Detecting and highlighting text in images

Ivan Pakhomov
Department of Software Engineering
National Research University Higher School of Economics
Moscow, Russia
ivan_pahomov@mail.ru

***Abstract*— The work describes the problem of detecting and highlighting text in images. For the comparison, it contains the existing methods to solve this problem, advantages and disadvantages of them and, as a result, own approach to solving the task is proposed.**

Keywords—text detection; images; recognition; classifiers; classification features.

I. Introduction

Optical text recognition of the images is a very important issue, which has significant amount of practical applications: indexing photos and videos, mobile text recognition, robot navigation.

Nowadays, a digital camera is available in almost all modern phone, smartphone and tablet. The number of digital photos and videos on the Internet is increasing significantly. According to official statistics [1], 100 hours of video are uploaded to video hosting service YouTube every minute. Thus, there is a need to find a way to effectively manage these multimedia resources and analyze their contents.

The text which contains high – level semantic information is well suited to solve the problem. For example, the text contained in images on the Internet often relates to the content of web pages. The text on the covers of books and magazines is often necessary for indexing: two books with identical design but different titles will look the same if the text on the cover is unknown. News headlines as well as subtitles usually contain information about when and where the event occurred. Moreover, in contrast to other information that can be obtained from images, text is created by people, so it can directly determine the contents without any calculations.

In the modern world, people are surrounded by a vast amount of textual information such as labels, signs and billboards. Unfortunately, not everyone has the opportunity to use it. For example, a device that reads aloud can be useful for the visually impaired. However, healthy people may also face challenges, e.g. the problem of the language barrier in a foreign country. For that category of people, there are programs which can translate text from photos to the language selected by the user.

Navigating using GPS or Glonass is convenient enough, but it has some disadvantages. For example, in the places where the satellite signal is not available, it cannot be used. That's why, in case of an emergency, rescue robots need to use visual information for orientation. House numbers, signs on the buildings, road signs, various schemes – all of this can be used, but only if robot can recognize the text on them. In addition, there are many practical problems where it is necessary to be able to automatically recognize the text in the image: scan car numbers for automatic fixing of violations or mapping various organizations using panoramic images of streets.

In some cases, text selection has independent meaning. For example, time of appearance of the title in the video news shows the beginning of a new scene that can be used in automatic video summarization. Or maybe there is a need to attract the user's attention to some text.

K. Jung and colleagues [2] gave the definition of a system for obtaining information from a text image, which consists of four steps:

- *Detection*. In this step, it is determined whether there is text in the image or not.
- *Localization*. At the second stage, the location of the text is determined. Usually, the result of this step is a rectangle, which contains text.
- *Extraction*. Highlighted text areas are cleared of everything extraneous, background is removed. The text is grouped into words and symbols.
- *Recognition*. At the last stage, there is a transformation graphics to text.

Out of all stages, detection and localization of text are critical to overall system performance. Moreover, these two steps may be considered together. Finally, if the text is found, its location is determined.

In recent years, there have been suggested large number of methods to solve these problems, but quick and accurate selection of text in the photos is still a significant problem because of the large diversity of fonts, sizes, colors, methods of spatial orientation. Often, the problem is exacerbated by complex background, lighting changes, obstacles, image distortion and loss of quality in compression.

In this paper, the task is to explore some of the currently existing methods for the detection of text and to build own system to detect and highlight text in the images of poor quality.

II. Related works

The existing methods for highlighting text can be divided into two groups, based on the analysis of regions and on the analysis of connected components. Methods, based on the analysis of regions, perform texture analysis of the image fragments. The vector of values, consisting of numerical evaluations of different textural properties, is generated for each fragment, called region. This vector is input to the input classifier, which estimates the degree "text" of the region. Then, adjacent text regions are combined to produce blocks of text. From the fact that textural features of text areas differ from the signs of non-text areas, such methods can detect text even in noisy images.

Methods based on the analysis of connected components divide the whole image into separate components of any kind, for example, by color. Non-text components are discarded by means of heuristics or classifiers. Because the number of found segments is relatively small, these methods have a lower computational cost, than methods based on the analysis of regions, and the selected text components can be used directly for recognition.

Although the existing methods claim impressive results, there are several problems. The methods based on the analysis of regions are relatively slow, and their performance is sensitive to the text`s location. On the other hand, methods based on analysis of connected components cannot accurately segment text without information about the location and scale of the text. Moreover, there are many non-text components that are easily confused with the text in individual analysis. For example, wheel of car can be taken for the letter "O". Some works offer mixed approaches which use methods based on the analysis of regions and analysis of connected components.

A. *Analysis of regions*

The system described in the paper [3] is an illustrative example of the system based on the analysis of regions. Adam Coates et al proposed an interesting approach based on learning without a teacher to receive signs. They developed the system, which consists of three stages:

- Unsupervised learning algorithm. It is used to get a set of calculated features of image fragments, obtained from a training set;
- The number of attributes is reduced by the use of spatial association [4];
- The classifier is trained to select text.

At the first stage of the system, image fragments collection of 8 by 8 pixel grayscale is assembled. All fragments are pretreated. For this purpose, the intensity values and gradient of each fragment are normalized, by subtracting the values of these variables at each point of mathematical expectation and by multiplying the resulting difference by the standard deviation. Then, whitening [5] [6] based on the analysis of zero components is applied. Whitening is used for pre-processing of the vector`s features. It normalizes the values of the vectors, so that the coefficients of variation of the individual values are equal. The authors use square fragments side of 32 pixels. For each area of 8 by 8 pixels in this fragment feature vector is calculated. After that, averaging association is used, that is just a new vector is calculated as the arithmetic average of all the vectors, describing eight-pixel area. If this is not done, the amount of information will increase many times.

There is using a sliding window method for detection. The vector of signs is calculated for each thirty-second pixel image fragment. These calculations are performed for differently scaled images to detect text of different sizes. Then, each pixel of the original image is assigned with the maximum result of the classifier, got for all fragments at different scales. The obtained values are binarized with a certain threshold, and the result is a mask which indicates the presence of the text in the image. By varying the threshold of binarization, it can obtain different values of accuracy and completeness. Accuracy is calculated as the ratio of correctly labeled pixels to the total number of labeled pixels, completeness – as the ratio of the number of correctly labeled pixels to the total number of pixels that had to be mark.

This approach does not show the highest results (accuracy of 61% at a density of 69%), but is interesting, because it does not use some logically justified signs and gets them itself.

B. *Analysis of connected components*

In the paper [7], B. Epstein, E. Ofek, and E. Wexler presented the allocation method of independent components based on the operator of stroke width. The operator of stroke width – a statement that assigns each pixel of the original image to stroke width which it is most likely treated.

First, the verges are allocated using the Sobel operator. Then, starting from a point on the selected face towards increasing gradient for light text on a dark background, or decreasing, for dark on light, all the pixels to the next face are marked. For all the marked pixels, their stroke width equal to their number is indicated. If a pixel has been previously marked, the value of its stroke width varies insignificantly. Neighboring pixels, for which the stroke widths differ by no more than three times, are together grouped, and connected components are formed. Next, the filtered components are grouped into rows. At this stage, the components are filtered further. For example, individual components are not considered. Text within a line should have approximately the same stroke width, the distance between characters, character size. Additionally, the average color of adjacent characters is compared.

The results stated in the article [7] are very high: 71% accuracy at a density of 60%. It is important to emphasize, that this algorithm is quite fast, 15 times faster than the closest analogue. It is unknown how and on what machines its performance was tested. Also, due to the fact that the connected components have already been allocated, it does not need additional text extraction. As disadvantages, should be noted that, all the freestanding symbols and non-horizontal lines are discarded when filtering. This circumstance greatly reduces the completeness.

C. Kumar and A. Perrault [8] implemented the algorithm described in the previous article [7] in the Nokia mobile phone N900. In the process of implementation, they encountered some difficulties. For example, they could not simultaneously find light text on a dark background and dark on light: for this they needed to run a search twice. In addition, they had to change or omit some filtering rules components, due to poor results. In the course of such improvements, the results on a specially selected set of data: 99% of accuracy, 100% of completeness.

The disadvantages of this method can be seen from most of its description. Like all methods based on analysis of the connected components, this approach filters out a single character. Like in most methods, it is impossible to distinguish handwritten text and text with merging characters, because it will stand out as one component and discarded during filtration.

C. *Hybrid method*

The hybrid method is an attempt to combine the analysis of independent components and the analysis of regions to highlight text. Thus, in [9] a system consisting of three stages is proposed. In the first pretreatment step, an analysis region is used to find regions which can contain text. The gradient direction histogram is used as a feature, and the cascade classifier is used as a classifier. It should be noted that the purpose of this step is not an accurate selection of the text, but the definition of the probability that in a given area may be text. To do this, for each piece of image, regardless of whether it is accepted or rejected, we translate the output values of the classifier in the posterior probability using a calibration method of the classifier. Thus, probability maps are constructed for each image in the pyramid, and then they are projected to the original image and create a map of probabilities for the original image. This card is used to make adaptive binarization by Niblack adapted algorithm, in which each component is assigned a certain intensity value. The next step is a transition to the independent components analysis. To do this, we consider only the components with certain intensity values.

To analyze the components the conditional random field model is used with the following features:

- Normalized width and height.
- The ratio of width to height.
- The ratio of the number of pixels belonging to the component number of pixels belonging to the minimum bounding rectangle.
- The average value of the probability that the pixels inside the components are part of the text. For this feature, there is using the map of probabilities, obtained in the analysis phase regions.
- Compact – the ratio between the area of a rectangle describing the square perimeter and components. It allows filtering out components having a complex shape.
- The average value of the gradient at the boundary components.

From the analysis of signs, it can be clearly seen that this approach allows us to find the rows of any shape, but filters out–standing characters. The results of testing the system described: 67% accuracy, completeness 69%. The speed of this method is far superior to methods based on the analysis of regions, but inferior to the methods based on analysis of components.

III. PROPOSED APPROACH

Based on the research, following way to solve the problem is proposed as the improvement: to build a classifier stage by stage (by cascade) using boosting algorithm to enhance the weak classifiers (decision trees).

Boosting algorithms such as discrete AdaBoost, real AdaBoost, LogitBoost, Gentle AdaBoost, are used in pattern recognition problems, because they are poorly exposed retraining, compared with other machine learning algorithms. Moreover, the use of boosting allows easy use of the feature vector of high dimensionality.

All boosting algorithms are close in their structure, so therefore from now will be considered real AdaBoost, which, according to the results presented in this paper [10] is the best of all algorithms for boosting two-class classification.

A. *AdaBoost*

Real AdaBoost classifier [11] is a generalization of discrete AdaBoost [12]. Consider both of them.

Suppose we have training set $(x_1, y_1), \dots, (x_N, y_N)$, where x_i – vector of features, and $y_i \in -1,1$ – class label. Then the classifier:

$$F(x) = \sum_{m=1}^{M} c_m f_m(x),$$

where $f_m(x)$ – weak classifier, which returns a value from the set -1,1, and c_m – constants. The result of the classifier is defined as the sign of the function $F(x)$:

$$sign(F(x))$$

AdaBoost trains weak classifiers on a weighted training set, increasing the weight of the elements that were incorrectly classified. These steps are repeated for all the weighted values, and then the final classifier is represented as a linear combination of the classifiers received for each of the steps. Below is a detailed algorithm of discrete AdaBoost:

1. Initial weights are specified: $w_i = \frac{1}{N}, i = 1 \dots N$
2. For each $m = 1,2, \dots, N$
   a. Train classifier $f_m(x) \in -1,1$ using a weighted training set.
   b. Calculate the classification error

      $$err_m = E_w\left(1_{(y \neq f_m(x))}\right), c_m = \log(\frac{1-err_m}{err_m}), ,$$

      E_w – the mathematical expectation calculated for the weighted training set, and $1_{(s)}$ – identifier of the set S.
   c. Refresh weights

$$w_i = w_i \exp\left[c_m 1_{(y \neq f_m(x))}\right],$$

$i = 1,2,\dots,N$ and normalize them, so that $\sum_i w_i = 1$, $S_w = \sum_{i=1}^{N} w_i, w_i = \frac{w_i}{S_w}$

At each iteration, the weights of misclassified vectors are increased by the value determined by the classification weighted error.

d. The result: $sign(\sum_{m=1}^{M} c_m f_m(x))$

L. Breiman [13] demonstrated that the use of classifiers based on trees as weak classifiers gives good results. Besides, various tests have shown that increasing the number of classifiers increases the accuracy of the algorithm on the test sets, which demonstrates the stability of AdaBoost to overfitting.

Real AdaBoost is a generalization of discrete AdaBoost. It uses predictors that return the probability of belonging to the class. The set of values of the weak classifier is already in the field of real numbers. $sign(f_m(x))$ – defines a class and $|f_m(x)|$ – the probability. Below is the algorithm of real AdaBoost:

1. Initial weights are specified: $w_i = \frac{1}{N}, i = 1 \dots N$
2. For each $m = 1,2,\dots,N$
   a. $p_m(x) = P_w(y = 1|x) \in [0,1]$
   b. $f_m(x) \longleftarrow \frac{1}{2}\log\left(\frac{p_m(X)}{1- p_m(X)}\right) \in$ R
   c. Refresh weights $w_i = w_i \exp[-y_i f_m(x_i)]$ $i = 1,2,\dots,N$ and normalize them, so that $\sum_i w_i = 1$, $S_w = \sum_{i=1}^{N} w_i, w_i = \frac{w_i}{S_w}$
   d. The result: $sign(\sum_{m=1}^{M} f_m(x))$

To reduce the computation time for these models without a significant loss in accuracy technique of circumcision is used. In the course of the algorithm, while the number of trees is increasing, large number of examples from the training set is classified correctly. Therefore, weight of these examples is decreased. Examples with low weights give a small contribution to training of the weak classifiers. Therefore, these examples may be removed at training, without a lot of harm to the learning outcomes of the weak classifier. For this purpose the threshold value may be specified for cutting the training set. It should be noted that this procedure is repeated for each weak classifier, and these clipped examples can be used for training in the following stages.

B. Decision trees

Decision tree – is a balanced binary tree, which can be used for classification and for regression tasks. The procedure of the prediction begins with the root vertex. From the each non-leaf vertex prediction procedure goes to the left or right, depending on the value of the specified variable feature vector, whose index is stored in the current node. The value of the variable is compared with a threshold value stored in the node. If the value is less than the threshold, the procedure goes to the left, otherwise – to the right. Each node uses a pair of index and variable threshold. This pair is called a partition. When it reaches a leaf node, the value stored in it is used as a result of the classifier. Trees are built recursively starting from the root node. All the training set is used to split root. At each node, the best split is chosen using some criterion. In machine learning classification Gini impurity is used [13]. This criterion is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset. To compute Gini impurity for a set of items, suppose i takes on values in $\{1, 2, \dots, m\}$, and let f_i be the fraction of items labeled with value i in the set:

$$I_{G(f)} = \sum_{i=1}^{m} f_i(1 - f_i) = \sum_{i=1}^{m} (f_i - f_i^2) = 1 - \sum_{i=1}^{m} f_i^2$$

All data are separated in accordance with the selected partition into two subsets, which are used for training of the left and right subtrees. Recursive procedure can stop at one of the following cases:

- maximum depth of the tree is reached;
- power training set in the node is less than a given threshold, and is not representative;
- all examples in the node belong to the same set, or, in the case of the regression, the variation between them is small;
- the best selected separation does not give significant gain in comparison with a random choice.

C. Viola–Johns classifier

P. Viola and M. Jones [14] presented an interesting approach to the construction of classifiers, which they used to find human faces in the image. This idea is to build a cascade of multiple classifiers, and discard individual fragments gradually, at each step.

It is well known that more complex classification function is more accurate, but its ability to generalize is smaller. Minimization of structural risks provides a formal method for selecting classifier with the right balance of complexity and correctness, in the case, where the main factor is to reduce the errors.

Another important limitation is the computational complexity. The computation time and error – completely different things, that's why it is theoretically impossible to choose an optimal balance. Nevertheless, for many classification functions computation time depends on the structural complexity. In such cases, the cost reduction of time depends on the complexity reduction. Nevertheless, the computation time depends on the structural complexity for many classification functions. . In such cases, the reduction of time cost depends on the reducing the complexity.

But this direct analogy does not work in applications where the balance between the marks strongly shifted towards any class. For example, when you select text, there are thousands of fragments without the text and only a small part contains the text. Oddly enough, there are good results can be achieved with a sufficiently high percentage of correctly classified fragments and very fast classification.

The key point is that despite the fact that it is impossible to build a simple classifier with a very low error rate, but, in some cases, it is possible to build a classifier, which is practically never wrong to classify text areas.

For example, it is possible to create a very fast classifier, which is correct in the text fields, but wrong in 90% of cases on the non-text areas. Such a detector can be used for pre-filtering: if the fragment is marked as "no text", it immediately discarded, and if the fragment is marked as "text", it requires an additional classification. Sequence of such classifiers (each following is harder, slower and more accurate that the last) is a cascade of classifiers.

For classification, slightly modified AdaBoost can be used. Normal AdaBoost tries to minimize the total error at each step, but we want to minimize the number of misclassified text regions. The idea is to change the balance of the scales in favor of the data with positive marks. But this is not enough just to change the balance in the first step, because AdaBoost weights will be changed and the second weak classifier will be symmetric. Therefore, it is necessary in addition to the initial balance variation, to change the standard AdaBoost formula:

$$w_i = w_i \exp[\frac{1}{N} y_i \log(\sqrt{k}) - y_i f_m(x_i)]$$

$i = 1,2,\dots,N,$ where k – positive versus negative preference coefficient.

IV. Features for classification

In the classification a large number of different features is used. There are ten the most commonly used:

- The standard deviation of intensity

  If the value of the intensity varies very slightly within the limits image fragment, it is likely that this fragment does not contain any text.

- The standard deviation of the gradient

  This criterion shows the range of values of the gradient within the fragment. Can filter out solid areas, but requires additional calculation of the gradient values.

- Entropy

  Can be calculated by the formula:

  $$H(x) = -\sum_{i=1}^{n} p(i) log_2 p(i)$$

- Statistics of derivative by X

  This feature uses the observations described in the article [9], consisting of the different values of the derivative in different regions containing the text

- Statistics of derivative by Y

  Is identical to statistics of derivative by X

- Histogram of gradient values

  Before calculating characteristic values, original image fragment is always scaled to the size of 20 by 20 pixels, thereby obviating the need for normalization values of the histogram. Values divided into twenty equal intervals, whereby this feature gives us the twenty values.

- Histogram of the intensity values

  Is identical to the histogram of gradient values

- Histogram of intensity gradient

  This histogram is built on two values. All quantities that are included in one interval of intensity histogram are divided into intervals by value gradient Splitting happens for twenty intervals. In total, the result is four hundreds values.

- Histogram of bar width

  The idea of this feature is that the bar width distribution in text areas are different from the bar width distribution in non-next areas. Before calculating, the image is compressed to a size of 20 by 20 pixels, to increase the speed of operation, because the transform is a time-consuming operation. All values are distributed, as in other characters, using histograms, at the twenty slots.

- Separation of Gaussians

  This feature shows how many divided the expectations of Gaussians. If they almost merge with each other, then the background and text are not distinguishable. Can be calculated by the formula:

  $$\Delta M = \frac{|M(G_1) - M(G_2)|}{255},$$

  where $M(G_1)$ – the mathematical expectation of the Gaussian;

  255 – the maximum difference in the intensities of eight-bit image.

Thus, using the features listed above, as well as combinations thereof, different results can be obtained, as different speed and precision. For example, we may proceed from operation speed characteristics: at the first cascades we can use faster but less accurate features and slower but more accurate on the latter.

V. Low quality images

There is pretreatment required, for the images with low resolution and quality to remove noise, to smooth background and to enhance the contrast between text and background. For this purpose, Weiner filter [15] is suitable. Also, in this case, the bilateral filter is appropriate to use. Bilateral filter allows to enhance the contrast between areas of different colors and to make transitions sharper. The latter is particularly important in cases where the text is almost completely blends into the background.

For example, in figure 1 graph of the results of the classifier on the worst images of the bilateral filter parameters: radius and range of weight. It can be seen that the harmonic mean varies widely and can exceed eight times the results obtained without pre-filtering.

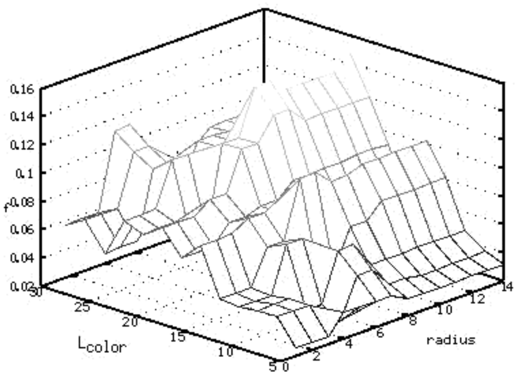

Fig. 1. Use of bilateral filter

VI. Results

The system was developed using C++ programming language. It was trained and tested on ICDAR 2013 data sets. Testing was carried out according to the procedure, described in the article [3]. All fragments of the image, which were classified and labeled as text, formed a mask. For the true values markup dataset was used. Further, completeness and accuracy were calculated using the following formulas:

$$Accuracy = \frac{N_{true+}}{N_+}$$

$$Completeness = \frac{N_{true+}}{N_{true+} - N_{true-} + N_-}$$

where: N_+– the number of pixels classified as text;

N_- – the number of pixels classified as non-text;

N_{true+} – the number of text pixels classified as text;

N_{true-} – the number of non-text pixels classified as – non-text.

The results of the classifier for now: accuracy – 51% completeness – 88%. Figure 2 shows the image on which the classifier showed the best result. This image characterized by very good contrast between text and background, weak blurring and it is easily readable.

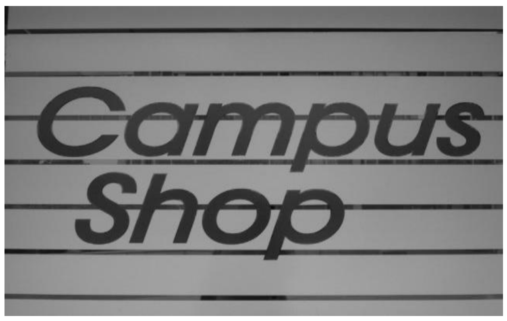

Fig. 2. Image with the best classifier`s result

Figure 3 presents image on which the classifier showed the worst results. The text on this image almost merges with the background, and the stones, that stood out as the text, look much clearer.

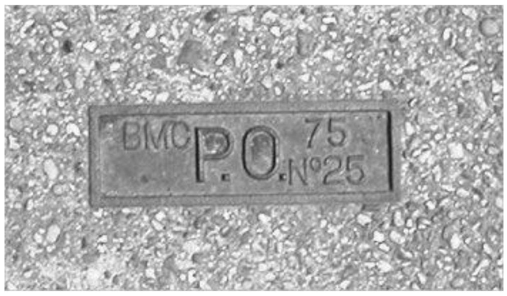

Fig. 3. Image with the worst classifier`s result

Of course, there is still much work to optimize a system highlighting and detecting text, but the most importantly, the valuable theoretical basis has been laid, and, on this basis, it is possible to construct an effective and power system to solve not only this problem, but related problems too.

Conclusions and further work

The problem of this paper is considered upon the problem of detecting and highlighting text in images. Its importance in today's multimedia world was emphasized and the possibility of its application was described. Further, the most known and widely used approaches to solving the problem were considered as well as their pros and cons, and possible update options. On this basis, the more optimal approach was offered and the nuances of the implementation were discussed as well as results. As already mentioned above, the proposed system has a great list of practical applications, but immediate plans include the solution of graphic content indexing problem in the video, and, possibly, porting it to the mobile device to create a mobile application for the recognition of text from the photos taken on the device.

References

[1] YouTube Press (2014). *YouTube Official Statistics.* Retrieved April 1, 2014, from http://www.youtube.com/yt/press/statistics.html

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

[3] Coates A., Carpenter B., Satheesh S. Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning.

Document Analysis and Recognition, *ICDAR 2011 International Conference*, 2011, 440-445.

[4] Boureau Y.L., Bach F., LeCun, Y. Learning mid-level features for recognition. *Computer Vision and Pattern Recognition (CVPR)*, 2010 IEEE Conference, 2559–2566.

[5] Hyvarinen A., Oja E. Independent component analysis: algorithms and applications. *Neural Networks Oxford*, 2000, 13(4-5), 411– 430.

[6] Picard Rosalind W. *Decorrelating and then Whitening data.* Retrieved December 25, 2013, from http://courses.media.mit.edu/2010fall/mas622j/whiten.pdf

[7] Epshtein B., Eyal O., Yonatan W. *Detecting Text in Natural Scenes with Stroke Width Transform.* Retrieved December 20, 2013, from http://www.math.tau.ac.il/~turkel/imagepapers/text_detection.pdf

[8] Kumar S., Perrault A. *Text Detection on Nokia N900 Using Stroke Width Transform.* Retrieved November 19, 2013, from http://www.cs.cornell.edu/courses/cs4670/2010fa/projects/final/results/group_of_arp86_sk2357/Writeup.pdf

[9] Yi-Feng P., Xinwen H., Cheng-Lin L. A Hybrid Approach to Detect and Localize Texts in Natural Scene Images, 2011, *NJ, USA: Trans. Img. Proc. Piscataway.*

[10] Friedman Jerome, Hastie Trevor, Tibshirani Robert. Additive Logistic Regression: a Statistical View of Boosting. *Annals of Statistics.* 1998. T. 28. c. 2000.

[11] Schapire Robert E., Singer Yoram. Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning.* 1999. T. 37, № 3. C. 297– 336.

[12] Freund Yoav, Schapire Robert E. Experiments with a New Boosting Algorithm. *ICML. Morgan Kaufmann,* 1996. C. 148– 156

[13] Breiman, L., Friedman, J. H. Classification and regression trees. *Wadsworth Publishing Company*, 1984.

[14] Viola Paul, Jones Michael. Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade. Advances in Neural Information Processing System 14. *MIT Press*, 2001. C. 1311–1318.

[15] Wiener Norbert. Extrapolation, Interpolation, and Smoothing of Stationary Time Series. *The MIT Press*, 1964.