Robust Binarization for Video Text Recognition

Zohra Saidane and Christophe Garcia
Orange Labs
4, rue du Clos Courtel BP 91226
35512 Cesson Sévigné Cedex - France
firstname.lastname@orange-ftgroup.com

Abstract

This paper presents an automatic binarization method for color text areas in images or videos, which is robust to complex background, low resolution or video coding artefacts. Based on a specific architecture of convolutional neural networks, the proposed system automatically learns how to perform binarization, from a training set of synthesized text images and their corresponding desired binary images, without making any assumptions or using tunable parameters. The proposed method is compared to state-of-the-art binarization techniques, with respect to Gaussian noise and contrast variations, demonstrating the robustness and the efficiency of our method. Text recognition experiments on a database of images extracted from video frames and web pages, with two classical OCRs applied on the obtained binary images show a strong enhancement of the recognition rate by more than 40%.

1. Introduction

Recognizing artificial text embedded in images provides high level semantic clues which enhance tremendously automatic image and video indexing. While for printed document, optical character recognition (OCR) systems have already reached high recognition rates, and are widely commercialized, recognition of superimposed text in images and videos is still the subject of active research.

Current character recognition systems require a binarization step that aims at separating the text pixels from the background pixels of the processed image. Most of the text image binarization methods rely on global or local discriminating thresholds. These thresholds are determined according to some statistical analysis of the luminance or chrominance distribution generally based on histograms, without taking into account the characters shapes.

A well known global thresholding technique is the Otsu’s method [9]. It assumes that the text and the background intensities are homogeneous. The threshold is then selected by partitioning the image pixels into two classes in order to maximize the between-class intensity variance which maximizes the separability of the resultant classes. However, due to transparency effects on complex background, these two classes are often not separable and the Otsu’s method cannot find a suitable threshold value.

Another global thresholding technique, proposed by Lienhart et al. [7], consists in choosing the intensity separating value halfway between the intensity of the text color and the background color as a threshold. Estimating the intensity of the text and the background colors is based on the histogram distribution of the two upper and lower rows of the text box (as they should contain mainly background pixels), and on the histogram distribution of the four central rows of the text box (as they should contain mainly text pixels).

Most local methods are also based on histogram distributions and statistical analysis. These techniques mainly choose a threshold for different image blocks instead of a threshold for the whole image. Niblack [8] computes a threshold based on the mean and the standard deviation of each block. One drawback of this approach is the noise created in areas which do not contain any text, due to the fact that a threshold is estimated and applied in these cases as well. Sauvola et al. [10] have been inspired from the work of Niblack. In addition, they made a hypothesis on the gray values of text and background pixels: text pixels are assumed to have gray values near 0 and background pixels are assumed to have gray values near 255. This method gives better results for document images, but creates additional problems for video frames whose contents do not always correspond to the hypothesis, even if images containing bright text on dark background are reversed to be on the desired configuration.

Obviously, local methods are more robust than the global ones when the intensity of the text or that of the background is not homogeneous. However, their performance depends heavily on the size of the image blocks and on several tun-
able parameters [8, 10].

Although a lot of binarization techniques are concerned only with gray level images, an increasing interest is given to the binarization of color text images.

To use the chrominance information, some methods apply the thresholding techniques, previously cited, on one or more color channel independently and then fuse the different results [6]. Some others quantify the colors into a reduced number of dominant colors based on clustering algorithms, such as K-means. In [3], this number is fixed to four. Then, the color or the two colors with the biggest rate of occurrences in the image border areas are considered as the background colors. The remaining colors are to be assigned to text or background. All the combinations are considered, each of which results in a possible binary image. The best one among them is determined by analyzing the periodicity of the vertical profiles of the binary image. This kind of methods depends on a lot of parameters, usually difficult to determine (the number of quantification levels, the size of the image border, thresholds of the periodicity, etc.).

In this paper, we propose a novel automatic binarization scheme for color text images, based on supervised learning, without making any assumptions or using tunable parameters. Our contributions to the current state of the art are the following.

- Developing a new convolutional neural network architecture that processes color text images to directly produce binary text images.
- Using conjointly color distribution and the geometrical properties of characters.
- Insuring robustness to noise, to complex backgrounds and to luminance variations.

The remainder of this paper is organized as follows. Section 2 describes in detail the architecture of the proposed neural network, especially the two new layers that we added to the classical convolutional neural network to deal with the specific problem of binarization, namely, the up-sampling and the inverse convolution layers. It explains also the training process. Experimental results are reported in Section 3. Conclusions are drawn in Section 4.

2. CTB: Convolutional Text Binarizer

2.1. Architecture of the CTB

The proposed neural architecture, called Convolutional Text Binarizer (CTB), is based on convolutional neural network architecture [5, 4]. As shown in Fig.1, it consists of five different heterogeneous layers. Each layer contains feature maps which are the results of either convolution, sub-sampling, up-sampling or inverse convolution operations. Applying and combining these automatically learnt operations insure the extraction of robust features, leading to the automatic construction of the binary image.

The first layer is the input layer E; it consists of $N_E = 3$ input maps, each of them corresponding to one color channel of the image, according to the color space (RGB, YUV, etc.). Their pixel values are normalized to the range [-1, 1]. The RGB color space has been chosen in our experiments.

The second layer is a convolution layer $C_1$. It is composed of $N_{C_1}$ maps. Each unit in each map is connected to a $M_1 \times M_1$ neighborhood (biological local receptive field) in the maps of previous layers. Furthermore, the trainable weights (convolutional mask) forming the receptive field, are forced to be equal for the entire map (weight sharing). A trainable bias is added to the results of each convolutional mask. Each map can be considered as a feature map that has a learnt fixed feature extractor that corresponds to a pure convolution with a trainable mask, applied over the maps in the previous layer. Multiple maps lead to the extraction of multiple features.

The third layer is a sub-sampling layer $S_2$. It is composed of $N_{S_2}$ feature maps. It performs local averaging and sub-sampling operations. We use this sub-sampling layer to reduce by two the spatial resolution which reduces the sensitivity to shifts, distortions and variations in scale and rotation. It is fully connected to the previous layer $C_1$ which results in combining the local features extracted in the layer $C_1$ and extracting more complex information.

At this stage, the required features are extracted from the input image and the following steps will perform mainly the...
construction of the binary image.

The fourth layer is an up-sampling layer $U_3$. It contains $NU_3$ maps $U_3k$. Each set of non-overlapping $M_2 \times M_2$ unit of each map is connected to a unit in each map of the previous layer (cf. Fig.2). The $M_2 \times M_2$ synaptic weights and the bias are trainable and are shared over each map.

The fifth and final layer is the inverse convolution layer that contains the output map $F$. It is fully connected to the previous layer based on an inverse scheme of the convolution. In fact, in the convolution layer, each neuron of the convolution map is connected to a set of $M_1 \times M_1$ neighbor element in the input map whereas each set of $M_1 \times M_1$ neuron of the inverse convolution map $F$ is connected to one unit of the previous layer (cf. Fig.3) in order to obtain an output map of the size of the input image. Sigmoid activation function is used and the neuron outputs are continuous and vary between -1 and 1.

2.2. Training of the CTB

Training such a network requires a training set containing pairs of images: the color text images and their corresponding binary images. To avoid hard and tremendous annotation task and manual text binarization, and because we are convinced of the good generalization ability of the neural network, we choose to use synthetic images that we build automatically. Thus, a set of binary text images of size $W \times H = 48 \times 24$ is constructed using different fonts. For each one, we randomly choose a text color and a background color and we apply different types of noise (uniform and Gaussian noises), and then we apply smoothing filters in order to obtain images looking like real ones and presenting a lot of variability. Fig.4 shows some examples of training images.

Our training set contains 4500 RGB color images. We choose to build $NC_1 = NS_2 = NU_3 = 30$ maps in layers $C_1$, $S_2$ and $U_3$. We use linear activation functions in $C_1$ and the sigmoid in $S_2$, $U_3$, and $F$. The convolution and the inverse convolution window size $M_1 \times M_1$ is $5 \times 5$. The sub-sampling and the up-sampling factor is two in each direction. The different parameters governing the proposed architecture, i.e., the number of layers, the number of maps, as well as the size of the receptive fields, have been experimentally chosen.

The training phase was performed using the classical back-propagation algorithm with momentum modified for being used in convolutional networks as described in [5]. At each iteration, the three RGB channels of a color text image are presented to the network as inputs and its corresponding binary image as the desired output map. The weights are updated as function of the error between the obtained output image and the desired binary image.

The objective of the network being to obtain final values $F_h$ equal to the desired values $D_h$, we classically choose to minimize the MSE (Mean Square Error) between obtained and desired values over a validation set of $N_v = 500$ images.

$$MSE = \frac{1}{N_v \times W \times H} \sum_{k=1}^{N_v} \sum_{h=1}^{W \times H} (F_h - D_h)^2$$

After training, the system is able to produce an output map for a given color input text image. The last step consists of converting this output map $F$ on a binary image according to the sign of the neurons outputs. Let $f_{i,j}$ be the value of the element $(i,j)$ in the output map $F$. The pixel $P_{i,j}$ at the position $(i,j)$ in the binary image is assigned to 0 if $f_{i,j} < 0$ and to 1 if $f_{i,j} \geq 0$. 

![Figure 2. The up-sampling layer](image)

![Figure 3. The inverse convolution layer](image)
3. Experimental results

To test the performance of our method, we use two databases. The first one is composed of synthetic images, which allow us to evaluate the robustness of our method according to noise and contrast variations. The second one contains real images collected from video frames and web pages.

When testing real images, we first crop manually the text box. Then, we resize it so that its height is equal to the retina height \( H = 24 \), while keeping the same aspect ratio. Afterwards, we move a sliding window of the CTB retina size along the text box. We present each window image to the CTB in order to obtain the corresponding window binary image. Finally, concatenating these binary window images provides us with the binary image of the text box.

We compare our method to the following well-known binarization methods: the Otsu method [9], the Niblack method [8], the Sauvola method [10], and the Lienhart method [7]. The evaluation is based on the MSE between obtained binary images and desired binary images for the tests on contrast and noise, and on the recognition rates of a free OCR, named TESSERACT [1], and a trial version of a commercial one, named ABBy FineReader 8.0 [2] for test on real text images.

To test the impact of noise on the different binarization methods, we build a database of 200 text images add an increasing Gaussian noise, using the variances 3, 5, 7, 9 and 11.

Fig.5 shows the stability of our method with respect to noise addition and that it gets the lowest MSE. We notice that Otsu method, which is a global thresholding technique, is the most sensitive to noise addition with a variation of 0.46 units in MSE.

![Figure 5. MSE versus additive Gaussian noise](image)

To test the influence of contrast variations, a database of 1000 synthetic text images is used. Then the images are classified according to their contrast. The contrast is computed based on the following formula:

\[
\text{Contrast} = \frac{|\text{TxtColor} - \text{BckColor}|}{\text{TxtColor} + \text{BckColor}}
\]

where \( \text{TxtColor} \) and \( \text{BckColor} \) are the color of the text and the background, respectively.

Fig.6 shows that our method is extremely robust to contrast variation, followed by the Lienhart and the Niblack methods. Notice that generally, methods are more influenced by contrast variation than by noise variation. In fact, if we compute the relative variation rate for each curve, we find that the average relative variation rate for the noise variation curve is 3.19%, while for the contrast variation curve, it is 7.07%.

![Figure 6. Recognition rates versus contrast variations](image)

Fig.7 reports the recognition rates of the two OCRs (Tesseract and Abby FineReader 8.0) applied on real text images, binarized by these different methods. The database is composed of 161 images collected from video frames and web pages, containing 2252 characters. The images contain complex background, with high chrominance variability. Some examples of this database are shown in Fig.8 and others with binarized images obtained by the CTB and by the Lienhart method (as it gets the second best results) are shown in Fig.9.

Fig.7 illustrates the impact of the robust binarization provided by our method both on free and commercial OCR. We reach a recognition rate of 86.234% with the commercial OCR, while the other methods get recognition rates lower than 50%. It can be noticed that a slight gain in MSE have a tremendous impact on text recognition results. In fact, some erroneous pixels around a character, do not have high
impact on the MSE, but can lead the recognizer to misclassify the character.

**Figure 7. Recognition rates on binary images**

**Figure 8. Some examples from the collected database for text recognition**

**4. Conclusion**

In this paper, we have proposed a novel approach based on a specific convolution neural network architecture designed for complex color text image binarization. Based on supervised learning, our system does not need any tunable parameter and takes into account both color distributions and the geometrical properties of characters. Experimental results on noise and contrast variations show that the proposed method is robust and greatly enhance text recognition rates using classical document OCRs. As an extension to this work, we believe that processing not only superimposed text but also scene text with the proposed system is possible thanks to its generalization capabilities.

**References**