Automatic Scene Text Recognition using a Convolutional Neural Network

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 recognition method for color text characters extracted from scene images, which is robust to strong distortions, complex background, low resolution and non uniform lightning. Based on a specific architecture of convolutional neural networks, the proposed system automatically learns how to recognize characters without making any assumptions, without applying any pre-processing or post-processing and without using tunable parameters. For this purpose, we use a training set of scene text images extracted from the ICDAR 2003 public training database. The proposed method is compared to recent character recognition techniques for scene images based on the ICDAR 2003 public samples dataset in order to contribute to the state-of-the-art method comparison efforts initiated in ICDAR 2003. Experimental results show an encouraging average recognition rate of 84.53%, ranging from 93.47% for clear images to 67.86% for seriously distorted images.

1. Introduction

Natural text scene images contain important semantic information such as names of streets, institutes, shops, road signs, traffic information, etc.

While for printed document, optical character recognition (OCR) systems have already reached high recognition rates, and are widely commercialized, recognition of characters in natural scene images is still the subject of active research. In fact, this task is a challenging problem because of low resolution, complex background, non uniform lightning or blurring effects.

Most of the state of the art text image recognition methods are based on template matching which is also the case of most available commercial OCR systems mainly designed for printed text. This is an old principle that was proposed for OCR in 1929 by Tausheck. It reflects the technology at that time, which used optical and mechanical template matching. Light passed through mechanical masks is captured by a photo-detector and is scanned mechanically. When an exact match occurs, light fails to reach the detector and so the machine recognizes the characters printed on the paper. Nowadays, the idea of template matching is still used but with more sophisticated techniques. A database of models is created and matching is performed based on a distance measure. The models are generally composed of specific features that depend on the properties of the pattern to be recognized.

Chen at al [1] proposed a method based on character side profiles, in videos. First, a database is constructed with left, right, top and bottom side-profiles of sample characters. Then the candidate characters are recognized by matching their side-profiles against the database. This method requires of course a 'cleaned' binary text image. Therefore, they apply various pre-processing techniques before the recognition step, namely: shot boundary detection, edge-based text segmentation, multiple frame integration, gray-scale filtering, entropy-based thresholding, and noise removal using line adjacency graphs (LAGs). The authors reported a character recognition rate varying from 74.6% to 86.5% according to the video type (sport video, news, commercial videos).

Another template matching method was proposed by kopf et al. [3]. They have chosen to analyze the contour of a character and derive features extracted from the curvature scale space (CSS). This technique which is based on the idea of curve evolution, requires also binary text images. A CSS image is defined by the set of points where the curvature is null. In many cases, the peaks in the CSS image provide a robust and compact representation of a contour with concave segments. For characters without concave segments (e.g. 'I' and 'O'), the authors proposed the extended CSS method, where the original contour is mapped to a new contour with an inverted curvature, thus, convex segments become concave and the problem is solved.

The matching process is done by comparing the feature vectors (CSS peaks) of an unknown character to those of the characters that are stored in a database. It might be necessary to slightly rotate one of the CSS images to best align the peaks. In fact, shifting the CSS image left or right corresponds to a rotation of the original object in the image. Each character is stored only once in the database, and for instance, the horizontal moves compensate small rotations of italic character.

If a matching peak is found, the Euclidean distance of the height and position of each peak is calculated and added to the difference between the CSS images. Otherwise, the height of the peak in the first image is multiplied by a penalty factor and is added to the total difference. If a matching is not possible, the two objects are significantly different. This rejection helps to improve the overall results because noise or incorrectly segmented characters are rejected in the matching step. A recognition rate of 75.6% is reported for a test set of 2986 characters extracted from 20 artificial text images with complex background.

Yokobayashi et al [8, 9] proposed two systems for character recognition in natural scene images. Both of them rely on two steps: the first one is the binarization step and the second one is the recognition step based on an improved version of GAT correlation technique for grayscale character images.

In [8], the authors proposed a local binarization method applied to one of the Cyan/Magenta/Yellow color planes using the maximum breadth of histogram. This idea is based on the hypothesis that the more information entropy of grayscale occurrence a given image has the more effectively and easily a threshold value of binarization for the image can be determined, given that the breadth of grayscale histogram is closely related to the information entropy contained in the color plane. Therefore, they apply local binarization to the selected color plane. They compute the mean value of gray levels of all pixels in the selected color plane. Then, if a majority of nine pixels in a 3x3 local window have smaller gray levels than this mean, the pixel is considered as a character pixel, otherwise it is considered as a background pixel.

Once a binary image is obtained, an improved GAT correlation method [7] is applied for recognition. This is a matching measure between the binary character image and a template image. As templates, the authors use a single-font set of binary images of alphabets and numerals, which explains the need for the previously mentioned binarization step.

To obtain a measure robust to scale change, rotation, and possible distortion in general, the correlation should be computed on transformed images. Therefore, the authors search for optimal affine transformation components, which maximize the value of normalized cross-correlation, by using an objective function based on a Gaussian kernel. Once these parameters are determined, they compute the correlation between the transformed input image and a template image. Then, they compute the correlation between the input image and the transformed template image, and finally the average of these two values is used as the match score. The authors report an average recognition rate of 70.3%, ranging from 95.5% for clear images to 24.3% for little contrast images, from the ICDAR 2003 robust OCR sample dataset.

In [9], the authors proposed a binarization method based on three steps. Firstly, color vectors of all pixels in an input image are projected onto different arbitrarily chosen axis. Secondly, they calculate a maximum between-class separability by setting an optimal threshold according to the Otsu's binarization technique [6]. Thirdly, they select the axis that gives the largest between-class separability and the corresponding threshold for binarization of the input image. Then, they decide which class corresponds to characters or background according to the ratio of black pixels to white ones on the border of the binarized image. As in their previous work [8], once the binary image is obtained, an improved GAT correlation method is applied for recognition. The authors report an average recognition rate of 81.4%, ranging from 94.5% for clear images to 39.3% for seriously distorted images, from the ICDAR 2003 robust OCR sample dataset.

One can notice that in all the works mentioned above, there is a need for a pre-processing steps (i.e. binarization) and for finding optimal tunable parameters.

In this paper, we propose a novel automatic recognition scheme for natural color scene text images, based on supervised learning, without applying any pre-processing like binarization, without making any assumptions and without using tunable parameters. Moreover, our system makes a direct use of color information and insures 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. It explains also the training process. Experimental results are reported in Section 3. Conclusions are drawn in Section 4.

2. The proposed recognition method

2.1. Architecture of the neural network

The proposed neural architecture is based on convolutional neural network architecture (CNN) [2, 4]. CNNs are hierarchical multilayered neural networks that combine three architectural ideas to ensure some degree of shift, scale, and distortion invariance:

- local receptive fields, to extract elementary features in the image

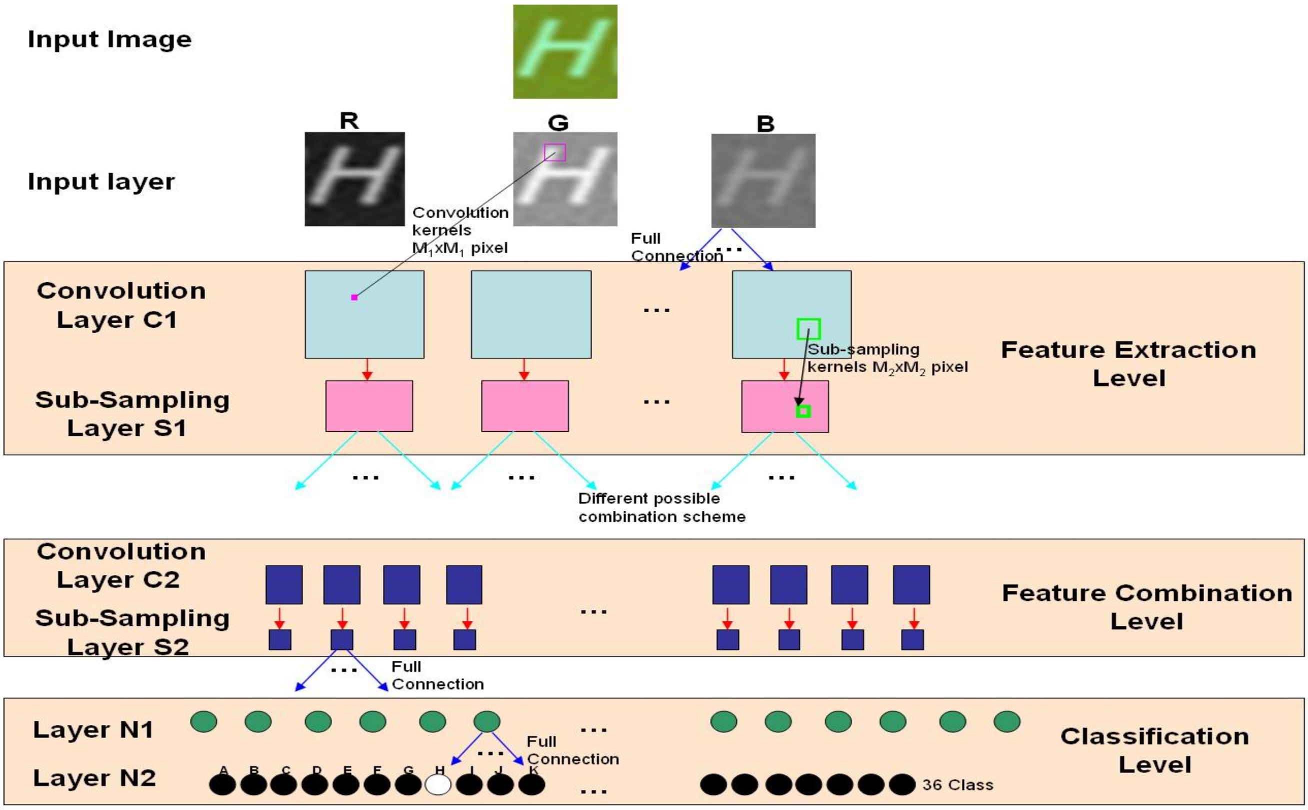

Figure 1. Architecture of the Network

- shared weights, to extract the same set of elementary features from the whole input image and to reduce the computational cost

- spatial sub-sampling, to reduce the resolution of the extracted feature maps.

As shown in Fig.1, the proposed network consists of seven different heterogeneous layers. Each layer contains feature maps which are the results of convolution, sub-sampling, or neuron unit operations. Applying and combining these automatically learnt operations ensure the extraction of robust features, leading to the automatic recognition of characters in natural images.

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, depending on the color space (RGB, YUV, CMY, etc.). Their pixel values are normalized to the range [-1, 1]. The RGB color space has been chosen in our experiments. We can distinguish three main levels:

LEVEL 1: Feature extraction level, relying on the C_1 and S_1 layers.

The layer C_1 extracts some specific features (oriented edges, corners, end points) directly from the three color channels. In fact, this layer is composed of NC_1 maps. Each unit in each map is connected to a $M_1 \times M_1$ neighborhood (biological local receptive field) in each of the color channels of the input layer. 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. Thus, 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 channels in the input layer. Multiple maps lead to the extraction of multiple features.

Once a feature is extracted its exact position is no longer important; only its relative position to other feature is relevant. Therefore, each map of the third layer S_1 results from local averaging and sub-sampling operations on a corresponding map in the previous layer C_1. So, the layer S_1 is composed of $NS_1 = NC_1$ feature maps. 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.

LEVEL 2: Feature combination level, relying on the C_2 and S_2 layers.

Layer C_2 allows extracting more complex information; outputs of different feature maps of layer S_1 are fused in order to combine different features. There are many possible combination schemes, the scheme used here will be explained later in section 2.3.

As in the first level, in this second level also, we consider a sub-sampling layer S_2, where each map results from local averaging and sub-sampling operations applied on a corresponding map in the previous layer C_2. Indeed, this progressive reduction of spatial resolution compensated by a progressive increase of the richness of the representation, which corresponds to the number of feature maps, enables a large degree of invariance to geometric transformations of the input.

LEVEL 3: Classification level, relying on the N_1 and N_2 layers. Each of them is a fully connected layer and contains classical neural units. The choice of the number of units in N_1 is empirical; whereas the number of units in N_2 depends on the number of classes to be recognized. If we consider the alphanumerical patterns, we will get 62 classes (26 lower case characters, 26 upper case characters and 10 digits). In order to reduce the number of output classes and consequently the number of neural weights to be learnt in between layers N_1 and N_2, we propose to use only 36 classes, where corresponding upper case and lower case characters are fused. This is made possible thanks to the strong generalization abilities of the proposed network.

2.2. The Database

We believe that using a public database is important to contribute to the clear understanding of the current state of the art. Therefore, we choose to use the ICDAR 2003 database, which can be downloaded from "http://algoval.essex.ac.uk/icdar/Datasets.html".

ICDAR 2003 proposed a competition named robust reading [5] to refer to text images that are beyond the capabilities of current commercial OCR packages. They chose to break down the robust reading problem into three subproblems, and run competitions for each of them, and also a competition for the best overall system. The sub-problems are text locating, character recognition and word recognition. Due to the complexity of the database, contestants participated only to the text locating contest.

In this paper, we propose to perform character recognition on the ICDAR 2003 single character database. This database is divided into three subsets: a train subset (containing 6185 images), a test subset (containing 5430 images), and a sample subset (containing 854 images). These character images are of different size (5x12, 36x47, 206x223), different fonts, different colors, and present different kinds of distortion.

We used the train and the test subsets (a total of 11615 images) for training our network. Moreover, to enhance the robustness of the system, we increased the number of images in the training set by adding the corresponding negative images, and corresponding noisy images (we add Gaussian noise) of the original dataset to the final training set. Therefore, the final training set contains 34845 images.

We tested the performance of our system on the sample subset of 854 images.

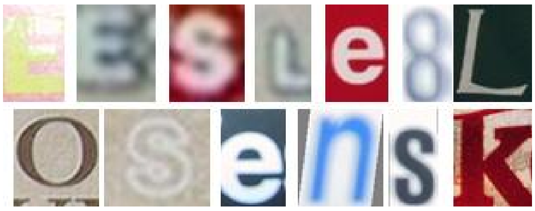

Figure 2. Examples of images of the training set

2.3. Training the network

We choose to normalize the size of all the images to 48 lines by 48 columns in RGB color space.

As mentioned before, we use 34845 color character scene images, $N_t = 30000$ in the training set and $N_v = 4845$ in the validation set. In fact, to avoid overfitting the training data and to increase the generalization ability of the system, a part of the whole training set is kept as validation set. This validation set is used to evaluate the network performance through iterations, as explained later on.

We choose to build $NC_1 = NS_1 = 6$ maps in layers C_1, and S_1; $NC_2 = NS_2 = 16$ maps in layers C_2, and S_2; 120 units in N_1; and $N_{Class} = 36$ units in layer N_2.

The combination scheme used is the following (figure 3): The first six C_2 feature maps take inputs from every contiguous subset of three feature maps in S_1. The next six take input from every contiguous subset of four feature maps in S_1. The next three take input from some discontinuous subsets of four feature maps in S_1. Finally, the last one takes input from all S_1 feature maps.

The convolution window size $M_1 \times M_1$=$M_2 \times M_2$=5×5 for both convolution layers C_1 and C_2. The sub-sampling factor is two in each direction for both sub-sampling layers S_1 and S_2.

We use linear activation functions in C_1 and C_2 and sigmoid activations fonctions in S_1, S_2, N_1 and N_2. The different parameters governing the proposed architecture, i.e.,

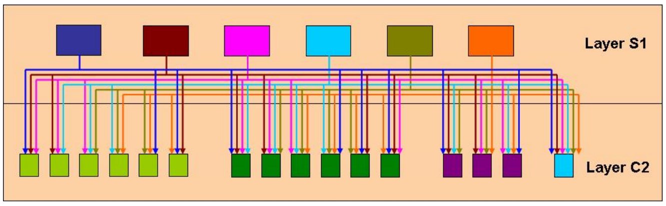

Figure 3. Feature Maps Combination Scheme

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 [4] and it consists of the following steps:

1. Construct the desired output vector $\{D_h\}_{h=1..N_{Class}}$: for a given character image, belonging to class h, this desired output vector is constructed by setting its h^{th} element to 1, and setting the remaining elements to -1.

2. Compute the actual output vector $\{O_h\}_{h=1..N_{Class}}$: the character image is presented to the network as input, the last layer output vector represents the actual output vector.

3. Update weights: the weights are updated by backpropagation of the error between the actual output vector and the desired output vector.

4. Repeat step 1 until 3 for the N_t character images of the training set.

5. Compute the MSE (Mean Square Error) over the validation set: for every charater images of the validation set, repeat step 1 and 2, and then compute the MSE between the actual output vectors $\{O_{h,k}\}_{h=1..N_{Class},k=1..N_v}$ and the desired output vectors $\{D_{h,k}\}_{h=1..N_{Class},k=1..N_v}$ as follow:

$$MSE = \frac{1}{N_v \times N_{Class}} \sum_{k=1}^{N_v} \sum_{h=1}^{N_{Class}} (O_{h,k} - D_{h,k})^2 \quad (1)$$

6. Save the weights values if MSE is decreasing.

7. Repeat steps 1 until 6, until the desired number of iterations is reached.

After training, the system is ready to recognize automatically and rapidely color character images, without the need of any binarization preprocessing. In fact, the system is able to produce directly an output vector $\{O_h\}_{h=1..N_{Class}}$ for a given color input character image. The index corresponding to the highest component of this vector is considered as the recognized class.

After 40 training iterations, the proposed network achieves an average recognition rate of 91.77% on the whole training set.

3. Experimental results

To assess the performance of the proposed method, we use the sample subset of the ICDAR 2003 character database.

Table 1. Classification of images in ICDAR 2003 robust OCR Sample dataset.

Group	Number of Images
Clear	199
Background design	130
Multi-color character	54
Nonuniform lightning	40
Little contrast	37
Blurring	210
Serious distorsion	28
Total	698

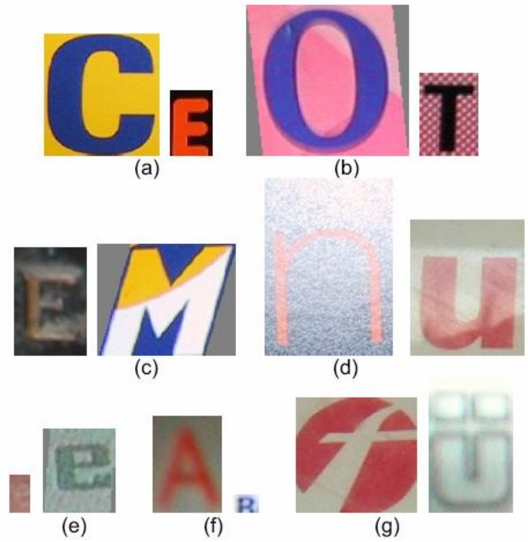

Figure 4. Examples of images in robust OCR Sample dataset classified into seven groups. (a) Clear. (b) Background design. (c) Multi-color character. (d) Nonuniform lightning. (e) Little contrast. (f) Blurring. (g) Shape distortion.

Given the wide range of variability contained in this database, and in order to compare our system to the recent works of Yokobayashi et al [8, 9], we consider the classification proposed in [8, 9] of 698 selected images from the above mentioned dataset, into seven groups according to the degree of image degradations and/or background complexity. Table 1 shows the number of images in each group and figure 4 shows examples of images in each of the seven groups.

Once training has been performed, the system is now ready to be used. We present the image to be recognized to the network after having resized it to the system retina size. Then we take the highest output of the network last layer and we consider the corresponding class as the recognized character class.

Processing the entire test set (698 images) takes about 26 seconds.

Figure 5 shows the results of our method compared to [8] and [9]. The performance of our system reaches a recognition rate of 84.53% which outperforms the methods [8] and [9]: it ranges from 67.86% for seriously distorted images to 93.47% for clear images. Compared to the methods [8] and [9], the performance of our system is less affected by the categorization of the test set, especially in the case of non-uniform lighting condition and serious distorsion, which is due to the good generalization ability of convolutional neural networks.

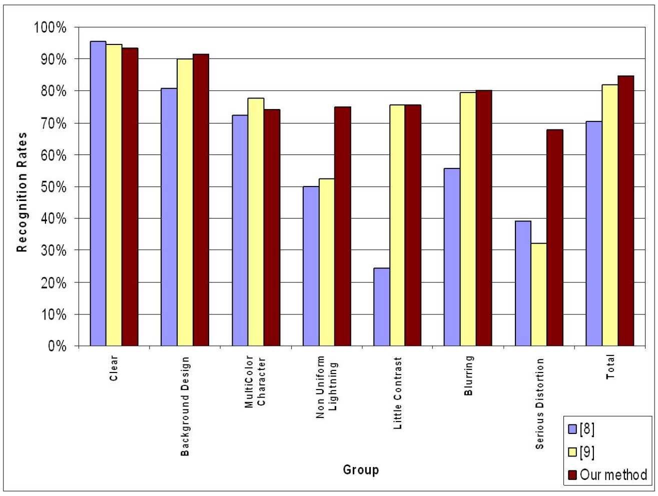

Figure 5. Recognition rates for each group of the test set

Figure 6 shows the the top N cumulative recognition rates, where the correct answer is within the N best answers (i.e. the N heighest outputs).

Here again our system outperforms the methods [8] and [9]. Furthermore, we notice that the cumulative recognition rate of the first two candidates is above 90%, showing the efficiency and the robustness of the proposed neural system.

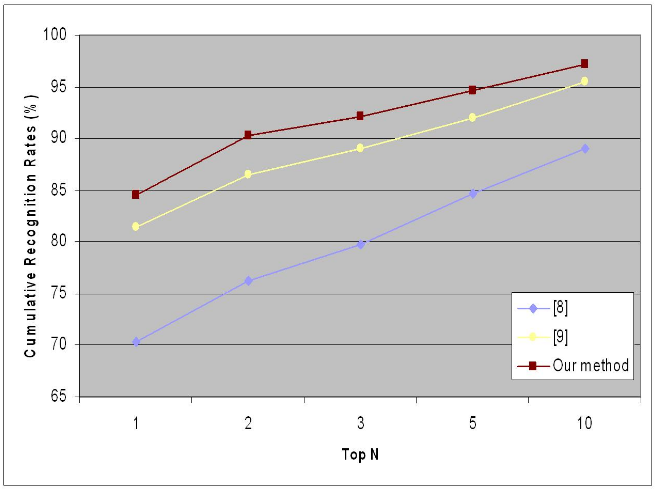

Figure 6. Cumulative recognition rates

4. Conclusion

In this paper, we have proposed an automatic recognition system for complex color scene text image recognition, based on a specific convolution neural network architecture. Thanks to supervised learning, our system does not require any tunable parameter and takes into account both color distributions and the geometrical properties of characters.

Only two state of the art methods have been tested on the public and actually challenging ICDAR 2003 robust reading data set. In this paper, we contribute to the state-of-the-art comparison efforts initiated in ICDAR 2003, and we show that our system outperforms these existing methods.

As future work, we plan to consider words recognition by including statistical language modeling in a more general network, using the proposed system has a building block.

References

[1]

[2] T. Chen, D. Ghosh, and S. Ranganath. Video-text extraction and recognition. *TENCON 2004, IEEE Region 10 Conference*, 1:319–322, Novembre 2004.

[3] C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 26(11), November 2004.

[4] S. Kopf, T. Haenselmann, and W. Effelsberg. Robust character recognition in low-resolution images and videos. Technical report, Department for Mathematics and Computer Science, University of Mannheim, April 2005.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient based learning applied to document recognition. *Proc. of the IEEE*, November 1998.

[6] N. Otsu. A threshold selection method from gray-level histogram. *SMC-9*, 1979.

[7] T. Wakahara, Y. Kimura, and A. Tomono. Affine-invariant recognition of gray-scale characters using global affine transformation correlation. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-23:384–395, 20-24 Aug. 2001.

[8] M. Yokobayashi and T. Wakahara. Segmentation and recognition of characters in scene images using selective binarization in color space and gat correlation. *Eighth International Conference on Document Analysis and Recognition ICDAR'05*, 1:167–171, 29 Aug.-1 Sept. 2005.

[9] M. Yokobayashi and T. Wakahara. Binarization and recognition of degraded characters using a maximum separability axis in color space and gat correlation. *18th International Conference on Pattern Recognition ICPR 2006*, 2:885–888, 20-24 Aug. 2006.