Combinaison de réseaux de neurones à convolution pour la reconnaissance de caractères manuscrits en-ligne.

Emilie POISSON[*], Christian VIARD-GAUDIN[*]
et Pierre-Michel LALLICAN[**]

[*]*Ecole polytechnique de l'université de Nantes – IRCCyN UMR CNRS 6597*
Rue Christian Pauc, BP 50609, 44306 NANTES Cedex 3, France
Mél : {Emilie.Poisson ; Christian.Viard-Gaudin} @polytech.univ-nantes.fr

[**]*VISION OBJECTS - 9, rue du Pavillon, 44980 Sainte Luce sur Loire, France*
Mél : pmlallican@visionobjects.com

RÉSUMÉ. *Dans ce papier, nous étudions différentes techniques et couplages de réseaux de neurones à convolution dédiés à la reconnaissance de caractères manuscrits isolés en-ligne. Nous détaillons plusieurs réseaux à convolution, un TDNN traitant les données en-ligne, un SDNN travaillant sur les données hors-ligne et une architecture hybride appelée SDTDNN couplant les informations dynamiques et statiques. Nous apportons des premiers résultats déjà prometteurs sur ces différentes architectures. Les bases de tests utilisées sont les bases UNIPEN pour le TDNN et le SDTDNN, la base MNIST pour le SDNN.*

ABSTRACT. *In this paper, we study the techniques and coupling of convolutional neural networks for online isolated handwritten character recognition. Three architectures are investigated, a TDNN that processes online features, a SDNN that relies on the offline bitmaps of the character and an hybrid architecture called SDTDNN coupling the dynamic information and the static one. We produced first some already promising results on these various architectures. Results on UNIPEN database are reported for the TDNN online recognition system, while the MNIST database has been used for the SDNN offline classifier.*

MOTS-CLÉS : *réseaux à convolution, TDNN, SDNN, couplage de données en-ligne/hors-ligne, Reconnaissance de caractères manuscrits en ligne.*

KEYWORDS: *Convolutional Neural Networks, TDNN, SDNN, combination of online and offline information, online handwritten characters recognition.*

1. Introduction

Nos travaux s’intègrent dans le contexte de la reconnaissance de l’écriture en-ligne destiné aux systèmes mobiles communicants (assistant numérique personnel, ardoise électronique, smart-phone). Dans ce domaine, il importe encore d’améliorer les performances de reconnaissance tout en respectant des contraintes fortes sur le nombre de paramètres à stocker et la vitesse de traitement..

Un des premiers objectifs de nos travaux est d’optimiser une architecture de réseaux moins conventionnelle qu’un MLP et permettant une très grande robustesse aux déformations et perturbations. Dans cette optique, nous avons opté pour l’étude, le développement et le test d’un réseau de neurones à convolution. En effet, comme le souligne, le récent article [LeCun et al.,2001] ceux-ci présentent de remarquables propriétés pour prendre en compte directement des formes 2D en s’affranchissant de l’étape toujours délicate d’extraction de caractéristiques pertinentes.

Un deuxième objectif, dans le cadre d’un système de reconnaissance en-ligne, est d’étudier l’apport et la complémentarité de la représentation statique du tracé par rapport au signal dynamique de celui-ci. En effet, deux trajectoires différentes du stylet peuvent conduire à la même forme graphique, dans ce cas, la représentation statique sera plus robuste, et inversement un même caractère peut avoir des représentations graphiques distinctes qui soient produites par des mouvements très voisins, donnant cette fois un avantage à la représentation dynamique. On peut espérer dans ces conditions qu’une approche combinant les deux types d’information permette d’améliorer les performances de reconnaissance [Alimoglu et al., 1997]. C’est ce que nous étudions dans les différentes expériences que nous avons menées.

Pour permettre ce couplage, une architecture modulaire a été construite, elle autorise un fonctionnement complètement paramétrable : en simple MLP, en réseau à convolutions sur les données statiques ou sur les données en-ligne, avec couplage en sortie, avec couplage en interne sur les couches cachées.

2. Les réseaux à convolution

Les réseaux à convolution sont dérivés des architectures de type perceptron multi-couches (Multi Layer Perceptron : MLP), cependant ils utilisent des poids partagés, liés à la fenêtre de convolution, qui leur permettent une extraction implicite de caractéristiques locales.

Pour bien souligner la différence des réseaux de neurones à convolution par rapport aux réseaux classiques de type MLP, analysons le principe de reconnaissance sur le caractère "a", Fig.1. Un neurone d'un MLP est entièrement connecté à tous les neurones de la couche précédente tandis que pour un réseau à convolution, un neurone est connecté à un sous-ensemble de neurones de la couche précédente. Ainsi

chaque neurone peut être vu comme une unité de détection d'une caractéristique locale, d'une singularité structurelle particulière telle que la détection d'un trait vertical ou horizontal, voire d'une boucle. Le long de la trajectoire, la matrice des poids correspondant à la fenêtre glissante est identique (notion de poids partagés) : même détection, même convolution.

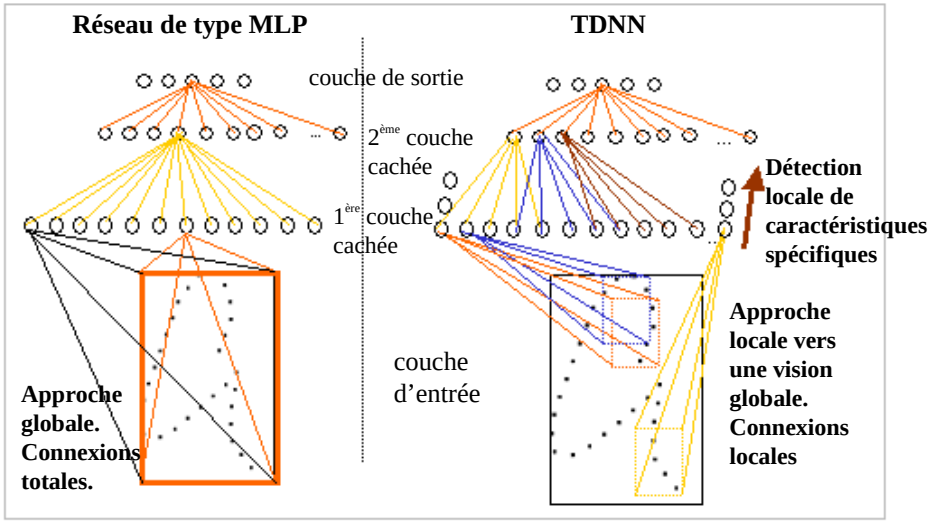

Figure 1. *Illustration de la différence Perceptron/TDNN sur le caractère 'a'*

L'utilisation des poids partagés réduit le nombre de paramètres dans le système facilitant alors la généralisation. Ce type de réseau a été appliqué avec succès à la reconnaissance des chiffres [Guyon et al., 1995]. Avant d'introduire une architecture modulaire basée sur la complémentarité des réseaux à convolution temporelle et spatiale, une étude de leurs caractéristiques et performances intrinsèques est menée.

2.1. Le TDNN

Le TDNN est un réseau de neurones à décalage temporel introduit en reconnaissance de la parole puis transposé pour des données de nature séquentielle [LeCun et al., 1995, il est donc adapté à la reconnaissance d'écriture manuscrite en-ligne. Dans [Poisson et al., 2001], nous détaillons l'étude de la topologie du réseau : taille de la fenêtre d'analyse, nombre de couches, contraintes sur le partage des poids, algorithme d'apprentissage ($1^{er}/2^{nd}$ ordre). L'architecture retenue du TDNN comporte deux parties principales (Fig. 2). La première, correspondant aux couches basses, implémente les convolutions successives permettant de transformer progressivement

une séquence de vecteurs caractéristiques en une autre séquence de vecteurs caractéristiques d'ordre supérieur. La seconde correspond à un MLP classique, elle reçoit en entrée l'ensemble des sorties de la partie TDNN. Ces deux blocs sont dans notre implémentation complètement paramétrables.

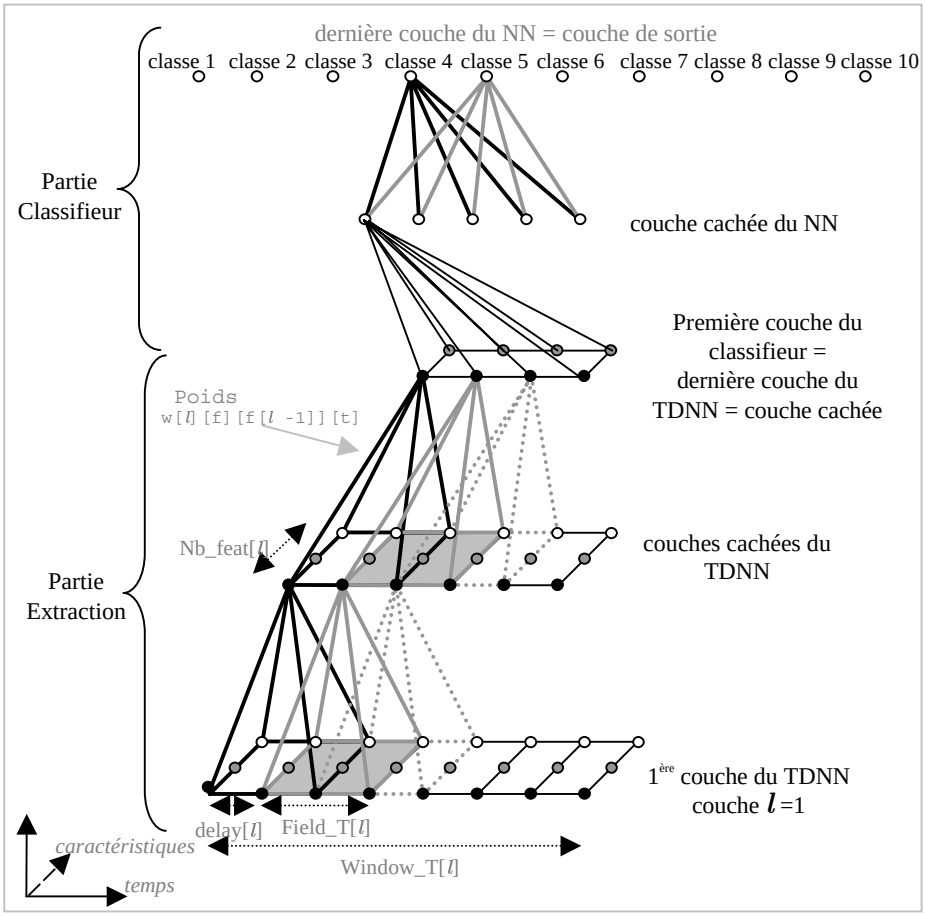

Figure 2. *Architecture du TDNN*

Les données en-lignes utilisées (bases Unipen [Guyon et al., 1994]) ont été dans un premier temps ré-échantillonnées de sorte à s'affranchir de la vitesse du tracé et afin obtenir des échantillons avec un nombre fixe de points (50). Ensuite, 7 caractéristiques normalisées sont extraites en chaque point : position (2), direction

(2), courbure(1), état du stylet (1), cf. Fig. 3. Enfin, le réseau est entraîné sur la base d'apprentissage avec une technique classique basée sur un gradient stochastique, celle-ci donnant, d'après les tests que nous avons pu mener à bien, d'aussi bons résultats qu'une méthode du second ordre.

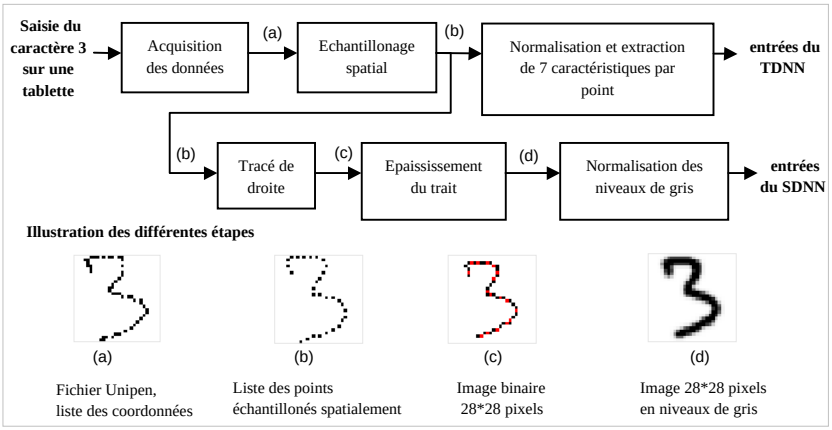

Figure 3. *Illustration des pré traitements des entrées du TDNN et du SDNN*

Base UNIPEN	Nb. de Classes	Nb. d'ex. en apprentissage	Nb. d'ex. en test	**TDNN Taux de reconnaissance sur la base de test**	**MLP Taux de reconnaissance sur la base de test**
Chiffres	10	10 423	5 212	**97,9**	**97,5**
Minuscules	26	34 844	17 423	**92,8**	**92,0**
Majuscules	26	17 736	8 869	**93,5**	**92,8**

Tableau 1. *Performances comparées d'un TDNN et d'un MLP*

Nous présentons, tableau 1, les performances comparées obtenues avec les meilleures configurations trouvées à la fois pour des architectures de type MLP et TDNN. On peut souligner les performances significativement supérieures obtenues par le TDNN et ce sur les trois sous-ensembles : chiffres, minuscules, majuscules. En effet, celui-ci permet de réduite le taux d'erreur de 16 % sur les caractères chiffres à près de 10 % sur les minuscules. Par ailleurs, il est important de préciser que l'architecture TDNN, grâce à sa contrainte des poids partagés, demande une moindre capacité de stockage pour ses paramètres. On ramène, par exemple, le nombre de coefficients du MLP-chiffre de 36 110 (100 neurones sur la couche cachée) à 17 930 pour le TDNN-chiffre, (fenêtre de largeur 20, délai de 5, 20

caractéristiques locales, 100 neurones pour la couche cachée du classifieur) soit une diminution d'un facteur supérieur à 2. Cela confère au TDNN sur le plan implémentation un avantage certain pour les applications embarquées, de plus, il est établi qu'à performances égales (même biais), plus un système est simple, meilleures sont ses capacités de généralisation (variance moins élevée) [Bishop, 1995].

2.2 Le SDNN

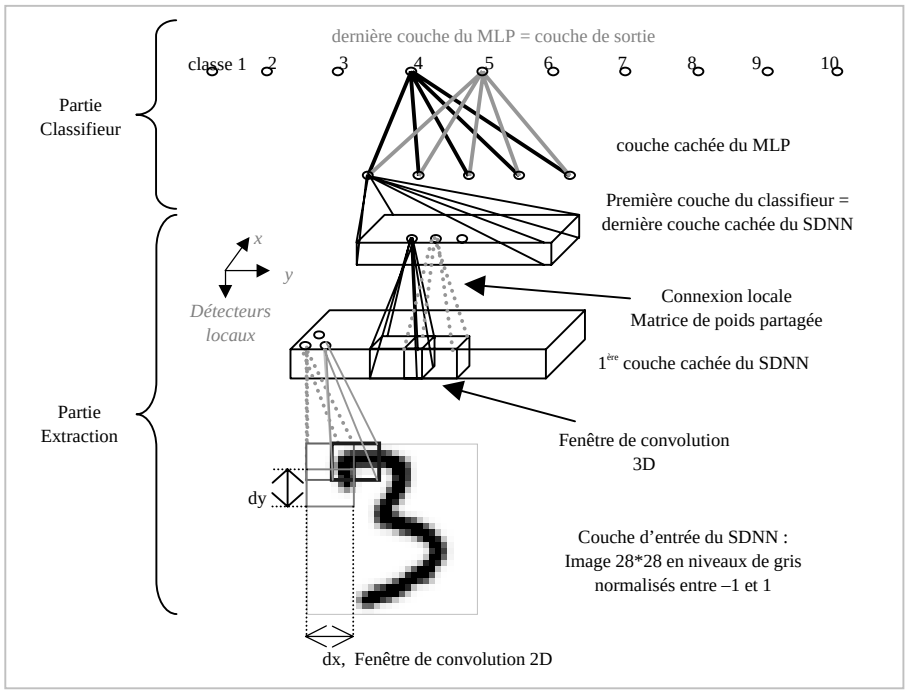

Figure 4. *Architecture du SDNN*

Avec le TDNN, la nature temporelle des données est exploitée par le système de reconnaissance, cela permet souvent de lever des ambiguïtés et d'identifier plus facilement certains caractères. A l'inverse, certains ordonnancements temporels sont perturbateurs, en particulier des signes diacritiques ou des retouches faites sur un tracé. Dans ce cas la représentation picturale est plus stable, c'est donc ce type de signal que se propose d'exploiter le SDNN. Le SDNN est un réseau de neurones à convolution qui n'exploite plus la notion temporelle mais s'intéresse aux positionnements spatiaux des données. Le SDNN est une généralisation du TDNN à une topologie 2D. La figure 4 illustre la structure typique de ce réseau.

Les méta-paramètres à fixer pour ce réseau concernent la taille de la fenêtre de convolution, le décalage spatial, le nombre de caractéristiques locales, le nombre de couches cachées pour la partie extracteur et pour la partie classifieur. Ceux-ci ont été déterminés expérimentalement, le meilleur compromis a été obtenu avec 2 couches cachées, (fenêtre de 6*6, un décalage de 2 pour chaque couche), 20 caractéristiques locales et un classifieur linéaire. Ces expérimentations ont été conduites sur les caractères isolés de la base hors-ligne MNIST [LeCun et al., 2001], l'entrée du réseau correspondant à une image 28*28 en niveaux de gris normalisés entre [-1,1].

Réseau	Nombre de poids	Taux de reconnaissance en 1ère position	
		Base d'apprentissage	Base de test
MLP Sur les pixels	159 010	99,4%	98,2%
MLP Caractéristiques [Tay, 2002]	36 610	99,2%	98,6%
SDNN LeNet5 (pixels)	60 000	-	99,05%
SDNN proposé (pixels)	**18 370**	**99,9%**	**98,5%**

Tableau 2. *Performances sur la base MNist (60000 chiffres en appr., 10000 en test)*

Les résultats obtenus (tableau 2) vont dans le même sens que pour le TDNN, d'une part des performances sensiblement supérieures à celles d'un MLP avec une diminution importante du nombre de poids du réseau, ce qui pour nous est un enjeu majeur pour des applications portables de faibles capacités.

Nous cherchons en fait à utiliser ce reconnaisseur hors-ligne pour l'appliquer à des données disponibles originellement sous formes de séquences de points (bases Unipen). Il convient pour cela de synthétiser les images à partir des trajectoires. Cette transformation est évidemment plus aisée que la transformation inverse [Lallican et al., 2000], la figure 3 illustre les différentes étapes mises en œuvre pour cela. Nous pouvons dès lors utiliser les mêmes bases pour le TDNN et le SDNN.

3. Croisement des performances du TDNN et du SDNN

KO	110 (2,1 %)	**38** (0,7 %)	72 (1,4 %)
OK	5 102 (97,9 %)	4 937 (94,8 %)	**165** (3,1 %)
	TDNN \ SDNN	4 975 (95,5 %)	237 (4,5 %)
	Total : 5212 ex	OK	KO

Tableau 3. *Performances croisées du SDNN et du TDNN sur la base Unipen-chiffre*

Le TDNN et le SDNN offrent des performances de reconnaissance très intéressantes séparément, il est intéressant d'étudier le comportement respectif de l'un vis-à-vis de l'autre et de dégager les gains potentiels que l'on peut attendre d'un couplage des deux systèmes. Le tableau 3 présente la répartition des échantillons de la base de test en sortie des deux reconnaisseurs suivant qu'ils soient bien reconnus (OK) ou non (KO). Nous constatons effectivement une complémentarité intéressante due à l'apport des deux types d'informations celle dynamique et celle spatiale.

4. Coopération des informations en-ligne et hors-ligne

Deux techniques de couplage ont été testées. Tout d'abord nous avons employé un simple couplage en sortie des deux configurations précédemment définies en réalisant une moyenne géométrique des sorties, ensuite, nous proposons une coopération via les couches cachées, cf Fig. 5.

4.1. Combinaison en sortie

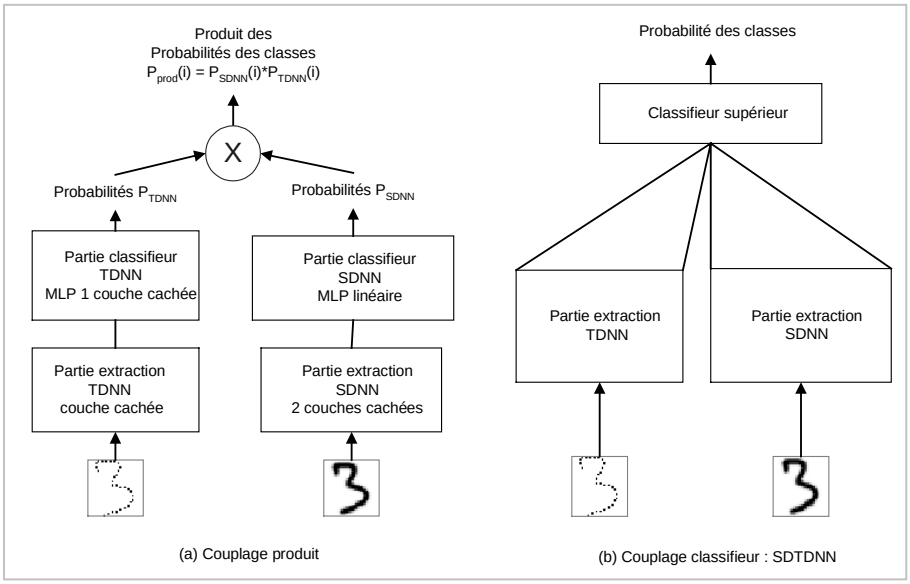

Figure 5. *Coopération en sortie ou par les couches cachées*

Cette configuration, appelée couplage produit, calcule le produit des sorties des réseaux TDNN et SDNN entraînés séparément. Les sorties de chaque réseau correspondent aux probabilités a posteriori des classes Prob(C|O), obtenues par la fonction de transfert de type Softmax des neurones de la couche de sortie.

Le tableau 4 montre l'intérêt de coupler des informations de types statiques et dynamiques. Le couplage produit permet de réduire de près de 10% le taux d'erreur sur la base de test chiffres Unipen. On peut constater sur ce tableau que le couplage produit perd 27 exemples reconnus par le TDNN (18) ou le SDNN (9), il sera intéressant d'analyser par la suite une pondération des sorties du réseau, pondération du TDNN par rapport au SDNN. Par ailleurs une analyse fine des prétraitements possibles des données en-ligne vers des données hors-lignes conduira à diminuer largement cette perte et accroître significativement le nombre d'éléments reconnus par le couplage produit et non reconnu par le TDNN et SDNN.

	Rec. TDNN et Rec. SDNN	Rec. TDNN Non SDNN	Non TDNN Rec. SDNN	Non TDNN Non SDNN	Total
Reconnu Produit	4 937 (94,8 %)	147 (2,8%)	29 (0,5%)	3 (0,1%)	5 116 (**98,2%**)
Non Reconnu Produit	0	18 (0,3%)	9 (0,2%)	69 (1,3%)	96 (1,8%)
Total	4937(94,8%)	165 (3,1%)	38 (0,7%)	72 (1,4%)	5212

Tableau 4. *Exemples-chiffres Unipen acquis ou perdus par un couplage produit*

4.2. Combinaison SDTDNN

Pour que le couplage des deux systèmes bénéficie d'un apprentissage global unique, nous avons construit une architecture modulaire dite SDTDNN, Space Displacement and Temporal Delay Neural Network. Cette structure (Fig. 5.b) est composée d'une partie inférieure d'extraction de caractéristiques et d'une partie supérieure de classification unique (MLP linéaire) donnant en sortie les probabilités a posteriori des classes. La première partie correspondant aux couches basses intègre deux modules : un module TDNN traitant les caractéristiques en-ligne et un module traitant l'image correspondante. La dernière partie combine les caractéristiques en-ligne et hors-ligne pour affecter à chaque classe une probabilité d'appartenance.

Base Unipen	TDNN 20/5/20 MLP 100		SDNN 6/2/20 – 6/2/20 MLP lin.		Couplage Produit		SDTDNN	
	Poids	%	Poids	%	Poids	%	Poids	%
Chiffres	17 930	97,9	18 370	95,5	36 300	98,2	13 392	97,9
Minuscules	19 546	92,8	23 506	86,6	43 052	93,0	25 248	92,8
Majuscules	19 546	93,5	23 506	89,5	43 052	94,4	25 249	93,6

Tableau 5. *Comparaison des performances des différents réseaux*

Les résultats obtenus (Cf. tableau 5) montrent que si l'on souhaite améliorer les performances de reconnaissance, c'est le couplage produit qui donne les meilleurs

résultats. Cela se paye évidemment par une augmentation sensible du nombre des paramètres puisque les systèmes travaillent en parallèle. A l'inverse, il est possible de réduire le nombre de paramètres en utilisant l'architecture SDTDNN proposée tout en conservant le même niveau de performances qu'avec le TDNN seul.

5. Conclusion

Nous avons présenté une nouvelle architecture multi-modulaire basée sur les réseaux de neurones à convolution, destinée à être intégrée sur des systèmes mobiles de faibles capacités. Notre étude a porté sur l'analyse des performances individuelles, puis couplées de ces réseaux – TDNN, SDNN, SDTDNN.

Les résultats présentés ici montrent que cette architecture offre un bon compromis performance/complexité dans le cadre des applications visées. Nous pensons pouvoir encore améliorer ce compromis et envisageons d'étendre son utilisation à une reconnaissance de mots cursifs en-ligne.

Références :

F. Alimoglu, E. Alpaydin : "Combining Multiple Representations and Classifers for Pen-based Handwritten Digit Recognition", *ICDAR'97*, p. 637-660, Ulm, Août 1997.

C.M. Bishop : *« Neural Networks for Pattern Recognition »*, Oxford University Press. ISBN 0-19-853849-9, p. 116-161, 1995.

I. Guyon, L. Schomaker, S. Janet, et al. : "First UNIPEN benchmark of on-line handwriting recognizers organized by NIST". Technical Report BL0113590-940630-18TM, AT&T Bell Laboratories, 1994.

I. Guyon, et al. : "Pénacée : A Neural Net System for Recognizing On-line Handwriting", In E. Domany, J. L. van Hemmen, and K. Schulten, editors, *Models of Neural Networks*, volume 3, p. 255-279, Springer, 1995.

P-M. Lallican, C. Viard-Gaudin, S. Knerr, « From Off-line to On-line Handwriting Recognition », *IWFHR'2000*, Amsterdam, Netherlands, p. 303-312, Sept. 11-13, 2000.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Intelligent Signal Processing*, p. 306-351, 2001.

Y. LeCun and Y. Bengio, "Convolutional Networks for Images, Speech, and Time-Series," in *The Handbook of Brain Theory and Neural Networks*, (M. A. Arbib, ed.), 1995.

E. Poisson, C. Viard-Gaudin, « Réseaux de neurones à convolution : reconnaissance de l'écriture manuscrite non contrainte », *Valgo 2001* (ISSN 1625-9661), n° 01-02, Oct. 2001.

Y.H. Tay, "Off-line Handwriting Recognition using artificial Neural Network and Hidden Markov Model"- Thèse de l'université de Nantes et de l'université de Technologie de Malaisie, Mars 2002.