine de la reconnaissance de chiffres manuscrits hors lignes). Ce système se caractérise par une erreur de reconnaissance faible due à une capacité de rejet importante. Un tel classifieur pourrait donc être utile dans des domaines où la banque de données des observations du phénomène à classifier est pauvre.

Le tableau 4 montre que l'augmentation de la base de données ne provoque pas une augmentation critique du nombre de nœuds de l'arbre. L'arbre de décision obtenu est constitué au maximum de 1036 nœuds (apprentissage sur 6000 chiffres) ce qui est relativement raisonnable si on compare ceci à l'analyse d'un

vecteur de 55 primitives binaires (2^{55} mots possibles). Au niveau implémentation informatique et surtout en terme d'encombrement mémoire, le stockage de 1000 nœuds en mémoire RAM est bien en dessous des capacités des ordinateurs disponibles aujourd'hui. Finalement, il est important de noter que sur les 55 primitives qui ont été fournies au système de reconnaissance, la totalité de ces dernières n'a pas été retenue pour la construction de l'arbre de décision binaire. La figure 8 montre le nombre de primitives retenues et le nombre de fois où elles ont été utilisées en fonction de la taille de la base d'apprentissage.

Dans tous les cas, la totalité des primitives n'a pas été retenue. Par exemple, pour un apprentissage sur 1000 chiffres, seulement 32 primitives ont été utilisées pour la construction de l'arbre de décision binaire. Le vecteur de primitives a donc été diminué dans des proportions importantes. De plus, l'analyse complète d'un vecteur de 55 primitives implique l'extraction de toutes les primitives ce qui est coûteux en temps de calcul. Dans notre approche, le nombre de questions posées par l'arbre de décision binaire pour effectuer la reconnaissance varie entre 4 et $1\frac{1}{2}$ (apprentissage sur 1000 chiffres). Une primitive étant extraite uniquement lorsqu'un nœud demande la valeur booléenne associée à cette dernière, le temps machine alloué à l'extraction des primitives est optimisé

Finalement, il est intéressant de constater qu'une même primitive peut être utilisée très souvent (apprentissage sur 6000 chiffres : primitive numéro 15 (projection) utilisée 34 fois). En fait, à partir d'une primitive de base, l'arbre de décision binaire construit plusieurs primitives qui seront utilisées dans des contextes bien spécifiques de manière à produire une hypothèse de reconnaissance.

6. conclusion

L'apprentissage sur 6000 chiffres a montré qu'il y a création de nœuds impurs. L'analyse des bases de données associées à ces nœuds a mis en évidence des variabilités de l'écriture extrêmes qui produisent des confusions, ou bien des classes qui ont tendance à se ressembler (exemple des chiffres 4-9 ou 0-8 ou 7-1). La baisse du pourcentage de confusion sera possible grâce à l'ajout de nouvelles primitives très spécifiques qui permettront de lever les ambiguïtés restantes.

La structure utilisée pour modéliser l'objet graphique facilite l'extraction de ces primitives. Elle est ouverte et peut donc être utilisée dans d'autres domaines que la reconnaissance de chiffres manuscrits. Il suffit, dans certaines applications, de rajouter aux extracteurs de primitives d'autres plus spécifiques aux formes à reconnaître. La phase d'identification de l'arbre binaire se chargera de retenir les extracteurs les plus pertinents.

Notre critère de rejet, très simple à implémenter, permet d'augmenter de manière significative l'efficacité du système de reconnaissance.

Utilisation des 55 primitives

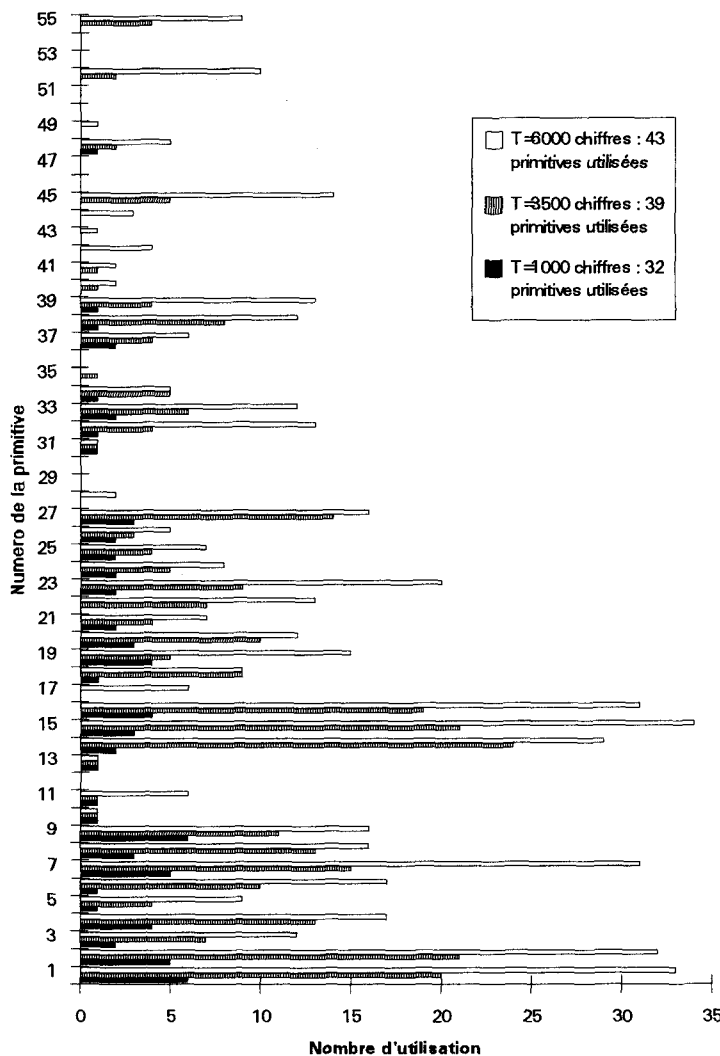


Figure 8. – Fréquence d'utilisation des primitives.

L'augmentation de la base d'apprentissage permet d'améliorer les résultats de reconnaissance et il faudra voir les limites de l'apprentissage en continuant à l'augmenter. Toutefois, il est intéressant de constater qu'il est possible, grâce au critère de rejet introduit, d'obtenir de bons résultats, en utilisant des bases d'apprentissage de tailles réduites. Ce point peut être déterminant dans l'étape de choix d'un classifieur dans une chaîne de reconnaissance où l'on sait que l'on dispose de peu de données pour effectuer la phase d'apprentissage.

L'apprentissage est entièrement automatique et a pu s'accommoder d'une base d'apprentissage très compliquée avec des chiffres parfois difficilement lisibles par un être humain et surtout

mal segmentés dans de nombreux cas. Nous avons donc traité un problème proche de la réalité en assumant d'une part l'extrême variabilité de l'écriture d'un scripteur à un autre et d'autre part le fait que l'étape de segmentation dans une chaîne de reconnaissance n'est jamais fiable à 100%.

Le procédé de sélection des primitives à retenir au niveau des nœuds de l'arbre considère, pour le calcul de l'information transmise, une sous-population de la population totale de chiffres. Nous ne faisons aucune hypothèse d'indépendance des variables aléatoires associées aux primitives (hypothèse très restrictive et souvent peu réaliste).

Nous avons montré qu'il est possible à partir d'un jeu suffisant de primitives de retenir une sous-population permettant de donner de bons résultats sur le problème de reconnaissance. De plus, les primitives sont calculées au fur et à mesure de la phase de reconnaissance. Ceci conduit à utiliser quelques primitives de l'ensemble potentiellement utilisables par le classifieur, pour reconnaître une forme.

Nos travaux futurs s'orientent dans deux directions ayant le même objectif : baisser le taux de rejet sans augmenter les confusions.

La première option est de commander des opérations de nettoyage du chiffre manuscrit lorsqu'il y a rejet pour procéder à une deuxième reconnaissance de la forme, par le classifieur.

La deuxième approche vise à exploiter le mécanisme de classification qui permet d'arriver à un nœud terminal. Il serait intéressant d'utiliser le fait qu'un nœud terminal est le siège d'un contexte de reconnaissance bien spécifique avec des représentants de ce contexte pour développer des opérations visant à valider une hypothèse de reconnaissance lorsqu'il y a rejet. L'arbre de décision binaire serait alors un pré-classifieur très évolué fournissant à un autre classifieur ou « vérificateur », une hypothèse de reconnaissance avec le contexte (représentants du contexte sous la forme d'un groupe de chiffres issus de la base d'apprentissage) qui a permis de formuler cette dernière. Nous travaillons actuellement sur la construction de ce vérificateur.

Remerciement

Nous remercions Monsieur Yves Lecourtier pour ses remarques constructives qui ont permis d'améliorer cet article.

Ce travail est réalisé grâce à des subventions d'un contrat état-région et de crédits Gdr CNRS.

ANNEXE

Nous donnons dans le tableau suivant la correspondance entre les numéros utilisés dans cet article pour désigner les primitives binaires et leurs significations.

Numéro n	Abréviation	Signification	$I(Z; Y_n / B_0)$
1	E1	Présence/absence extrémité	0,36
2	E2	Présence/absence extrémité	0,11
3	E3	Présence/absence extrémité	0,45
4	E4	Présence/absence extrémité	0,14
5	E5	Présence/absence extrémité	0,14
6	E6	Présence/absence extrémité	0,07
7	E7	Présence/absence extrémité	0,43
8	E8	Présence/absence extrémité	0,12
9	E9	Présence/absence extrémité	0,25
10	Oh	Présence/absence boucle haute	0,29
11	Ob	Présence/absence boucle basse	0,26
12	Od	Présence/absence boucle double	0,21
13	Ot	Présence/absence boucle totale	0,37
14	Th	Présence/absence trait horizontal haut	0,35
15	Tm	Présence/absence trait horizontal milieu	0,22
16	Tb	Présence/absence trait horizontal bas	0,3
17	Ec1	Présence/absence carrefour	0,01
18	Ec2	Présence/absence carrefour	0,01
19	Ec3	Présence/absence carrefour	0,06
20	Ec4	Présence/absence carrefour	0,11
21	Ec5	Présence/absence carrefour	0,26
22	Ec6	Présence/absence carrefour	0,21
23	Ec7	Présence/absence carrefour	0,11
24	Ec8	Présence/absence carrefour	0,21
25	Ec9	Présence/absence carrefour	0,06
26	Ca0	Présence/absence de 0 carrefour	0,5
27	Ca1	Présence/absence de 1 carrefour	0,26
28	Ca2	Présence/absence de 2 carrefours	0,19
29	Ca3	Présence/absence de 3 carrefours	0,04
30	Ca>3	Présence/absence de plus de 3 carrefours	0,001
31	O	Présence/absence boucle	0,57
32	O1	Présence/absence ouverture	0,2
33	O2	Présence/absence ouverture	0,33
34	O3	Présence/absence ouverture	0,16
35	O4	Présence/absence ouverture	0,001
36	O5	Présence/absence ouverture	0,001
37	O6	Présence/absence ouverture	0,26
38	O7	Présence/absence ouverture	0,22
39	O8	Présence/absence ouverture	0,16
40	O9	Présence/absence ouverture	0,02
41	O10	Présence/absence ouverture	0,001
42	O11	Présence/absence ouverture	0,01
43	O12	Présence/absence ouverture	0,001
44	O13	Présence/absence ouverture	0,01
45	O14	Présence/absence ouverture	0,21
46	O15	Présence/absence ouverture	0,01
47	O16	Présence/absence ouverture	0,001
48	O17	Présence/absence ouverture	0,02
49	O18	Présence/absence ouverture	0,01
50	O19	Présence/absence ouverture	0,001
51	O20	Présence/absence ouverture	0,001
52	O21	Présence/absence ouverture	0,16
53	O22	Présence/absence ouverture	0,001
54	O23	Présence/absence ouverture	0,001
55	O24	Présence/absence ouverture	0,11

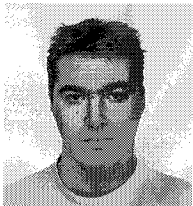
BIBLIOGRAPHIE

- [1] F. Ali et Th. Pavlidis, Syntactic Recognition of handwritten Numerals. IEEE Transaction on Systems, Man and Cybernetics, vol. n°7, pp. 537-541, 1977.
- [2] C. Tappert, C. Y. Suen, T. Wakahana, The state of the art in on-line handwriting recognition. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. PAMI-12, pp. 787-807, 1990.
- [3] M. Wing, Variability in Handwritten Characters. Visible Language, XIII 3, pp 283-298, 1979.
- [4] M. Shridar and A. Badreldin, Recognition of isolated and simply connected handwritten numerals. Pattern Recognition, vol. 19, N° 1, pp. 1-12, 1986.
- [5] P. Gader, B. Forester, M. Ganzberger, A. Gillies, B. Mitchell, M. Whalen and T. Yocum, Recognition of handwritten digits using template and model matching. Pattern Recognition, Vol. 24, N° 5, pp. 421-431, 1991.
- [6] L. Heutte, Reconnaissance de caractères manuscrits : application à la lecture des chèques et des enveloppes postales. Thèse de doctorat, Université de Rouen, 1994.
- [7] C. Y. Suen, C. Nadal, R. Legault, T. A. Mai and L. Lam, Computer Recognition of Unconstrained Handwritten Numerals. Proceedings of the IEEE, Vol. 80, n°7, pp. 1162-1180, 1992.
- [8] C. Roche, Information utile en reconnaissance des formes et en compression de données. Application à la génération automatique de système de reconnaissance optique et acoustique. Thèse d'état ES Sciences Mathématiques, Paris VI, 1972.
- [9] D. de Brucq, K. Romeo, M. Lachkar, N. Feray, F. Bagui, Une méthode d'évaluation du codage en reconnaissance de chiffres manuscrits. 3^{ème} Colloque National sur l'écrit et le document, CNED'94, pp. 19-26, 1994.
- [10] Q. R. Wang and C. Y. Suen, Analysis and Design of a Decision Tree Based on Entropy Reduction and Its Application to Large Character Set Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, n° 4, pp. 406-417, 1984.
- [11] D. Geman, B. Jedynak, Shape Recognition and Twenty Questions. Rapport de recherche INRIA, No. 255, 1993.
- [12] L. Breiman, J. Freidman, R. Olsen and C. Stone, Classification and Regression Trees. Wadsworth, Belmont, CA, 1984.
- [13] S. W. Lee, L. Lam and C. Y. Suen, A systematic evaluation of skeletonization algorithms. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 7, n° 5, pp. 239-261, 1993.
- [14] A. Belaid et Y. Belaid, Reconnaissance des formes. Méthodes et applications. InterEdition, 1992.
- [15] P. Marthon, A. Bruel et G. Biguet, Squelettisation par calcul d'une fonction discriminante sur un voisinage de 8 points. Actes du 2^e congrès AFCET RFIA, pp. 107-117, toulouse, 1979.
- [16] L. Lam and C. Y. Suen, Structural classification and relaxation matching of totally unconstrained handwritten zip-code numbers. Pattern Recognition, Vol. 21, n° 1, pages 19-31, 1988.
- [17] T. Pavlidis, Algorithms for graphics and image processing. Computer Science Press, 1982.
- [18] G. Gaillat et M. Berthod, Panorama des techniques d'extraction de traits caractéristiques en lecture optique des caractères. Revue Technique THOMSON-CSF, Vol. 11, n° 4, pp. 943-959, 1979.
- [19] C. Y. Suen, M. Berthod and S. Mori, Automatic Recognition of Handprinted Characters - The State of the Art. Proceeding IEEE, Vol. 68, n° 4, pp. 469-487, 1980.
- [20] C. Y. Suen, Distinctive features in automatic recognition of handprinted characters. Signal processing, Vol. 4, pp. 193-207, 1982.
- [21] M. Beun, A flexible method for automatic reading of handwritten numéraux. Philips Technical Review, Vol. 33, pp. 89-101 and 130-137, 1973.
- [22] B. Cherabo, L. Massip-Pailhes, S. Castan. Description polygonale basée sur l'extraction d'ilots de confiance. Actes du 8^e congrès AFCET RFIA, pp. 841-846, Lyon-Villeurbanne, 1991.
- [23] L. Miclet, Méthodes structurelles pour la reconnaissance des formes. Edition EYROLLES, 1984.
- [24] A. M. Yaglom, I. M. Yaglom, Probabilité et information. Dunod, pp. 33-89, 1969.
- [25] C. Nadal and C. Y. Suen, Applying human knowledge to improve machine recognition of confusing handwritten numerals. Pattern Recognition, vol 26, n° 3, pp. 381-389, 1993.

Manuscrit reçu le 1er février 1995.

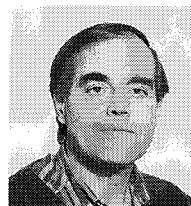
LES AUTEURS

Nicolas FERAY



Nicolas Feray a soutenu une thèse de doctorat en septembre 1995, en traitement de l'image, à l'Université de Rouen. Depuis, il exerce des activités d'enseignement et de recherche au laboratoire PSI-La3I de l'Université de Rouen. Ses travaux portent essentiellement sur la reconnaissance de formes et plus particulièrement sur l'intégration de plusieurs classifieurs dans une même chaîne de reconnaissance.

Denis de BRUCQ



Denis de Brucq est professeur à l'Université de Rouen. Ses travaux sur le filtrage non-linéaire permettent la détection de changements de modèle. Les techniques inventées s'appliquent en reconnaissance de l'écrit. Par ailleurs, en traitement numérique de l'image pour la dermatologie, des méthodes permettant la détection des mélanomes sont mises en œuvre.