Les caractères ne sont pas la clef des champs

Anne-Laure Bianne[1,2], Christopher Kermorvant[1], Patrick Marty[1], Farès Menasri[1]

[1] A2iA, Articifial Intelligence and Image Analysis,
40 bis, rue Fabert, 75007 Paris
http://www.a2ia.com

[2] Équipe Multimédia, Laboratoire TSI, Télécom ParisTech,
37-39 rue Dareau, 75014 Paris
http://www.tsi.enst.fr

Résumé : Cet article contribue à une meilleure compréhension de deux aspects de l'utilisation de la reconnaissance automatique de caractères manuscrits dans une application industrielle. Dans un premier temps, nous comparons le comportement de trois reconnaisseurs à l'état de l'art sur trois bases de données de chiffres isolés. Nous montrons ainsi qu'il n'est pas possible d'extrapoler les résultats obtenus sur des bases réalistes mais non bruitées à des données réelles bruitées. Ensuite, nous présentons et évaluons un système conçu pour la reconnaissance de champs de caractères, une application réelle. Nous montrons que le succès d'une telle application dépend d'éléments autres que le taux de reconnaissance caractère.

Mots-clés : reconnaissance de caractères isolés, reconnaissance de champs flottants, réseaux de neurones à convolutions, automates à états finis pondérés.

1 Reconnaissance de chiffres manuscrits isolés

Dans cette section, nous utilisons 3 bases de chiffres isolés, 2 bases réalistes mais peu bruitées (MNIST, USPS) et une base réelle et bruitée (A2iA-AV-Digit) afin d'évaluer trois classifieurs de l'état de l'art sur une tâche de reconnaissance de chiffres. L'objectif de cette étude était d'évaluer dans quelle mesure les résultats obtenus sur une base réaliste mais peu bruitée comme MNIST se généralisent à une base réelle et bruitée.

Les bases de données considérées pour cette étude sont les suivantes :

MNIST : base de chiffres manuscrits isolés construite par LeCun *et al.* (1998), très utilisée pour l'évaluation d'algorithmes d'apprentissage (60 000 exemples d'apprentissage, 10 000 exemples de test).

Extended-MNIST : version étendue de MNIST dans laquelle la base d'apprentissage a été augmentée en appliquant des transformations aléatoires à chacune des 60 000 images selon la procédure proposée par Simard *et al.* (2003) (300 000 exemples d'apprentissage, 10 000 exemples de test).

	ConvNet	SVM-RBF	kppv-IDM
USPS	4.78 ± 0.93	4.68 ± 0.92	3.29 ± 0.78
MNIST	0.88 ± 0.18	1.43 ± 0.23	0.86 ± 0.18
Extended-MNIST	0.65 ± 0.16	0.92 ± 0.19	-
A2iA-AV-Digit-60k	2.27 ± 0.29	9.67 ± 0.58	11.57 ± 0.63
A2iA-AV-Digit	1.14 ± 0.21	5.12 ± 0.43	-

TAB. 1 – Taux d'erreur des trois classifieurs (avec intervalle de confiance binômial à 0.05%) sur les ensembles de test des différentes bases de données.

USPS : base de chiffres manuscrits isolés (7291 exemples d'apprentissage et 2007 exemples de test).

A2iA-AV-Digit : images de chiffres issues de formulaires fiscaux (240 257 exemples d'apprentissage et 10 000 exemples de test). Cette base se caractérise par une mauvaise qualité de l'image et la présence importante de bruit.

A2iA-AV-Digit-60k : afin d'obtenir une sous-base comparable à la base d'apprentissage de MNIST, nous avons sélectioné 60 000 exemples parmi les exemples d'apprentissage de la base A2iA-AV-Digit.

Les classifieurs testés sont les suivants :

SVM : un classifieur SVM à noyau RBF entraîné avec la librairie LibSVM[1].

K-ppv avec des modèles de distorsion d'images : nous avons construit un classifieur de type k Plus Proches Voisins (k-ppv) basé sur l'algorithme IDM de Keysers *et al.* (2007) en utilisant le logiciel W2D fourni par C. Gollan[2] et en appliquant le protocole expérimental recommandé par ce dernier.

Réseau de neurones à convolutions nous avons entraîné un réseau de neurones à convolutions selon l'architecture proposée par LeCun *et al.* (1998) en remplaçant le classifieur gaussien par un perceptron multi-couches.

Résultats

Les résultats des trois classifieurs sur les 5 bases de données sont présentés dans le Tableau 1. Le résultat important est que le taux d'erreur obtenu pour la base bruitée A2iA-AV-Digit-60k est beaucoup plus élevé que celui pour MNIST. Deux classifieurs, le Réseau à Convolutions et le k-ppv-IDM, qui obtiennent des résultats similaires sur MNIST (moins de 1% d'erreurs) , obtiennent respectivement 2.27% et 11.57% d'erreurs sur A2iA-AV-Digit-60. Il est cependant intéressant de noter que lorsque le nombre de données à l'entraînement augmente, les taux d'erreur décroissent (Extended-MNIST, A2iA-AV-Digit). Ces résultats montrent bien qu'il est impossible d'extrapoler les résultats en reconnaissance obtenus sur MNIST ou USPS à une base de données de chiffres plus bruitée. Nous pensons que ces résultats devraient encourager la communauté à être plus critique vis-à-vis de l'utilisation de la base MNIST ou d'autres bases similaires pour évaluer les algorithmes.

[1] http ://www.csie.ntu.edu.tw/˜cjlin/libsvm/

[2] http ://www-i6.informatik.rwth-aachen.de/˜gollan/w2d.html

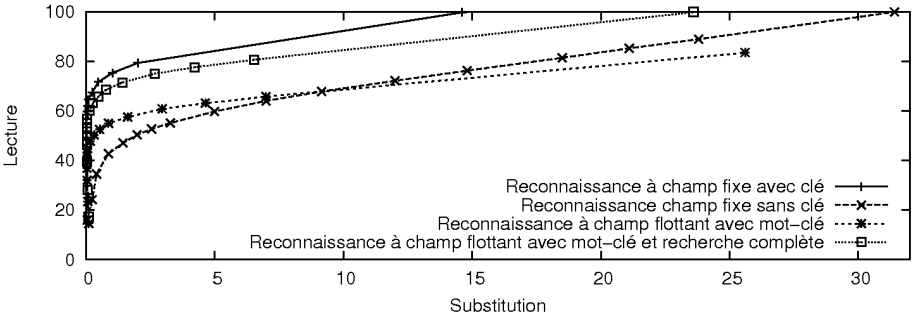

FIG. 1 – Reconnaissance de champs fixes et de champs flottants : taux de lecture vs taux de substitution pour quatre différents protocoles expérimentaux

2 Reconnaissance de champs flottants manuscrits

Dans les applications industrielles, la reconnaissance de chiffres isolés n'est jamais une tâche en soi, mais fait toujours partie d'un problème plus complexe. L'objectif de cette seconde étude était de mettre en évidence les différents aspects nécessaires à la construction d'une application réelle basée sur la reconnaissance de caractères isolés. L'application visée était la reconnaissance d'un champ contenant un code fiscal à format fixe.

Description du reconnaisseur de champs

Nous avons construit un reconnaisseur de champs basé sur un reconnaisseur de caractères de type Réseau de Neurones à Convolutions présenté dans la section précédente.

La reconnaissance d'un champ se déroule comme suit : l'imagette du champ est divisée en 16 imagettes de même largeur. Pour chaque imagette, le reconnaisseur calcule la probabilité *a posteriori* de chaque classe de caractère. La séquence des 16 caractères qui composent le code fiscal est modélisée par un transducteur à états finis pondérés (WFST) (Allauzen *et al.*, 2007). L'expression régulière qui correspond aux formes valides du code fiscal est convertie en un WFST. Le WFST résultant de la composition de ces deux WFST permet de modéliser les séquences de chiffres reconnues respectant les contraintes syntaxiques du code fiscal. Nous avons extrait les N meilleurs chemins du transducteur (N=100 dans nos expériences) et avons filtré les codes reconnus avec la clef de contrôle. Le résultat du reconnaisseur de champ est le code reconnu, filtré et avec la probabilité la plus grande.

Résultats

Les résultats de reconnaissance sont donnés sur la Figure 1 par une courbe présentant la *lecture* (rapport du nombre de résultats dont le score est supérieur au seuil de rejet sur le nombre total de champs à reconnaître) en fonction de la *substitution* (rapport du nombre de résultats incorrects dont le score est supérieur au seuil sur le nombre de

résultats dont le score est supérieur au seuil).

Dans un premier protocole expérimental, la localisation du champ est donnée par l'annotation (champ fixe). Ces expériences mettent en évidence deux aspects essentiels pour la construction d'un reconnaisseur de champs :

La clef de validation : son utilisation permet de réduire le taux d'erreur sur l'ensemble des documents de 31.4% à 14.6%.

Le score de confiance : en jouant sur le seuil de rejet basé sur le score de confiance, il est possible de descendre jusqu'à 0.01% d'erreurs pour un taux de lecture à 65%,

Dans un second protocole expérimental, la localisation de la position du champ est automatique (champ flottant). Ces expériences montrent que la localisation automatique du champ a un impact important sur les performances :

Localisation avec mot-clé : pour un taux de substitution de 15%, le taux de lecture est de seulement 73% alors qu'il est de 100% en reconnaissance de champs fixes.

Localisation avec mot-clé et recherche complète : dans le cas ou aucun mot-clé n'est trouvé pour la localisation du champ, le reconnaisseur est utilisé pour une recherche complète dans une zone prédéfinie, permettant d'obtenir, pour un taux de substitution faible (<1%), un taux de lecture inférieur de seulement 5% au taux de lecture d'un champ avec la localisation annotée.

3 Conclusion

Dans cet article, nous avons revisité deux aspects de la reconnaissance de caractères. Premièrement, nous pensons que les bases de données actuelles de caractères isolés sont trop simples et utilisées depuis trop longtemps dans la communauté pour refléter l'état de l'art de la reconnaissance de caractères pour des applications industrielles. Nos expériences montrent que le taux d'erreur peut augmenter radicalement d'environ 1% pour MNIST à plus de 10% sur une base réelle bruitée. Deuxièmement, nous avons décrit et évalué un système basé sur la reconnaissance de caractères isolés et destiné à la reconnaissance de champs flottants. Nous avons montré que, outre le taux de reconnaissance de caractères, plusieurs autres composants permettent de réduire le taux d'erreur à un niveau acceptable pour une application industrielle.

Références

ALLAUZEN C., RILEY M., SCHALKWYK J., SKUT W. & MOHRI M. (2007). Openfst : a general and efficient weighted finite-state transducer library. In *Proceedings of the Conference on Implementation and Application of Automata*, p. 11–23.

KEYSERS D., DESELAERS T., GOLLAN C. & NEY H. (2007). Deformation models for image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**(8), 1422–1435.

LECUN Y., BOTTOU L., BENGIO Y. & HAFFNER P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**(11), 2278–2324.

SIMARD P., STEINKRAUS D. & PLATT J. (2003). Best practice for convolutional neural networks applied to visual document analysis. In *International Conference on Document Analysis and Recognition*, p. 958–962.