Reconnaissance automatique de texte dans des vidéos à l'aide d'un OCR et de connaissances linguistiques

Khaoula ELAGOUNI[1,2], Christophe GARCIA[3], Pascale SÉBILLOT[2]

[1]Orange Labs R&D
4 rue du Clos Courtel, 35112 Cesson-Sévigné Cedex, France

[2]IRISA, INSA de Rennes
Campus de Beaulieu, 35042 Rennes Cedex, France

[3]LIRIS, INSA de Lyon
17 avenue Jean Capelle, 69621 Villeurbanne Cedex, France

khaoula.elagouni@orange-ftgroup.com, christophe.Garcia@liris.cnrs.fr
pascale.sebillot@irisa.fr

Résumé – Cet article traite de l'extraction automatique d'éléments textuels incrustés dans des vidéos afin de décrire sémantiquement leur contenu. Pour ce faire, nous avons développé un OCR (*Optical Character Recognition*) vidéo, spécifiquement adapté pour détecter et reconnaître les textes incrustés. Reposant sur une approche neuronale, notre méthode se distingue par sa robustesse à la variabilité de styles et de tailles, à la complexité du fond et aux faibles résolutions de l'image. Nous introduisons également un modèle de langue qui pilote l'OCR vidéo afin de lever les ambiguïtés de la reconnaissance et réduire les erreurs de segmentation. L'approche, évaluée sur une base de journaux télévisés français, a obtenu des taux de reconnaissance de caractères de 95%, offrant ainsi la possibilité d'alimenter un système d'indexation de vidéos.

Abstract – Our work aims at helping multimedia content understanding by extracting textual clues embedded in digital video data. For this, we developed a video Optical Character Recognition (OCR) system, specifically adapted to detect and recognize embedded texts. Based on a neural approach, our method outperforms related work especially in terms of robustness to style and size variability, to background complexity and to low resolution of the image. We also introduced a language model that drives several steps of the video OCR in order to remove ambiguities related to recognition and reduce segmentation errors. This approach has been evaluated on a database of French TV news videos and achieves a character recognition rate of 95%, which enables its incorporation in a video indexing system.

1 Introduction

Avec le développement de nouveaux systèmes d'acquisition d'images et l'avènement des services de partage de vidéos, l'indexation automatique de documents multimédias est devenue cruciale pour gérer ces vastes collections. L'enjeu majeur consiste à extraire l'information pertinente permettant de résumer les contenus et retrouver les documents. Durant ces dernières années, certains travaux ont opté pour la prise en compte des textes incrustés dans les vidéos comme moyen d'accès à (une partie de) la sémantique de leurs contenus. Dans les recherches pionnières de Lienhart *et al.* [7], un OCR (*Optical Character Recognition*) commercial, peu adapté aux spécificités des vidéos, est utilisé pour reconnaître les textes détectés, ce qui conduit à des résultats jugés peu satisfaisants. Les approches suivantes [2, 11] se sont intéressées à des prétraitements utiles pour l'amélioration des performances d'extraction, toujours à l'aide d'OCR commerciaux. Une phase de binarisation permettant de séparer le texte du fond a ainsi été introduite. La mise au point d'OCR spécifiques pour la reconnaissance de caractères dans les vidéos a également fait l'objet de plusieurs travaux, conduisant à deux types d'OCR : ceux reposant sur la mise en correspondance de formes [2, 9] et ceux construits par apprentissage supervisé [4, 8]. Dans le premier type, l'enjeu principal consiste à définir des primitives qui représentent précisément les caractères, ce qui induit une variation considérable des performances selon les primitives retenues. Les méthodes du second type conduisent en revanche à des performances meilleures et sont le cadre de notre travail.

Dans cet article, nous proposons un système complet d'OCR spécifiquement adapté aux vidéos, qui permet de détecter et de reconnaître les textes incrustés. Outre l'efficacité et la robustesse de notre méthode de reconnaissance de caractères fondée sur une approche de classification neuronale, notre seconde contribution réside dans l'introduction d'un mode de supervision reposant sur un modèle de langue, qui pilote le système OCR et prend en compte le contexte lexical. Après avoir détaillé notre système complet d'OCR en section 2, nous décrivons et discutons, en section 3, les résultats du test de l'intégralité de la chaîne de traitement sur une base de vidéos réelles de journaux télévisés. La section 4 rappelle, quant à elle, nos contributions et ouvre des perspectives.

2 Un système complet de reconnaissance de texte

La figure 1 décrit les différentes parties de notre système. Les détails relatifs à chaque phase ainsi que leurs interactions sont présentés dans les sous-sections suivantes.

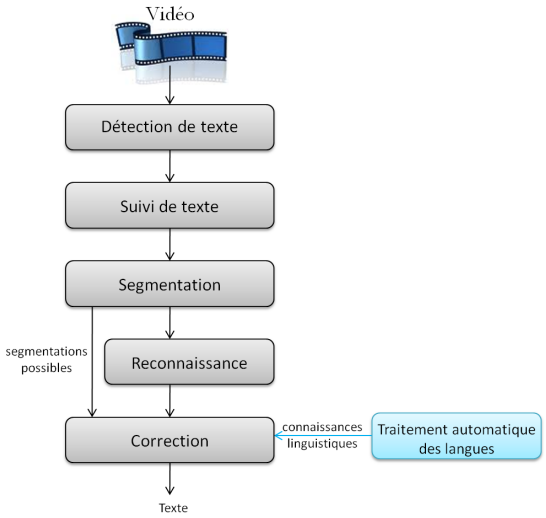

FIGURE 1 – Schéma synoptique de l'OCR vidéo proposé.

2.1 Détection et suivi de texte

La première étape de notre système consiste à détecter les textes incrustés dans les vidéos. Delakis et Garcia [3] ont proposé une solution efficace et très robuste qui repère les textes horizontaux incrustés dans les images et repose sur un modèle neuronal. Nous avons donc choisi d'adapter cette méthode au contexte de la vidéo. Pour cela, nous appliquons une détection toutes les 2 secondes [1] afin de repérer les nouveaux textes qui vont apparaître. Les textes traités étant statiques, nous utilisons la corrélation d'intensités comme mesure de similarité visuelle afin de déterminer, pour chaque texte détecté, les instants exacts de son apparition et de sa disparition.

2.2 Segmentation de caractères fondée sur l'algorithme du plus court chemin

Avant la phase de reconnaissance des textes détectés, une étape préliminaire de segmentation est nécessaire afin d'obtenir des images correspondant chacune à un caractère.

2.2.1 Analyse statistique des intensités

Afin de séparer les caractères, nous introduisons une phase de prétraitement qui distingue le fond du texte. Nous faisons l'hypothèse que les intensités de ces deux classes (le « texte » et le « fond ») suivent des distributions gaussiennes. En utilisant l'algorithme EM (Espérance-Maximisation), nous estimons les paramètres de ces distributions qui servent à la génération d'une première carte floue d'appartenance à la classe « texte ». Par ailleurs, nous introduisons l'intégration multi-images pour identifier le fond par son éventuelle variabilité au cours du temps. Nous générons une autre carte floue indiquant les probabilités d'appartenir à la classe « fond ». Nous proposons ensuite de fusionner ces deux cartes en optant pour un opérateur à comportement adaptatif. La figure 2 illustre un exemple de carte produite, utilisée dans la suite pour la segmentation des caractères.

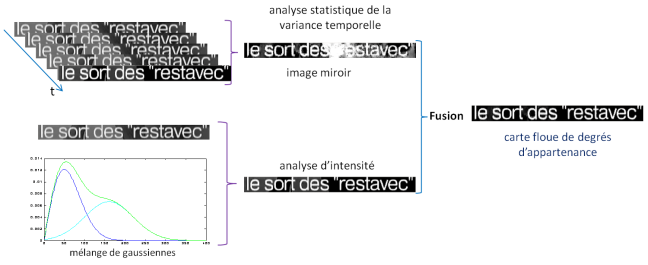

FIGURE 2 – Génération de carte floue d'appartenance.

2.2.2 Segmentation fondée sur l'algorithme du plus court chemin

Nous cherchons alors à déterminer les séparations non linéaires entre les caractères et qui s'adaptent à leur morphologie. En nous inspirant de [6], nous définissons la segmentation comme un problème de plus court chemin coupant verticalement la carte floue générée. Trois déplacements sont autorisés : 45 °, 90 ° et 135 ° par rapport à l'horizontale. À tout chemin est associé un coût égal à la valeur du pixel de plus forte probabilité d'appartenance à la classe « texte ». Deux types de segmentations sont distingués : celles dites « fiables » de coût inférieur à un certain seuil et celles dites « risquées » de coût plus élevé. Ces dernières pourront être remises en question par la suite (*cf.* section 2.4). Comme le montre la figure 3, les segmentations « risquées » peuvent correspondre à des sur-segmentations ou à des segmentations de caractères attachés à un fond complexe. Les chemins entre les mots sont identifiés comme un espace.

FIGURE 3 – Un exemple de segmentations obtenues : celles « fiables » représentées en vert et celles « risquées » en rouge.

2.3 Reconnaissance de caractères fondée sur une approche de classification neuronale

Nous présentons ici notre méthode permettant reconnaître les caractères segmentés. Contrairement à la majorité des mé-

1. Nous supposons que les textes incrustés dans les vidéos s'affichent pendant au moins 2 secondes, sinon ils ne seraient pas lisibles.

thodes de l'état de l'art, nous nous appuyons sur une approche de classification neuronale capable d'apprendre automatiquement et conjointement à extraire les primitives appropriées et à reconnaître les classes de caractères, sans aucune phase de binarisation.

Les réseaux de neurones à convolutions, ci-après nommés ConvNets, sont des réseaux de neurones particuliers introduits par LeCun *et al.* [5] pour reconnaître les formes visuelles à partir d'une image sans aucun prétraitement. Ils reposent sur les notions de champs réceptifs locaux, de poids partagés et d'opérations de sous-échantillonnage dans le domaine spatial. Grâce à ces principes, ils sont capables de traiter des formes extrêmement variables tout en étant robustes aux distorsions, à la variabilité d'échelle et aux transformations géométriques.

Dans le cadre de notre application, nous avons testé plusieurs configurations de réseau avant d'opter pour l'architecture de la figure 4. Le réseau prend en entrée une image de caractère en niveaux de gris. Les deux premières couches peuvent être interprétées comme des extracteurs de primitives alors que les deux suivantes permettent de combiner ces primitives. Les trois dernières couches de neurones sont chargées de la classification et produisent en sortie les degrés d'appartenance aux classes de caractères.

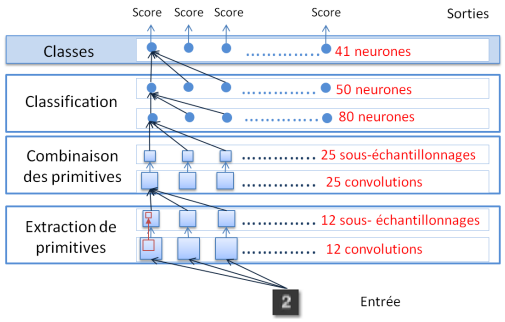

FIGURE 4 – Architecture du réseau de neurones à convolutions.

2.4 Intégration d'un modèle de langue pour améliorer les performances de la reconnaissance

Malgré les bonnes performances de la reconnaissance neuronale (*cf.* section 3), des erreurs peuvent être produites à cause d'une confusion entre caractères, de la complexité du fond et de la mauvaise qualité des vidéos. Pour pallier les ambiguïtés relatives à la reconnaissance locale caractère par caractère, nous proposons d'introduire des connaissances linguistiques qui vont piloter les étapes de l'OCR.

Les modèles de langue n-grammes ont montré leur capacité à améliorer les performances en reconnaissance de la parole en prenant en compte le contexte lexical. S'appuyant sur des analyses statistiques de corpora, ils permettent de prédire le prochain mot dans une phrase étant donné les mots qui viennent d'être employés. Dans notre application, un modèle n-grammes est appris afin d'estimer la probabilité qu'une séquence de lettres soit observée. En faisant l'hypothèse qu'un caractère ne dépend que de ses $n-1$ prédécesseurs et à l'aide de la librairie *SRILM* [10], le modèle est entraîné à apprendre les probabilités jointes des séquences de caractères sur un corpus de mots français. Ces probabilités sont ensuite intégrées dans notre chaîne pour obtenir les propositions les plus fiables de mots. Comme l'illustre la figure 5, pour chaque mot identifié, un graphe est construit. Chaque hypothèse de segmentation est représentée par un nœud où les séquences optimales de caractères sont générées et reçoivent un score mêlant résultats de la reconnaissance, probabilités du modèle de langue et hypothèses de segmentation. Les meilleurs mots candidats, retrouvés grâce à l'algorithme de Viterbi, sont enfin vérifiés à l'aide d'un dictionnaire.

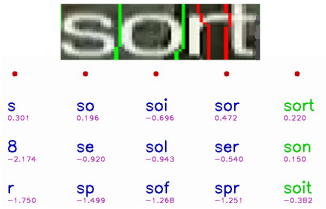

FIGURE 5 – Un exemple de graphe de reconnaissance obtenu.

3 Résultats expérimentaux

Nos expérimentations ont été réalisées sur un ensemble de 12 vidéos de journaux télévisés français. Chacun dure en moyenne 30 minutes et contient environ 400 mots incrustés, soit 2 200 caractères assez variables en tailles, couleurs, styles et fonds. 8 vidéos sont annotées pour servir au test de la chaîne complète de l'OCR et 4 sont employées pour générer une base de 15 168 images de caractères individuels parfaitement séparés, utilisée pour entraîner le réseau de neurones. 41 classes sont considérées : les 26 lettres, les 10 chiffres, 4 caractères spéciaux ('.', '-', '(', et ')') et l'espace entre mots.

Nous évaluons tout d'abord notre approche neuronale par ConvNets en reconnaissance de caractères et comparons ses performances à une classification par SVM (Séparateurs à Vaste Marge). Dorai *et al.* [4] ayant testé les modèles SVM sur des caractères incrustés dans les vidéos sur une base différente de la nôtre, nous avons eu recours à la librairie *LIBSVM* [1] pour implémenter leur méthode et la tester, dans plusieurs configurations (*cf.* tableau 1), sur notre base. De nombreuses architectures de ConvNets ont été également évaluées (*cf.* tableau 2). Parmi celles-ci, ConvNet_2 obtient le meilleur taux de reconnaissance : 98.04%. Les taux des autres architectures sont également bons mais inférieurs à 90% à cause d'un problème de généralisation pour ConvNet_1 et de sur-apprentissage pour ConvNet_3. Malgré ces limites, les taux de reconnaissance des ConvNets restent bien supérieurs à ceux des SVM où le meilleur résultat est de 81.18%.

Nous testons ensuite, sur les 8 vidéos annotées, l'influence de l'intégration du modèle de langue sur les performances du système complet. La figure 6 illustre certaines corrections apportées, notamment la levée de la confusion de caractères et

TABLE 1 – Taux de reconnaissance des configurations de SVM (C est le paramètre de pénalité, VS le vecteur support et TR le taux de reconnaissance).

SVM Id	C	Nombre de VS	TR de caractères
SVM_1	1	9215	75.46%
SVM_2	2	8544	81.18%
SVM_3	3	8091	80.65%

TABLE 2 – Taux de reconnaissance des architectures de ConvNets : C1 et C2 (resp. N1 and N2) sont les nombres de cartes de primitives (resp. neurones) des couches 2 et 3 (resp. 5 et 6).

ConvNets Id	C1	C2	N1	N2	TR de caractères
ConvNets_1	10	15	60	40	87, 17%
ConvNets_2	12	25	80	50	98.04%
ConvNets_3	15	20	120	80	89, 96%

l'élimination de sur-segmentations.

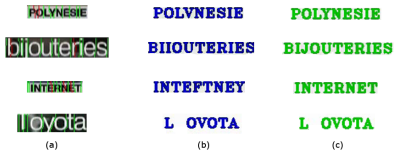

FIGURE 6 – Exemples de textes extraits et reconnus : (a) les images extraites, (b) et (c) les textes reconnus avant et après l'intégration d'un modèle de langue (trigrammes).

Le tableau 3 décrit l'impact de la taille de l'historique du modèle n-grammes. Le faible contexte du modèle bigrammes, s'il permet une hausse des performances, semble insuffisant. Le meilleur taux de reconnaissance de mots est obtenu par le trigrammes, le quadrigrammes n'améliorant pas les résultats tout en étant plus complexe. La diminution du taux de reconnaissance de caractères entre les tableaux 2 et 3 (de 98.04% à 92.69%) peut se justifier par des erreurs de segmentation, forcément absentes dans l'expérimentation sur la base de caractères, que le modèle de langue n'a pu rectifier. La prise en compte additionnelle du dictionnaire permet d'accroître encore le taux de reconnaissance de caractères (94.95%) et de mots (78.24%) de notre système.

4 Conclusion

Dans cet article, nous avons présenté une chaîne complète d'OCR spécialement conçue pour détecter et reconnaître les textes incrustés dans des vidéos. Se basant sur une approche neuronale, la méthode de reconnaissance de caractères proposée a permis d'obtenir de très bons résultats (98%) et a considérablement dépassé les performances de méthodes reposant sur des modèles SVM (81%). Nous avons aussi démontré que notre système tire profit de l'ajout de connaissances linguistiques (modèle de langue et dictionnaire) prenant en compte le contexte lexical. L'OCR vidéo proposé, évalué sur une base de vidéos de journaux télévisés, atteint ainsi un taux de reconnaissance de caractères très élevé d'environ 95% correspondant à un taux de reconnaissance de mots de 78%. Ces résultats prometteurs permettent d'envisager l'intégration de notre OCR dans un système d'indexation de vidéos. Plus particulièrement, les textes reconnus serviront à extraire des informations de haut niveau sémantique participant à l'indexation des vidéos conjointement avec des informations issues d'autres modalités.

TABLE 3 – Évaluation de l'influence du paramètre n.

n-grammes	TR de caractères	TR de mots
Système de référence	88.14%	63.04%
Bigrammes	89.37%	65.45%
Trigrammes	92.69%	74.34%
Quadrigrammes	90.13%	68.78%

Références

[1] C. Chang and C. Lin. LIBSVM : a library for support vector machines. 2001.

[2] T. Chen, D. Ghosh, and S. Ranganath. Video-text extraction and recognition. In *TENCON'04*, volume 1, pages 319–322, 2005.

[3] M. Delakis and C. Garcia. Text detection with convolutional neural networks. In *VISAPP*, volume 2, pages 290–294, 2008.

[4] C. Dorai, H. Aradhye, and J.-C. Shim. End-to-end video text recognition for multimedia content analysis. In *ICME*, pages 601–604, 2001.

[5] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, pages 255–258, 1995.

[6] S. Lee, D. Lee, and H. Park. A new methodology for gray-scale character segmentation and recognition. *PAMI*, 18(10) :1045–1050, 2002.

[7] R. Lienhart and F. Stuber. Automatic text recognition in digital videos. *Image and Video Processing*, pages 2666–2675, 1996.

[8] Z. Saidane and C. Garcia. Automatic scene text recognition using a convolutional neural network. In *CBDAR*, pages 100–106, 2007.

[9] T. Som, D. Can, and M. Saraclar. HMM-based sliding video text recognition for Turkish broadcast news. In *ISCIS*, pages 475–479, 2009.

[10] A. Stolcke. SRILM-an extensible language modeling toolkit. In *ICSLP*, volume 3, pages 901–904, 2002.

[11] J. Yi, Y. Peng, and J. Xiao. Using multiple frame integration for the text recognition of video. In *ICDAR*, pages 71–75, 2009.