TEXT DETECTION WITH CONVOLUTIONAL NEURAL NETWORKS

Manolis Delakis and Christophe Garcia
Orange Labs, 4, rue du Clos Courtel, 35512 Rennes, France
Manolis.Delakis@orange-ftgroup.com, Christophe.Garcia@orange-ftgroup.com

Keywords: Image understading, Neural networks, Text detection, Pattern recognition.

Abstract: Text detection is an important preliminary step before text can be recognized in unconstrained image environments. We present an approach based on convolutional neural networks to detect and localize horizontal text lines from raw color pixels. The network learns to extract and combine its own set of features through learning instead of using hand-crafted ones. Learning was also used in order to precisely localize the text lines by simply training the network to reject badly-cut text and without any use of tedious knowledge-based post-processing. Although the network was trained with synthetic examples, experimental results demonstrated that it can outperform other methods on the real-world test set of ICDAR'03.

1 INTRODUCTION

Any text appearing in an image can provide useful information for the task of automatic image annotation and other related problems. In order to recognize this text, we first need to detect the real text area inside the image and separate it from the background. In simplified scenarios of uniform background, text detection is straightforward and can be accomplished with simple image thresholding or color clustering. In cases, however, of cluttered background and free environments, detecting text is a challenging task as the image background and the text itself as well are unpredictable.

A survey on text detection can be found in (Jung et al., 2004). One can notice that the main effort is focused in finding out a sophisticated set of edge or texture-based features for a robust discrimination between text and non-text patterns. For instance, edgel features (Garcia and Apostolidis, 2000) that measure the density of oriented lines in an image portion have been proposed. The final classification on top of these features is performed by adaptive thresholds and by imposing knowledge-based geometric constraints. With the rise of robust and efficient machine learning techniques the tedious task of finding appropriate thresholds for an increasing number of features has been replaced by learning-based classifiers. Multilayer perceptrons (MLPs), for instance, have been used on top of wavelet features (Li et al., 2000) or Support Vector Machines (SVMs) on top of a carefully selected set of features (Lucas et al., 2005; Chen et al., 2004).

What is common in the above approaches, is that text detection is considered as a two-phase process where, firstly, the engineer has to manually select an appropriate set of features and, in the second step, learning takes place. The introduction of learning also in the first phase is a challenging task as image pixels result into high-dimensional input spaces, while the text pattern does not have any specific spatial distribution. In a simplified scenario, (Jung, 2001) used 3-layered MLPs that take input from 13×10 image areas and a separate arbitration stage for the final classification.

In this study, we propose the use of convolutional neural networks, introduced by (LeCun et al., 1998), for text detection directly from raw color pixels. In this advanced MLP architecture, feature extraction and classification are performed jointly in one step. The relative features and their combinations are learned by examples via the back-propagation algorithm. In addition to the automatic feature extraction, we also put emphasis on the good localization of the

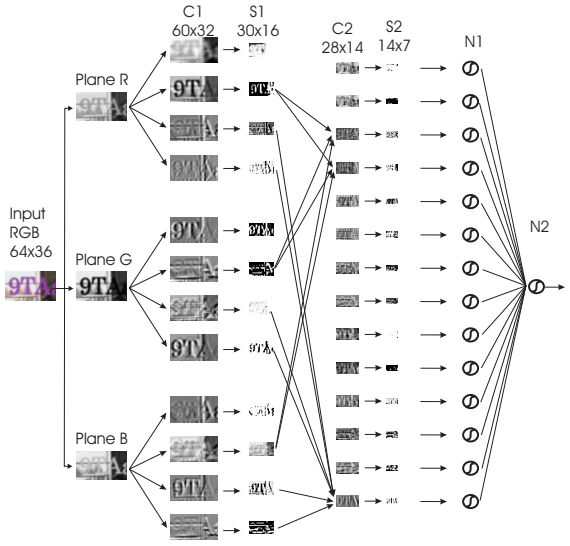

Figure 1: Illustration of the proposed network topology. Not all the connections between the layers $S1$ and $C2$ are depicted. The contents of the feature maps provide a visualization of the real features extracted from the input image, after training has been accomplished.

text lines by the network itself, instead of applying a set of tedious geometric constraints and local image processing. This was achieved by training the network to reject badly localized text. Even though we used computer-generated examples in the training corpus, we present results in real-world settings and compare with other methods.

The remainder of the paper is organized as follows. The convolutional topology is presented in section 2 and training details in section 3. The way the network is used to scan for text an image is presented in section 4. Section 5 reports experimental results and section 6 concludes this study.

2 NETWORK TOPOLOGY

An illustration of the topology of the convolutional network is given in Fig. 1. The network receives in its input plane an RGB image of 64×36 pixels, to be classified as containing text or not. Before feeding the network, the image is decomposed to its three color channels, forming the actual network input, which is 3 64×36 intensity image planes. Their pixel values are linearly scaled between -1 and +1.

The network employs a series of filters through the layers $C1$ to $S2$ which extract appropriate features and finally the classification is performed in the last two layers ($N1$ and $N2$). The filters are adjustable and are learned as legitimate networks weights instead of being hand-crafted as in other text detection approaches. Although the image channels are separated in the input layer, features from different channels are fused long before the final classification stage, between the layers $S1$ and $C2$. In the following, we give a brief description of the operations performed in each layer of the convolutional network. For more details, the interested reader is referred to (LeCun et al., 1998) and (Garcia and Delakis, 2004).

Layer $C1$ has 12 planes (called *feature maps*) that are grouped according to the image channel they process, as shown in Fig. 1. Each feature map performs a convolution over its input plane with a 5×5 trainable mask. The convolutional layer $C1$ is then followed by the subsampling layer $S1$ in order to enhance the robustness of the network to input deformations. More precisely, a local averaging over 2×2 neighborhoods of $C1$ is performed, followed by a multiplication with a trainable coefficient and the addition of a trainable bias. The final outcome is passed through a sigmoid.

Layer $C2$ contains 14 convolutional features maps, like the ones of $C1$, connected to some of the feature maps of the $S1$ layer. Mixing the outputs of feature maps helps in extracting more complex information by combining different features. In addition, robust features that are extracted in different color bands can now be fused for subsequent processing. In the proposed topology, each triplet (R_i, G_i, B_i) with $i = 1\ldots4$ of feature maps in $S1$ is connected to two feature maps of $C2$. The remaining 6 feature maps of $C2$ implement all the possible combinations of two triplets of $S1$.

The operation of the $S2$ and $C2$ layers is similar to that of $S1$ and $C1$, respectively, with the only difference that there are 3×3 convolution masks in $C2$. Finally, layers $N1$ and $N2$ contain standard sigmoid neurons. Note that each neuron in layer $N1$ is connected to only one $S2$ feature map. Overall, the network of Fig. 1 corresponds to a large topology with some hundred thousands connections but only with 2,319 adjustable weights. This special design results to built-in robustness to input deformations and to increased network generalization capabilities.

3 NETWORK TRAINING

In order to gather example text images, we used the ImageMagick[1] tool to artificially create a large corpus of 100,800 examples with varying background, text font and text color. Some of them are depicted in Fig. 2.a. The background can be a randomly selected color or a random image portion. In addition to

[1] An image processing software that allows printing text on images, http://www.imagemagick.org

Figure 2: Image examples (a) with text, (b) without or badly-cut text, and (c) gathered during bootstrapping.

the positive examples of text images, we created a set of negative examples with random parts of scenery images. We added to this a number of images containing multiline or badly-cut text in order to boost the network in precisely localizing the text when it is used to scan an image (section 4). A total number of 64,760 negative examples was thereby produced. Some of them are depicted in Fig. 2.b. Finally, in order to check the generalization of the network during training, a validation set containing 10,640 positive and an equal number of negative examples was created with the same method as for the training set.

The network was trained with the standard back-propagation algorithm. Target responses were fixed to -1 for negative and +1 for positive examples. The rejection ability of the network was boosted with a bootstrapping procedure: after some training cycles, false alarms are gathered by running the network on a set of scenery images. These false alarms are added to the set of negative examples and then training goes on until convergence is noticed. Some of the false alarms gathered are shown in Fig. 2.c. Due to bootstrapping, the set of negative examples was augmented to a total number of 114,407 examples. The network was finally able to correctly classify 96.45% of the positive examples and 97.01% of the negative ones on the (augmented) training set. Regarding the validation set, the figures are similar (95.84% and 97.45%, respectively).

4 IMAGE SCANNING

We describe in this section how the trained convolutional network is used to scan an entire input image in order to detect horizontal text lines of any height that may appear at any possible image location.

In order to detect text at varying height, the input image is repeatedly subsampled by a factor of 1.2 to construct a pyramid of different scales. The network is applied to any slice (scale) of this pyramid individually. As the neural network uses convolutional kernels at its first layers, instead of feeding the network at each possible image location, we can apply the network filters to the entire pyramid slice at once, thus saving a lot of computation time. This filtering procedure will provide the network responses as if it was applied at each image position with a step of 4 pixels in both directions, as two subsampling operations take place (in layers $S1$ and $S2$).

In the next step of the scanning procedure, the responses collected at each scale are grouped according to their proximity in scale and space to form a list of candidate targets. The horizontal extension of a group is determined by the left and right extremes of the group, while the scale is averaged. Cases of multiline text are easily discarded in favour of the actual text lines that constitute it because the network is trained to reject multiline text.

Finally, the rectangles of the candidates are inspected individually by forming local image pyramides around them and applying the network at each slice with step 1 in both directions in order to measure more effectively the density of the positive activations. The candidates that score low average activation are considered as false alarms and are rejected.

5 EXPERIMENTAL RESULTS

We tested our approach on the ICDAR'03 robust reading competition corpus (Lucas et al., 2005). It comes with a training set of 258 images and a test set of 251 images, all of them manually annotated. The corpus contains exclusively scene text, i.e., text that naturally appears in an image. Although our method uses synthetic examples that simulate superimposed text, we provide experimental results in the ICDAR'03 test set in order to test the generalization capabilities of the network and to compare with other methods.

Based on the performance on the training set of the corpus, we fixed the threshold on the average activation for false alarm rejection to 0.12 and confined our search for text in the range of 36 to 480 pixels high. Our method detects entire text lines while the ground truth of the corpus annotates every word separately (even if it consists of a single letter). This would cause reporting unfair detection rates. Thus, we followed the performance metric of Wolf (Lucas et al., 2005) to compare the rectangles of the detected text of our method to the rectangles of the ground truth. This metric takes into account one-to-many correspondences between the detected rectangles and the annotated ones.

Results are reported in table 1, in terms of precision and recall rates. The third column gives the standard f-measure, which combines precision and recall rates in one measurement (see (Lucas et al., 2005)

Table 1: Experimental results in the ICDAR'03 test set and comparison with other methods.

System	Precision	Recall	f	Detected
HWDavid	0.43	0.52	0.47	1916
Ashida	0.53	0.41	0.46	1515
Wolf	0.21	0.52	0.30	3477
Todoran	0.14	0.18	0.16	1368
ConvNN	**0.54**	**0.61**	**0.57**	**751**

for an exact definition). The fourth column of the table reports the total number of detected rectangles for each method. Ashida uses fast color-based target detection, supported by SVMs at a verification stage. Wolf uses SVMs on top of a set of edge and geometric features. The systems of HWDavid and Todoran are based on image edge filtering, morphological operations and other features, followed by geometric constraints. The last row of the table reports results of the proposed method. For all the methods, the reported results refer to the Wolf performance metric.

We see in table 1 that the proposed approach outperforms the other systems with 54% and 61% precision and recall rates, correspondingly, and with an f score of 57%. What is remarkably different is the number of detected rectangles (751) that are produced by the convolutional network. This is partly due to the fact that our scanning strategy detects complete text lines (and so fewer rectangles), partly due to the nice rejection capabilities of the network that result in fewer false alarms.

Some results of the proposed approach are shown in Fig. 3. We notice that the network is able to detect scene text that is slightly rotated and to separate multiline text. Although the false alarm rate is low, some patterns like the piano keys we see in the figure are difficult to reject as they resemble to a series of text characters. The network may miss some text which does not contain a lot of characters and thus cannot occupy a significant place inside its retina.

Regarding computational cost, the proposed system can process a 768×576 image in 1.25 seconds using a Xeon processor at 3.0 Ghz and searching in the range from 36 to 480 of font sizes.

Figure 3: Some result of the proposed approach on the ICDAR'03. test set.

6 CONCLUSIONS

We have presented in this study a convolutional neural network-based text detection system that learns to automatically extract its own feature set instead of using a hand-crafted one. Furthermore, the network learned not only to detect text in its retina, but also to reject multiline or badly localized text. Thus, the exact text localization does not require any tedious knowledge-based post-processing. Even though the network was trained with synthetic examples, experimental results demonstrated that it can compete with other methods in a real-world test set. Future work includes the inspection of the text binarization and recognition problems with convolutional neural networks.

REFERENCES

Chen, D., Odobez, J.-M., and Bourlard, H. (2004). Text detection and recognition in images and video frames. *Pattern Recognition*, 37(5):595–608.

Garcia, C. and Apostolidis, X. (2000). Text detection and segmentation in complex color images. In *Proceedings of ICASSP'00*, pages 2326–2329, Washington, DC, USA. IEEE Computer Society.

Garcia, C. and Delakis, M. (2004). Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1408–1423.

Jung, K. (2001). Neural network-based text location in color images. *Pattern Recognition Letters*, 22(14):1503–1515.

Jung, K., Kim, K. I., and Jain, A. (2004). Text information extraction in images and video: a survey. *Pattern Recognition*, 37(5):977–997.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, H., Doermann, D., and Kia, O. (2000). Automatic text detection and tracking in digital videos. *IEEE Transactions on Image Processing*, 9(1):147–156.

Lucas, S., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R., Ashida, K., Nagai, H., Okamoto, M., Yamamoto, H., Miyao, H., Z., J., Ou, W.-W., Wolf, C., Jolion, J.-M., Todoran, L., Worring, M., and Lin, X. (2005). ICDAR 2003 robust reading competitions: entries, results, and future directions. *International Journal on Document Analysis and Recognition*, 7(2-3):105–122.