2011 International Conference on Document Analysis and Recognition

AdaBoost for Text Detection in Natural Scene

Jung-Jin Lee*, Pyoung-Hean Lee*, Seong-Whan Lee*, Alan Yuille*† and Christof Koch*‡
*Department of Brain and Cognitive Engineering, Korea University, Seoul, 136-713, Korea
†Departments of Statistics, Psychology, Computer Science, UCLA, Los Angeles, California 90095, USA
‡Computation and Neural Systems, California Institute of Technology, Pasadena, California 91125, USA
Email: millwill@korea.ac.kr, leeplus@korea.ac.kr, swlee@image.korea.ac.kr, yuille@stat.ucla.edu, koch@klab.caltech.edu

***Abstract*—Detecting text regions in natural scenes is an important part of computer vision. We propose a novel text detection algorithm that extracts six different classes features of text, and uses Modest AdaBoost with multi-scale sequential search. Experiments show that our algorithm can detect text regions with a f= 0.70, from the ICDAR 2003 datasets which include images with text of various fonts, sizes, colors, alphabets and scripts.**

***Keywords*-text detection; text location; AdaBoost;**

I. Introduction

Detecting text in natural scenes, such as sign boards on streets and buildings, advertisements, traffic signs, movie marques and so on, is a core part of computer vision applications, including robotics, vehicle license plate recognition system and text reading programs for visually impaired person. Text detection consists of two steps. The first step involves detecting text regions in a given image, while the second step retrieves text information from these regions using OCR or other technologies. We here concentrate on the first step of text region detection in natural scenes.

Our system is based on the Modest AdaBoost algorithm which constructs a strong classifier from a combination of weak classifiers. In this paper, we use nodes of Classification And Regression Tree (CART) [1], a nonparametric decision tree that determines outcome variables from among a large number of features, as weak classifiers of our AdaBoost algorithm reduces false positives by using post adjustment.

II. Previous Work

A good overview of text detection algorithms can be found in ICDAR 2003 [2] and 2005 [3]. These competitions compared these algorithms in same environment for fair evaluation. Previous approaches can be divided according to whether or not they use machine learning techniques. A representative example for learning is Chen and Yuille, who used AdaBoost learning of joint-probabilities of features (X and Y derivatives, histogram of intensity and edge linking) [4]. On the other hand, Epshtein *et al.* did not exploit learning techniques, focussing instead on the fact that text has a constant stroke width. It reached a precision of 0.73 and a recall of 0.60 [5].

Figure 1. Some of the 495 ICDAR images used for testing

III. Datasets

We used two, publicly available, sets of natural images containing text; ICDAR 2003 (499 images) and a dataset of 307 images from Microsoft Research India (MSRI) [5]. The ICDAR dataset is an official standard in the recurrent text detection competitions [2], [3]. We used the MSRI images to train the AdaBoost-based algorithm and the ICDAR images to test it.

IV. Features

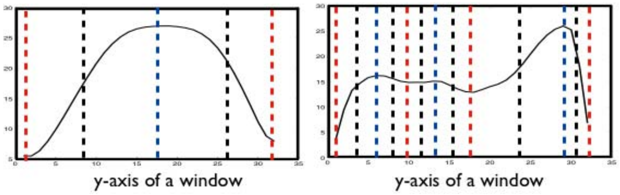
AdaBoost constructs a strong classifier from a combination of weak classifiers. We use 6 types of features sets for the CART weak classifier.

A. Variance and Expectation of X-Y Derivatives

Chen and Yuille used features based on the mean and standard deviation of X and Y derivatives [4]. For text, the distribution of the X derivative should have a concave shape, while the distribution of the Y derivative should be inconsistent (Fig. 2). We separate block areas based on local minima and maxima of the distribution of X and Y derivatives.

We empirically found that X derivatives only have a single local maxima without any local minima. To compute the X derivatives, we divided the ROI into three sub-blocks of images by calculating two borderlines using local maxima $(B_1 = [1, \frac{1+L_M}{2}), B_2 = [\frac{1+L_M}{2}, \frac{H-L_M}{2}], B_3 = (\frac{H-L_M}{2}, H]$; $B_i : i^{th}$ block, L_M : local maxima, H: height of a window).

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.93



(a)

(b)

Figure 2. (a) A graph represents distribution of the X-derivatives of text samples. (b) A graph represents distribution of the Y-derivatives of text samples. Blue dots mark local maxima, red dots mark local minima and black dots indicate mean of local maxima and minima.

For Y derivatives, we calculated three different types of borders based on local minima, local maxima and the mean of local minima and maxima ($B_1 = [1, \frac{1+L_k}{2}), B_{k+1} = [\frac{L_k+L_{k+1}}{2}, \frac{L_{k+1}+L_{k+2}}{2}, B_n = (\frac{L_{n-1}+L_n}{2}, H]$, where $B_i : i^{th}$ block, $L \in \{$local maxima, local minima, mean of local maxima and minima$\}$, n :number of points of each type of L, $L_k : k^{th}$ point of L). We extract a set of features by calculating the variance and expectation of each area separated.

B. Local Energy of Gabor Filter

Even though text includes letters of a variety of sizes, shapes and orientations, it tends to have higher spatial frequency components compare to non-text [6]. We used local energy to extract these high frequency components in four orientations.

We used four different orientations ($\theta = 0, \frac{\pi}{4}, \frac{\pi}{2}$ and $\frac{3}{4}\pi$) with three different radial frequency $f(0.2, 0.8 and 0.9)$ and σ channels ($\sqrt{3.5}, 1$ and $\sqrt{2.5}$).

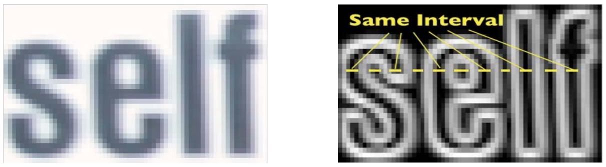
C. Statistical Texture Measure of Image Histogram

Statistical texture information is commonly used in image retrieval problems [7]. We here use 6 statistical texture measures of image histogram to differentiate text from non-text regions. Defining μ as the average of the intensity, Z_i a random variable indicating intensity, $p(Z)$ the histogram of intensity level in the image and L number of possible intensity level, we use the following six features

- *Variance of Histogram*:$\sum_{i=0}^{L-1}(Z_i - \mu)^2 p(Z_i)$
- *Squared sum of probability*: $\sum_{i=0}^{L-1} p^2(Z_i)$
- *Average Entropy*: $-\sum_{i=0}^{L-1} p(Z_i) \log_i p(Z_i)$
- *Relative Smoothness*: $1 - \frac{1}{1+\sigma^2}$
- *Skewness*: $\sum_{i=0}^{L-1}(Z_i - \mu)^3 p(Z_i)$
- *Kurtosis (Peakedness)*: $\sum_{i=0}^{L-1}(Z_i - \mu)^4 p(Z_i)$.

D. Measurement of Variance of Wavelet Coefficient

The discrete wavelet transform (DWT) transforms an image into an orthogonal wavelet form. For fast calculation of DWT, 4 approximation coefficients are needed (approximation, horizontal, vertical and diagonal coefficient) computed from low-pass decomposition filter. These coefficients can be used for reconstructing the original image. We use *Deubechies* wavelets for extracting these coefficients and calculate the variance of these 4 coefficients vector to be used as four features.



(a) Text in window

(b) Applying color gradient

Figure 3. Concept of color interval. (b) Illustrates edges from color gradient. Note that the intervals between adjacent edges are similar.

E. Edge Detection and Edge Interval Calculation

We use color gradients in RGB space to extract edge information [8].

Text has constant intervals between edges with similar size. We can construct a set of features based on this characteristic. We compute the size of intervals between edges in images. If a window has text, the size of interval has a skewed shape and the standard deviation of the interval size is smaller than that of non-text (Fig. 3). We use the mean and standard deviation of these intervals as two features.

F. Connected Component Analysis

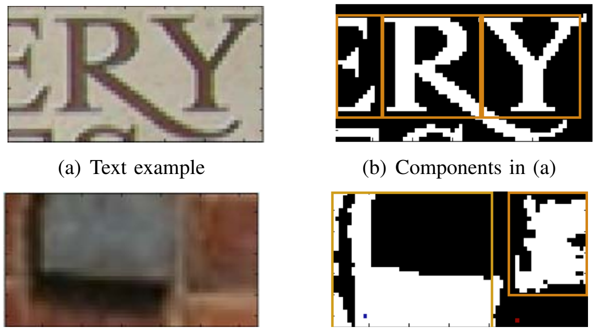
Text in natural scenes typically has two color components (background and foreground) that are aligned in the window. Contrariwise, non-texts often contains more than two colors. We applied k-means cluster over each window with $k = 2$ to discriminate text from non-text (Fig. 4). We extracted three features from this component analysis.

- *Component alignment*: If a window is well-fitted over a text area, the y coordinates of the center of each component are close to the center of the window. We used average distance among the components and the center coordinates along the y axis of the window as features.
- *Standard deviation of component location*: Although components alignment is useful in detecting text, some non-text components have small mean with high standard deviation. We use the standard deviation of the component location as a feature to overcome this problem.
- *Standard deviation of component size*: The size of text components are relatively similar to each other compare to the size of non-text components. We calculated standard deviation of each component size.

V. Proposed Scheme

A. Overall System Design

The complete algorithm (Fig. 5) has three phases: preprocessing (Section C), multi-scale sequential search (Sec-



(a) Text example (b) Components in (a)

(c) Non-text example (d) Components in (c)

Figure 4. Examples of connected components.

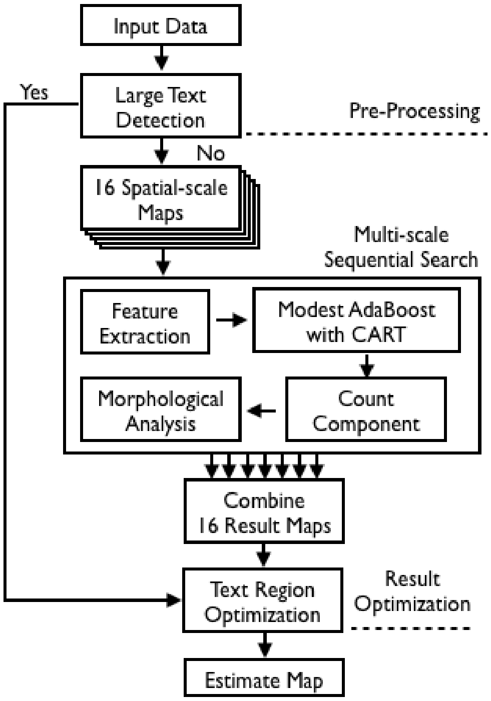


Figure 5. Design of the overall system architecture

Algorithm 1 Pseudocode of Modest AdaBoost

```
Step 1: Initialize data weights $\omega_0(x) = \frac{1}{N}$ with given training data $(x_1, y_1), \ldots, (x_N, y_N)$, with $N$ the number of training image.
Step 2: for m = 1 to M (Max iteration)
    1) Train weak classifier $h_m(x)$ by weighted least squares of $x_i$ to $y_i$ with weights $\omega_i$.
    2) Compute inverted distribution $\bar{\omega}_m = 1 - \omega_m$ and renormalize by $\bar{Z}_m$.
    3) Compute :
       $P_m^{+1} = P_{\omega_m}(y = +1, h_m(x))$
       $P_m^{-1} = P_{\omega_m}(y = -1, h_m(x))$
       $\bar{P}_m^{+1} = P_{\bar{\omega}_m}(y = +1, h_m(x))$
       $\bar{P}_m^{-1} = P_{\bar{\omega}_m}(y = -1, h_m(x))$
    4) Set $H_m(x) = P_m^{+1}(1 - \bar{P}_m^{+1}) - P_m^{-1}(1 - \bar{P}_m^{-1})$
    5) Update $\omega_{m+1}(i) = \frac{\omega_m(i) \cdot exp(-y_i \cdot (h_m(x_i) - \frac{m}{M}\log\sqrt{k}))}{Z_m}$
Step 3: Produce the final classifier
    $H(x) = \sum_{m=1}^{M} H_m(x)$
```

tion D) and text region optimization (Section E). Before start the system, we need to process learning with Modest AdaBoost (Section B).

B. Learning with AdaBoost

AdaBoost [9] is an effective machine learning method for classifying two or more classes. AdaBoost enhances the performance of a set of weak classifiers $\{h_m(x)\}$ - each of which has a performance that might only be marginally better than chances - by combining them into a strong classifier $H(x)$ using scalar weights $\{\alpha_m\}$ in each round t with input data x:

$$H(x) = sign\left(\sum_{m=1}^{M} \alpha_m \cdot h_m(x)\right) \quad (1)$$

There are several boosting algorithms that improve performance of vanilla-flavored AdaBoost. *Real AdaBoost* [10] computes the probability that a given pattern belongs to a class to perform optimization with respect to $h_m(x)$. *Gentle AdaBoost* [11] exploits weighted least-squares regression for deriving a reliable and stable ensemble of weak classifiers. We here use *Modest AdaBoost* [12] (see Algorithm 1) which modifies Gentle AdaBoost with an inverted distribution. For our dataset, it performs superior to Gentle and Real AdaBoost in tests.

The fraction of these images occupied by text is small. Put differently, there are many more non-text than text windows. To reflect this preponderance of non-text, we extract 10,000 text windows as positive samples and 40,000 non-text windows as negative samples (each window is 64 pixels wide and 32 pixels tall) from the 307 MSRI images to train our classifiers using Modest AdaBoost. Positive samples were obtained by hand-labeling and negative samples were extracted by a bootstrap process with random selection.

To allocate the appropriate weights to text versus non-text examples, we use asymmetric AdaBoost method [13], by reinterpreting the weak classifier $h'_m = h_m(x) - \frac{m}{M}\log\sqrt{k}$ with cost $k = 4$. CART constructs a decision tree by using 6 set of features which were extracted from positive 10,000 and negative 40,000 samples. Branches of the tree were indicated true and false, and nodes were constructed using these training samples. And each node of CART is used as an individual weak classifier of the Modest AdaBoost with maximum depth of CART equal to 5, and we set the maximum boosting steps (M) to 100 using AdaBoost toolbox [14].

C. Pre-Processing of the Dataset before Learning

We detect text regions using sequential search with a designated size of window sufficient to extract most textual elements. However, some images contain extremely large text in close-up. This makes text detection extremely challenging. To address this problem, we add a "large text detection" module as a pre-process. In this phase, it classifies images with extremely large text and extracts text region from the image. We resize the whole image to the size of the window (64×32 pixels) and apply Modest AdaBoost on the resized image. To ensure it has a large text, we analyze the resized image that has passed the text for likely text using connected components, and extract component areas as a text region.

D. Multi-scale Sequential Search

The size of text varies considerably, from 16×16 to 487×720 for the ICDAR images. We therefore use multi-scale images with 16 different spatial scales. The size of spatial scale increases linearly from 64×48 (width×height) to 1024×768 pixels. We generate a 64×32 window to search over an entire image with steps of 32 pixels in the x and 16 pixels in the y directions within each map. Each window performs a search procedure which need to pass through 4 steps. First, we extract features from images in a window and apply Modest AdaBoost to classify these as text or non-text.

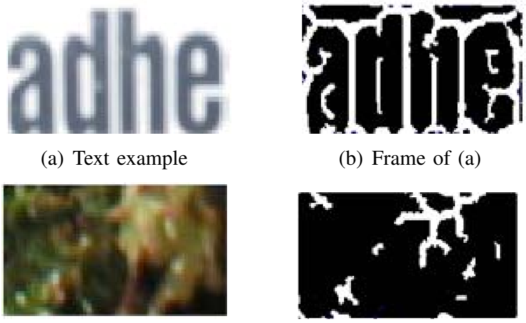
Rather than forcing the classifier to respond to a given window with 0 or 1, we use probabilities. The output of the filter $H(x)$ is the output of the final classifier of Modest AdaBoost (see Algorithm 1)

$$H(x) = \log \frac{p(W|y = text)}{p(W|y = nontext)} \quad (2)$$

To reduce false positives, we employ two additional methods, counting the numbers of component and morphological analysis, after AdaBoost classified a particular window as text.

1) Counting Numbers of Component: Most of text and complex non-text windows have a number of components with strong X and Y derivatives compared to those of non-text windows with simple patterns with strong orientations such as single line. We therefore multiply the number of components of the X derivatives in any one window with the number of y components in that window for additional discrimination.

2) Morphological Analysis: A morphological operation called 'skel' turns an image into a skeletal image [15]. After applying 'skel' operation with 5 iterations to each window, skeletal frames remain. We use these to distinguish false positives. Characters in a window tend to have similar properties such as color, intensity and so on. However, many non-text windows have different properties between objects within the window. This skews the result of the morphological operation. Thus, we regard these results from morphological operation as a property of non-text samples, and automatically remove them based on the skewness of the distribution (Fig. 6).



(a) Text example (b) Frame of (a)

(c) Non-text example (d) Frame of (c)

Figure 6. Examples of applying morphological operation.

E. Text Region Optimization

After all sequential processes are completed, we linearly combine the resulting maps at 16 spatial scales with equal weights into a single 1024×768 map. The estimated text regions are not perfect rectangular. These regions pass through a text region optimization stage, which maximizes the expected text region by constructing a rectangular region based on minimum and maximum positions of original region. In a second step, we derive an edge map from color gradient and use it as a criteria of region optimization to remove surrounding parts of the text window that do not contain text.

VI. Evaluation

To evaluate our algorithm, we employ the publicly accessible benchmark of natural scenes containing text [16] used in the ICDAR 2003 [2] and 2005 [3] competitions. The fact that the testing images derive from a different image dataset than the training images maximally challenges the generalization abilities of our method. We use the entire set of ICDAR images except 4 non-text images, for a total of 495 images.

The performance metrics are *precision*, the fraction of text windows which are correctly classified as text, and *recall*, the fraction of all text windows have been correctly identified. Finally f is a single scalar that is the harmonic mean of the precision and recall. For optimal performance, all three numbers should be unity.

We conduct a comparison to clarify contribution of features sets (Fig. 7). Even though each feature set yields

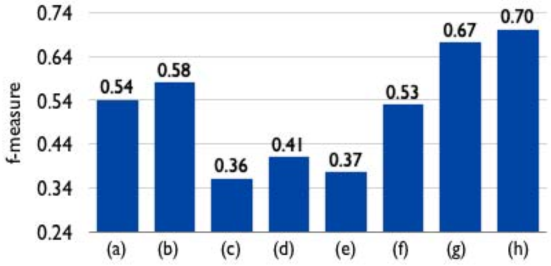


Figure 7. How do the different features contribute toward the overall AdaBoost performance? Shown is the f number when different features are used: (a) X-Y derivatives, (b) Local energy of Gabor filter, (c) Statistical texture measure of image histogram, (d) Measurement of variance of wavelet coefficient, (e) Edge interval and (f) Connected component (g) without additional processing (h) with pre-post processing

Algorithm	Precision	Recall	f-measure
Proposed System	0.66	0.75	0.70
Epshtein [5]	0.73	0.60	0.66
Becker [3]	0.62	0.67	0.62
Chen and Yuille [4]	0.60	0.60	0.58
Zhu [3]	0.33	0.40	0.33
Kim [18]	0.22	0.28	0.22
Ezaki [19]	0.18	0.36	0.22
Ashida [2]	0.55	0.46	0.50
HWDavid [2]	0.44	0.46	0.45
Wolf [20]	0.3	0.44	0.35

Table I
COMPARISON OF PERFORMANCE OF OTHER TEXT DETECTION ALGORITHMS. THESE RESULTS ARE FROM ICDAR 2003, ICDAR 2005 COMPETITIONS AND EPSHTEIN'S PAPER.

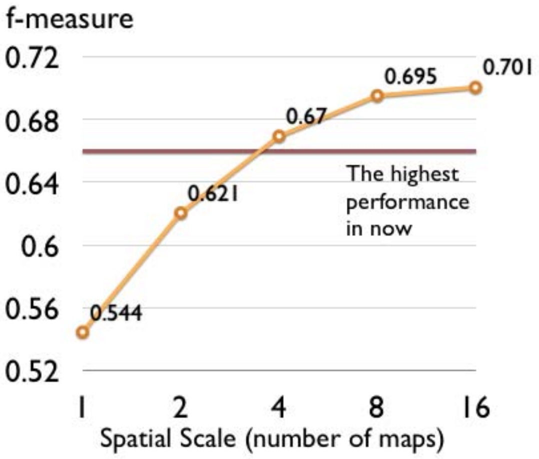
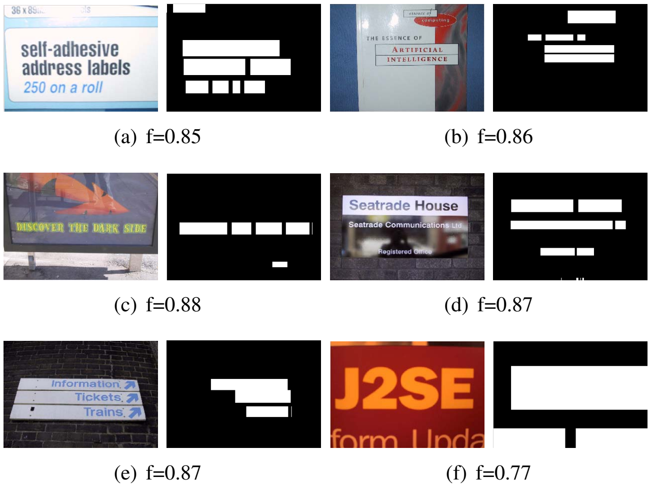


Figure 8. Performance of our algorithm as a function of the number of distinct linear scales.


self-adhesive address labels 250 on a roll
THE ESSENCE OF ARTIFICIAL INTELLIGENCE
DISCOVER THE DARK SIDE
Seatrade House
Seatrade Communications Ltd
Registered Office
Information
Tickets
Trains
J2SE
form Upda

(a) f=0.85
(b) f=0.86
(c) f=0.88
(d) f=0.87
(e) f=0.87
(f) f=0.77

Figure 9. Examples of successfully recognized text regions

weak performance, using the combination of features via AdaBoost results in an overall strong performance.

To find the trade-off between multiple maps with vary spatial scales, we evaluated the algorithm using 1, 2, 4, 8 or 16 spatial scale maps. For the last case, the largest scale is 1024×768 pixels and the smallest is 16 times smaller, *i.e.*, (64×48). As the number of spatial scale increases (Fig. 8), performance also increased.

We selected the algorithm with 16 spatial scales to compare against other algorithms running on the same image dataset. The results are displayed in Table 1. At the moment, our algorithm outperforms all other published text detection methods in terms of its f number.

VII. CONCLUSION AND REMARKS

We built a system using asymmetric Modest AdaBoost for detecting text in natural scenes. We extract 59 features within 64×32 pixels windows by applying 6 types of extraction strategies - X-Y derivatives, local energy of Gabor filter, statistical texture measure of image histogram, measurement of variance of wavelet coefficient, edge interval and analysis of connected components. We extract these features over 16 spatial scales to construct a CART as a weak classifier of the Modest AdaBoost. Counting components of X and Y gradients and morphological analysis enhanced the result of Modest AdaBoost.

The performance of our algorithm exceeds the performance of all published algorithms on the standard ICDAR benchmark [3], [5]. Yet, more remains to be done until we approach human performance. In particular, our algorithm performs sub-standard on low intensity text.

ACKNOWLEDGMENT

This research was supported by WCU(World Class University) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R31-10008).

References

[1] L. Breiman, *Classification and regression trees*. Chapman & Hall/CRC, 1984.

[2] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto *et al.*, "ICDAR 2003 robust reading competitions: entries, results, and future directions," *International Journal on Document Analysis and Recognition*, vol. 7, no. 2, pp. 105–122, 2005.

[3] S. Lucas, "ICDAR 2005 text locating competition results," in *Proceedings. Eighth International Conference on Document Analysis and Recognition, 2005.* IEEE, 2006, pp. 80–84.

[4] X. Chen and A. Yuille, "Detecting and reading text in natural scenes," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004.

[5] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2963–2970.

[6] W. Chan and G. Coghill, "Text analysis using local energy," *Pattern Recognition*, vol. 34, no. 12, pp. 2523–2532, 2001.

[7] B. Singh and B. Mazumdar, "Content Retrieval From X-RAY Images Using Color & Texture Features," *Methodology*, vol. 1, p. 6, 2010.

[8] S. Di Zenzo, "A note on the gradient of a multi-image," *Computer Vision, Graphics, and Image Processing*, vol. 33, no. 1, pp. 116–125, 1986.

[9] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *Machine Learning -International Workshop Then Conference-*. Citeseer, 1996, pp. 148–156.

[10] R. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine learning*, vol. 37, no. 3, pp. 297–336, 1999.

[11] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors)," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[12] A. Vezhnevets and V. Vezhnevets, "Modest AdaBoost-teaching AdaBoost to generalize better," *Graphicon-2005. Novosibirsk Akademgorodok, Russia*, 2005.

[13] P. Viola and M. Jones, "Fast and robust classification using asymmetric adaboost and a detector cascade," *Advances in Neural Information Processing Systems*, vol. 2, pp. 1311–1318, 2002.

[14] *MSU Graphics and Media Lab AdaBoost Toolbox*, Computer Vision Group, http://graphics.cs.msu.ru.

[15] R. Gonzalez, R. Woods, and S. Eddins, *Digital image processing using MATLAB*. Prentice Hall Upper Saddle River, NJ, 2004, vol. 624.

[16] *ICDAR 2003 database*, http://algoval.essex.ac.uk/icdar/Datasets.html.

[17] D. Chen, J. Odobez, and H. Bourlard, "Text detection and recognition in images and video frames," *Pattern Recognition*, vol. 37, no. 3, pp. 595–608, 2004.

[18] J. Kim, S. Park, and S. Kim, "Text Locating from Natural Scene Images Using Image Intensities," 2005.

[19] N. Ezaki, M. Bulacu, and L. Schomaker, "Text detection from natural scene images: towards a system for visually impaired persons," *Pattern Recognition*, vol. 2, pp. 683–686, 2004.

[20] C. Wolf, J. Jolion, and F. Chassaing, "Text localization, enhancement and binarization in multimedia documents," *Pattern Recognition*, vol. 2, p. 21037, 2002.