Text Detection in Natural Scene Images using Spatial Histograms

Shehzad Muhammad Hanif, Lionel Prevost
Université Pierre et Marie Curie Paris 06, ISIR FRE 2507
BC 252, 4 Place Jussieu 75252 Paris CEDEX 05, France
shehzad.muhammad@lisif.jussieu.fr, lionel.prevost@upmc.fr

Abstract

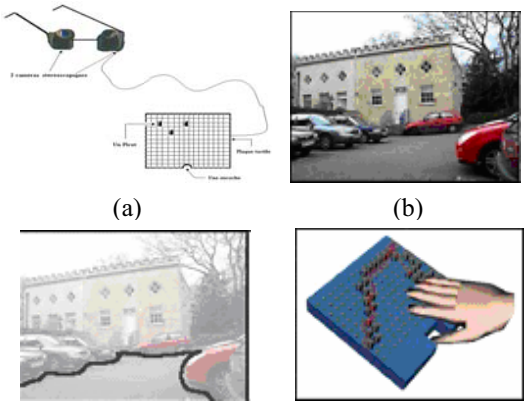
In this paper, we present a texture-based text detection scheme for detecting text in natural scene images. This is a preliminary work towards a complete system of text detection, localization and recognition in order to help visually impaired persons. We have employed spatial histograms computed from gray-level co-occurrence matrices for texture coding and three classifiers have been evaluated. Moreover, one feature selection mechanism is employed to select those histogram components that exhibit high discrimination power. The texture coding scheme is simple and can readily differentiate between text and non-text. The proposed scheme is evaluated on 50 images taken from ICDAR 2003 robust reading and text locating database. The results are encouraging with a text detection rate of 66% and a false alarms rate of 22% .

1. Introduction

This work is a part of the project called "Intelligent Glasses" [1] (Figure 1). The aim of the project is to help blind and visually impaired persons to know their environment in a better way. The Intelligent Glasses are a man-machine interface which translates visual data (such as 3D global information) onto its tactile representation. It has three parts, a bank of stereovision, a processing unit for visual perception and a handheld tactile of braille surface type. The visual data are acquired and processed by the vision system, while its tactile representation is displayed on a touch stimulating surface. In its original form, this system is able to provide information about different types of obstacles and their position with respect to user. Also, it can represent different geometrical shapes (square, rectangle, circle, arcs, curves….) on its tactile interface as well as braille symbols.

The need of textual information for blind and visually impaired persons is obvious. While taking into account this need, we have added an extra module to visual perception step of the above said system that will detect, localize and recognize the text in captured images and all this textual information will be displayed on the tactile surface.

Text detection, localization and recognition in images are regarded as basic operations in processing the captured images and are a necessary part of any application of camera based document analysis. In these recent years, they have gained a lot of attention. In general, the domain is divided into two parts based on the type of text appearing in images – one deal with super-imposed text appearing in images and videos called graphic text and other deal with the text appearing in the captured scene called scene text.



(a) (b) (c) (d)

Figure 1. Intelligent Glasses: (a) concept (b) scene (c) environment perception (d) representation

This paper addresses the problem of scene text detection in gray-level images and presents preliminary works in this domain. Our algorithm is simple and generates potential/candidate text regions

that can later be verified by a validation/verification scheme. We follow a general framework of candidate regions generation and validation as employed by various researchers [3][5][6]. We have not proposed any localization or recognition algorithm. Our detection method is based on spatial histograms computed from gray-level co-occurrence matrix (GLCM) [2]. These spatial histograms capture texture information present in the image. Based on the fact that text is a distinct texture, we can distinguish between text and non-text regions. Our classification scheme classifies image pixels as text or non-text. Furthermore, connected components can be extracted and each of them can be verified by the validation/verification scheme. In this paper, we have employed three different classifiers.

The rest of the paper is organized as follows. Section 2 contains overview of existing texture based text detection methods. Section 3 describes the construction of GLCM and computation of spatial histograms along with different classifiers used for classification. In section 4, natural scene image database and text detection results are described. Section 5 concludes this paper. Various prospects are also discussed in the section.

2. Overview of existing methods

Existing methods for text detection, localization and extraction can broadly be classified as gradient features based, color segmentation based and texture features based [8]. Here, we will concentrate on texture methods. Text is viewed as a unique texture that exhibits a certain regularity that is distinguishable from background. Humans can identify text of foreign languages even when they do not understand them largely due to its distinct texture. Various researchers have exploited this fact to detect text in images. The texture methods are largely used for text detection. Texture features can be extracted directly from the pixel's spatial relationships or from frequency data.

Wu et al.[3] proposed a texture segmentation method based on linear filtering using nine second derivatives of Gaussians filters at different scales to generate candidate text regions. The output of each filter is passed through a sigmoid transfer function. For each pixel location, local energy serves as feature vector for the pixel. The set of feature vectors is clustered using K-means algorithm. A verification algorithm is proposed by the authors to filter text-like regions.

Jung et al. [4] employed a multi-layer perceptron (MLP) classifier to discriminate between text and non-text pixels. A sliding window scans the whole image and serves as the input to neural network. Each center pixel of the window is classified as text or non-text pixel. The output image serves as a probability map where high probability areas are regarded as candidate text regions.

In [5], Crandell et al. have used a sophisticated approach to select those DCT coefficients that can distinguish text and non-text regions. They proposed text detection, localization, binarization and text tracking schemes to detect caption text from color video sequences of television channels. The scheme is claimed to work also on high contrast scene text. Text detection is based on text energy defined as sum of absolute of DCT coefficients. A subset of 19 DCT coefficients is chosen empirically by selecting coefficients with high discrimination power.

Gllavata et al. [6] used wavelet transform to perform texture analysis for text detection. A filter bank is used to extract low and high frequency sub-bands. These high frequency sub-bands are used in classification step to detect text. They have used k-means algorithm to cluster text and non-text regions.

An enhanced version of previous method is applied to color images by Saoi et al. [7] for text detection in natural scene images. In this technique, a color image is decomposed into R, G and B channels. Next wavelet transform is applied to all channels separately. High frequency sub-bands are considered for feature vector generation and k-means algorithm is employed for clustering. Contrary to previous method, this clustering is applied in a combined space.

As said earlier, texture is believed to be a rich source of visual information and it is easily perceived by humans. Thus texture methods are strong candidates to be adopted for text detection task. However, these methods are often computationally expensive and are greatly dependant on contrast of the text in an image, but lead to good results.

3. Proposed method

3.1. Texture coding scheme

We have proposed a simple texture coding method to detect scene text in gray-level natural scene images. We have used spatial histograms computed from gray-level co-occurrence matrix (GLCM) for texture coding. Gray level co-occurrence matrix $M_{(x, y)}(d, \theta)$ or second order histogram (which consider the relationship between groups of two pixels in the original image) was initially defined by Haralick [2]. Since then, GLCM has been widely used in remote-

sensing and analyzing satellite images. In most of the cases, this method is used in texture segmentation.

By simple definition, GLCM is a tabulation of how often different combinations of pixel values (gray levels) occur in an image. When divided by the total number of neighboring pixels $R_{(x, y)}(d, \theta)$ in the image, this matrix becomes the estimate of the joint probability $p_{(d, \theta, x, y)}(i,j)$ or $p(i,j)$ of two pixels, a distance d apart along a given direction θ having particular (co-occurring) gray values i and j. Moreover, x and y represent the spatial position of matrix. The dimension of GLCM is GxG where G is the number of gray-levels used to construct the matrix.

Generally, GLCM is computed over a small square window of size N centered at a certain pixel (x, y) and then window is moved by one pixel in the same manner like convolution kernel. Fine texture description requires small values of d and/or small window size, whereas coarse texture requires large values of d and/or large window size. An average over all orientations is taken so that these matrices are rotation invariant.

Figure 2 shows an example of construction of gray-level co-occurrence matrix for d = 1 and θ = {0°, 180°} and {90°, 270°}. The matrix $M_{(0, 0)}$ (1, 180°) is just the transpose of $M_{(0, 0)}$ (1, 0°). So to cover all orientations (8 in this case), we need only to compute first four orientations.

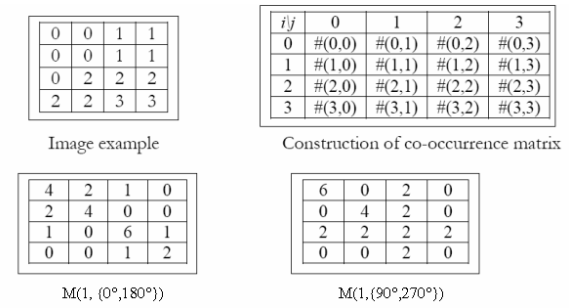


Figure 2. Construction of co-occurrence matrix[2]

3.2. Spatial Histograms

It is known that feature based algorithms are generally more stable than raw data (gray levels) based algorithms so a number of features can be calculated using the co-occurrence matrix (containing G^2 elements) for texture discrimination. Haralick defined 14 such features. Among these 14 features, contrast, entropy, homogeneity, energy are commonly used for image classification purpose.

However, in present work, we do not consider these features. As stated earlier, the GLCM represent a joint distribution $p(i,j)$ of pixels, so we can also take into account the marginal probabilities $p(i)$ or $p(j)$ of this joint distribution. Moreover, the GLCM is computed in various directions and distances and it also cover spatial relationship between the pixels, so the marginal distributions must contain information about texture, shape and spatial relationship between the pixels and represent useful information about the nature of the pixels in the image. As text and non text regions differ in their texture, shape and spatial relationships, so these probabilities are useful for the classification task. Another usefulness of marginal probabilities is their compactness as GLCM contains G^2 entries and most of them are zeros. On the other hand, marginal probability contains only G elements. Due to symmetric nature of GLCM, both probabilities $p(i)$ and $p(j)$ are equal, so we take only one of them into account. From now onward, these marginal probabilities will be called spatial histograms.

3.3. Classification

Three types of classifiers have been employed. Two of them are discriminative and third one is generative. The objective is to find a classifier that gives low false alarms and high text detection rate.

3.3.1. Maximum a posteriori (MAP) classifier. A likelihood estimate is computed based on the assumption that the posterior probability of the data (spatial histograms) is a uni-modal multivariate gaussian. Gaussian parameters i.e. mean and covariance matrix for each class are estimated on a training database by using maximum likelihood estimator. Mathematically, discriminant function based on log likelihood can be written as:

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) - \frac{G}{2}\log(2\pi) - \frac{1}{2}\log(|\boldsymbol{\Sigma}_i|) + \log P(C_i)$$

for i = 1(text), 2(non-text)

where $\boldsymbol{\mu}_i$ = mean of i^{th} class, $\boldsymbol{\Sigma}_i$ = covariance matrix of i^{th} class, $P(C_i)$ is the prior probability of i^{th} class (equally distributed on both classes) and G is the dimension of spatial histogram ($\mathbf{x}$)

In MAP classifier, for a test example, we simply find the class that maximizes the posterior probability.

$$\left(\underset{i}{\operatorname{argmax}} \; g_i(\mathbf{x}) \right)$$

3.3.2. Neural classifier. Next, we employed a multi-layer perceptron (MLP) classifier. Spatial histograms from training database are fed to neural classifier as input and the desired outputs are example labels: 1 for text, -1 for non-text. Cross-validation is used as a stopping criterion. The number of neurons in hidden cells is optimized during experimentation.

3.3.3. Text class generative model. Finally, we employed a generative model. Text class is modeled by the mean spatial histogram (MSH) estimated on the training database. A spatial histogram corresponding to a arbitrary pixel is compared to MSH through a similarity measure or distance. If that arbitrary pixel is a text pixel, then similarity measure (distance) gives a value close to 1(0), if not, the similarity (distance) value will be closer to 0(1). A simple threshold on the similarity measure will determine example's class. However, the selection of threshold is not trivial.

Furthermore, we have observed that spatial histograms do contain some zero elements so we can employ a dimensionality reduction scheme. For dimensionality reduction, we employ principal component analysis. We can see from principal components' cumulative energy curve (Figure 3) that 13 components are required to preserve 90% of energy. However, we don't want to reconstruct original data from these components. The goal is to find those components that can help in classification i.e. have high discrimination power. So retaining 13 components having 90% cumulative energy might not be the correct choice.

Hence, we have to adapt a procedure that selects an optimal threshold and optimal number of principal components in order to maximize the text detection rate and minimize false alarm rate. For this task, we employ a feature selection mechanism; more precisely it is a wrapper method for feature selection. During training, number of principal components and threshold value are found by exhaustive searching in which the similarity measure acts as part of the classifier. The percentage of false alarms is kept fixed and text detection rate is maximized on the training database by varying the number of principal components and threshold value.

4. Experimental results

4.1. Database

We have used ICDAR 2003 robust reading and text locating database [10] in our experimentation. The trial database is divided into two parts: TrialTrain and TrialTest. However, in our experimentation, we have used a total of 100 images taken from TrialTrain part. These images contain text with various font sizes, word lengths, orientations and colors. The size of images varies from 640x480 to 1024x768. There are 433 text segments in the images and font size varies from 10 pixels to 300 pixels. Out of these 100, 50 images are used for training and other 50 for test. For training different classifiers, 100,000 text examples and 100,000 non-text examples are taken randomly from 50 images. As a preprocessing step, images are converted to gray scale. No other preprocessing is employed.

4.2. Computation of gray-level co-occurrence matrices and spatial histograms

We compute GLCMs over a small square window of size N centered at a certain pixel (x, y) and then window is moved by one pixel in convolution kernel manner. GLCMs are computed in 8 directions (E, NE, N, NW, W, SW, S, SE) or (d = 1, θ = 0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°) and an average is taken so that these features are rotation invariant. In actual implementation only four orientation matrices are needed to be calculated and the other four orientations can be computed using transpose of these matrices. Moreover, five different square windows with size N = 5x5, 7x7, 9x9, 11x11, 13x13, 17x17 are used.

Due to intensive nature of computations, reduction of number of intensity levels (quantizing the image to a few levels of intensity) helps increase the speed of computation with negligible loss of textural information. The gray-levels are uniformly quantized to G levels between minimum and maximum gray levels of a window. We choose 32 gray-levels in our implementation. Once GLCMs are computed, spatial histograms can readily be computed by summing up rows or columns.

4.3. Text detector evaluation method

To evaluate the performance of a certain text detector, we adopt a pixel based evaluation mechanism. The target images (binary in nature) in ICDAR 2003 database contains one or more black (pixel values = 0) rectangular areas representing text regions. The pixel value for non-text pixels is 1. The database is designed for text localization. However, in our scheme, due to absence of localization step which generates rectangles around text strings, we have to evaluate performance of text detector with the given

target images where text regions are represented by rectangular regions and figure-ground data is missing.

The text detector generates either 0 (for text) or 1 (for non text) for each pixel of the input image. In pixel based evaluation mechanism, the output of text detector is compared with the target and a confusion matrix is created. For evaluation, two quantities, text detection rate and false alarm rate are computed.

4.4. Text detector results

In this section, we explain the training strategy and/or parameter tuning mechanism of each text detector as well as the results. Connected components can be extracted from the output binary image of the text detector in a post-processing step. Each connected component can be verified by a validation/verification scheme.

4.4.1. Maximum a posteriori (MAP) classifier. During parameter estimation of gaussian distribution, it has been observed that covariance matrices are ill-conditioned so only variances are considered (i.e. covariance matrices in the discriminant function are diagonal). However, variances are not equal. The text detector based on MAP gives a text detection rate of 72.5% and false alarm rate is 37%. The best window size is 17x17. The problem with this text detector is the high false alarms rate.

4.4.2. Neural classifier. The best neural classifier after experimentation has 32 inputs, 5 hidden cells and 2 outputs. The network was trained for 10000 epochs with cross-validation stopping. One-fourth of the training database is used in cross-validation while the rest is used for training. This text detector gives 66% text detection rate and 22% false alarm rate. The best window size is 17x17. In terms of false alarm rate, we can say that this text detector is better but text detection rate is dropped by 6%.

4.4.3. Text class generative model. As stated in §3.3.3, PCA is used for dimensionality reduction of spatial histograms. Figure 3 shows the principal components' cumulative energy curve obtained on training database for N = 17x17. It is clear that to retain 90% energy, 13 principal components are required. However as argued earlier, this may not be the correct choice for classification task.

Next we choose one similarity measure (distance) that will act as a part of the generative model. Four different similarity measures (distance), commonly

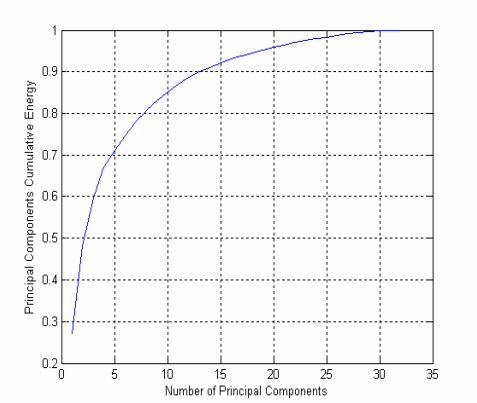


Figure 3. PCA components' cumulative energy curve

cited in literature, are used to measure similarity between two spatial histograms - say $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_G\}$ - a test example and $\mathbf{MSH} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_G\}$. These measures are:

- Cosine of the angle between two vectors

S1: $\mathrm{HSM}(\mathbf{X},\mathbf{MSH}) = \mathbf{X}^T\mathbf{MSH}/(\|\mathbf{X}\|*\|\mathbf{MSH}\|)$
where $\|.\|$ is the euclidean norm

- Histogram intersection

S2: $\mathrm{HSM}(\mathbf{X},\mathbf{MSH}) = \sum_i (\min(\mathbf{x_i},\mathbf{m_i}))/(\sum_i \mathbf{x_i})$

- Bhattacharya distance measure

S3: $\mathrm{HSM}(\mathbf{X},\mathbf{MSH}) = \sum_i (\sqrt{(\mathbf{x_i} * \mathbf{m_i})})$

- Quadratic distance

S4: $\mathrm{HSM}(\mathbf{X},\mathbf{MSH}) = \sqrt{(\mathbf{X} - \mathbf{MSH})^T \sum^{-1} (\mathbf{X} - \mathbf{MSH})}$
where $\sum$ = covariance matrix of spatial histograms for text class

The selection of similarity measure is done on the training database. The performance criterion is the discrimination power - the ratio of between-clusters-variance and within-cluster-variance. It is observed that the similarity measures S1, S2 and S3 give good results with discrimination power of 67%, 48% and 45% respectively. Quadratic distance performs worse and has a discrimination power of 36%. Histogram Intersection and Bhattacharya distance measures are designed for complete histograms, so after dimensionality reduction, we can't use them. Finally, we use S1 as a measure in feature selection due to its high discrimination power.

We want to compare the performance of this text detector with the neural classifier. So false alarm rate of neural classifier is used in this feature selection

method. Table 1. shows the number of principal components selected during feature selection method.

Table 1. Number of principal components chosen by feature selection method for different window size

Window Size	5x5	7x7	9x9	11x11	13x13	17x17
Number of principal components	32	2	2	2	2	4

The text detector gives 64.3% text detection rate and 22% false alarms on window size 17x17. Although, this text detector performs slightly poorer than the neural one but it is much simpler i.e. has few parameters. Only 4 components are used along with a simple histogram similarity criterion.

The text detection performance of above detectors is shown in figure 4. There is a slight difference in performance between neural classifier and generative model. On the other hand, maximum a posteriori classifier does not perform good due to high percentage of false alarms. The influence of window size on text detection is shown in figure 5. Characters are gradually detected as window size increases. Finally, some of the text detection results are shown in figure 6.

4.4.3. Comparison of spatial histograms with GLCM features

In this section, we will compare the performance of proposed text detectors with our previous work [9].

In our earlier work, we have used 6 features namely contrast, homogeneity, dissimilarity, entropy, energy and correlation proposed by Haralick [2], for text detection task. We employed maximum likelihood classifiers assuming mono & multi gaussian distributions for text and non-text classes and a neural classifier as text detectors. The maximum likelihood classifiers are: mono gaussian for text and non-text class (TNTSG), mono gaussian for text class (TSG), two gaussians for text and non-text class (TNTMG), two-gaussians for text class (TMG). Mahalanobis distance is used to compute likelihood estimate. The neural classifier (NC) is a two layer perceptron with 6 inputs, 20 hidden cells and 2 outputs. We have observed that two class model (TNTSG or TNTMG) is better than the single class model (TSG or TMG). Moreover, mono gaussian works better than two gaussians model. The neural classifier gives the best results: text detection rate is 64% and false alarm rate is 25%. On comparing, we can see that text detectors based on spatial histograms perform better than the GLCM features ones – an increase of 2% in text detection rate and a decrease of 3% in false alarms.

Spatial histograms are fast to compute as the number of operations required is less than that associated with GLCM features. The average time to calculate GLCM features on an image of 480x640 pixels using 17x17 window is 196 seconds while it is 135 seconds for the calculation of spatial histograms.

5. Conclusions and future work

In this paper, we have employed a simple texture coding scheme for text detection in natural scene images. We observe that spatial histograms computed from GLCM are better candidates for text detection task than GLCM features. Although, the performance is evaluated on a small test database of 50 images but the results are encouraging and we hope that performance evaluation of these text detectors on a larger database will validate these results and conclusions. We have shown that a simple generative model works equally well when compared to a neural classifier and the number of histogram's components required for effective classification is far less than the histogram dimension.

We are also working on a combination of these classifiers and hope the overall performance will improve. Currently, we have not filtered any detected text region by applying validation methods e.g. geometrical and spatial constraints, baseline detection, character alignment etc. We believe that such validation schemes will lower the false alarm rate.

Furthermore, we are exploring gradient methods as they can differentiate text and non-text regions. Gradient methods are rapid in calculation so one such method can be used to generate candidate text regions which can further be processed by our proposed texture scheme, thus making overall process fast.

6. References

[1] R. Velázquez, F. Maingreaud and Edwige E. Pissaloux, Intelligent Glasses: A New Man-Machine Interface Concept Integrating Computer Vision and Human Tactile Perception, EuroHaptics 2003, Dublin, Ireland, July 2003.

[2] Robert M. Haralick, K. Shanmugam, and Its'hak Dinstein, Textual Features for Image Classification," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, no. 6, pp.610-621, 1973.

[3] Victor Wu, Raghavan Manmatha, Edward M. Risemann, Text Finder: An Automatic System to Detect and Recognize Text in Images, IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 21, no. 11, pp. 1224-1228, 1999.

[4] Keechul Jung, Kwang I. Kim, Takeshi Kurata, Masakastu Kourogi, Jung H. Han, Text Scanner with Text Detection Technology on Image Sequences, Proceedings of 16th International Conference on Pattern Recognition (ICPR), vol. 3, pp. 473-476, 2002.

[5] David Crandall, Sameer Antani, Rangachar Kasturi, Extraction of Special Effects Caption Text Events From Digital Video, International Journal on Document Analysis and Recognition (IJDAR), vol. 5, pp. 138-157, 2003.

[6] Julinda Gllavata, Ralph Ewerth, Bernd Freisleben, Text Detection in Images Based on Unsupervised Classification of High Frequency Wavelet Coefficients, Proceedings of 17th International Conference on Pattern Recognition (ICPR), vol. 1, pp. 425-428, 2004.

[7] Tomoyuki Saoi, Hideaki Goto, Hiraoki Kobayashi, Text Detection in Color Scene Images Based on Unsupervised Clustering of Multi-channel Wavelet Features, Proceedings of Eight International Conference on Document Analysis and Recognition (ICDAR), pp. 690-694, 2005.

[8] Jian Liang, David Doermann, Huiping Li, Camera-based Analysis of Text and Documents : A Survey; International Journal on Document Analysis and Recognition (IJDAR) vol. 7, pp. 84-104, 2005

[9] _____ , Texture based Text Detection in Natural Scene Images – A Help to Blind and Visually Impaired Persons, Conference and Workshop on Assistive Technology for People with Vision and Hearing Impairments, Euro-Assist-5, Granada, Spain, August 2007 (To appear).

[10] ICDAR 2003 Robust Reading and Text Locating Competition
http://algoval.essex.ac.uk/icdar/RobustReading.html

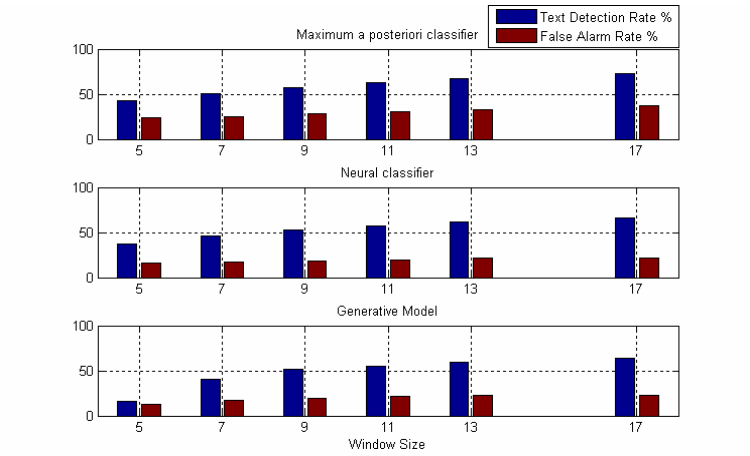


Figure 4. Performance of text detectors

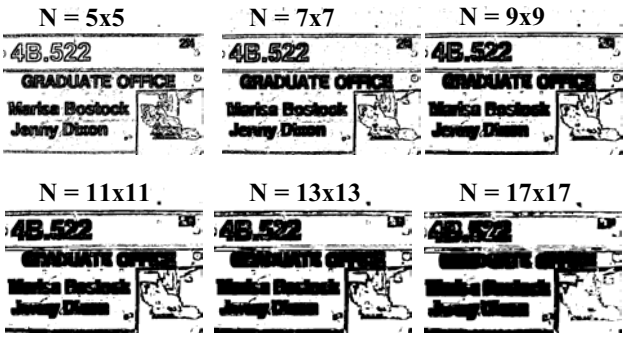


Figure 5. Effect of window size on text detection

Figure 6. Text detection examples (test database)
Text detector: neural classifier with window size 17x17