LEARNING BOTTOM-UP TEXT ATTENTION MAPS FOR TEXT DETECTION USING STROKE WIDTH TRANSFORM

S. Karthikeyan, Vignesh Jagadeesh and B. S. Manjunath

Department of Electrical and Computer Engineering, University of California Santa Barbara
{karthikeyan, vignesh, manj }@ece.ucsb.edu

ABSTRACT

Humans have a remarkable ability to quickly discern regions containing text from other noisy regions in images. The primary contribution of this paper is to learn a model to mimic this behavior and aid text detection algorithms. The proposed approach utilizes multiple low level visual features which signify visually salient regions and learns a model to eventually provide a text attention map which indicates potential text regions in images. In the next stage, a text detector using stroke width transform only focusses on these selective image regions achieving dual benefits of reduced computation time and better detection performance. Experimental results on the ICDAR 2003 text detection dataset demonstrate that the proposed method outperforms the baseline implementation of stroke width transform, and the generated text attention maps compare favorably with human fixation maps on text images.

Index Terms— Text Detection, Text Attention Maps, Stroke Width Transform, Visual attention

1. INTRODUCTION

Detecting text in natural scenes is an important problem for automatic navigation, robotics, mobile search and several other applications. Text detection in natural scenes is challenging as text is present in a wide variety of styles, fonts and shapes coupled with geometric distortions, varied lighting conditions and occlusions. Text detection techniques can be broadly classified into two categories: texture based approaches and connected component based approaches. Texture based approaches learn the texture differences between background and text regions. Image filtering techniques like Discrete Cosine transform [1] and Wavelet transforms [2] and Gabor filters [3] are commonly employed to represent the texture of text. These approaches typically use sliding windows and classify local image regions as text or non-text.

The second class of connected component (CC) based approaches are motivated by grouping pixels which exhibit similar text properties. The grouping happens at multiple levels : character, word and sentence. This is followed by a geometric filtering technique which removes false positives. Shivkumara et al. [4] proposed a CC approach in the Fourier-Laplace domain and geometric filtering using text straightness and edge density. Chen et al. [5] illustrated a CC based approach using Maximally Stable Extremal Regions (MSER). The popular Stroke Width Transform (SWT) [6] formulated by Ephstein et al. is also a CC based approach.

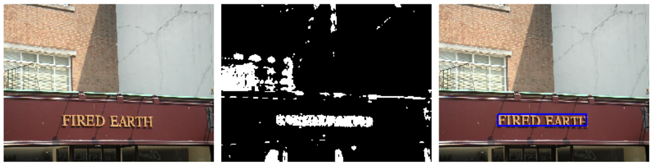

Fig. 1. Left to right: 1. Input image. 2. Text attention map derived using visual attention features. 3. The text detection output indicated by the blue rectangle. Best viewed in color.

SWT is an elegant approach to detect text. However, its performance heavily relies on the quality of edges which drive the transform computation. We propose a visual attention inspired solution to prune the search space of the SWT detector closer to text edges. In a free viewing task, human visual attention is heavily biased towards text regions [7] which have specific low level attention properties. Therefore, bottom up visual attention models which are designed to mimic human attention provide a useful prior for text detection. Given a set of training images, we compute several low level visual saliency maps, and train a classifier to understand both correctly and incorrectly labelled text and non-text regions provided by SWT detector. In a new test image, we use this classifier to produce a text attention map and SWT based text detection search is restricted to regions highlighted by this map improving both the speed and robustness of the detector. An example text detection obtained using our approach is shown in Fig.1.

In [8] Sun et al. also proposed a visual attention based text extraction approach based on Itti and Koch maps [9]. They used a predefined linear combination of the intensity, color and orientation channels to derive a map which filters false text blocks from potential character areas obtained by simple connected component analysis (CCA). This approach has several drawbacks. First, CCA based text detection is unreliable in the presence of noisy edges. Further, the weights for different features cannot be precomputed as in [8] when the number of bottom up features is large and finally [8] does not provide text attention map which prunes the detector search space. Our approach overcomes the limitations of [8] by enhancing the state of the art SWT detector. The primary contributions of our work are

- Learning a model to derive text attention maps for images from multiple bottom-up saliency features. These maps compare favorably to human fixations in text images.
- Utilizing the learnt text attention map to improve the speed and accuracy of the stroke width transform algorithm.

2. BACKGROUND: STROKE WIDTH TRANSFORM

The proposed work aims to improve Stroke Width Transform (SWT) algorithm. SWT is a CC based approach with four stages, stroke

This work was supported by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. Our other funding sources are N00014-12-1-0503, NSF III-0808772 and OIA-0941717.

width computation, character level grouping, geometric filtering and text line grouping. These stages are briefly described below.

Stroke Width Computation: Given an image, the edge map is computed using Canny edge detector. The gradient map is also obtained. From every edge pixel, rays are shot in the direction of the gradient until it encounters another edge pixel with an opposing gradient which is in the interval $[+\frac{\pi}{6}, -\frac{\pi}{6}]$ from the original gradient direction. If this condition is satisfied, pixels traced in this process potentially belong to the cross section of a stroke and are labelled as stroke pixels with width value equal to the euclidean distance between the two edges. If an opposing gradient is not encountered, the ray is discarded or no stroke value is assigned to the pixels traced in that process. However, this approach fails in the intersection of multiple strokes like the junction present in "T" as opposing gradient is absent in edges belonging to the junction. To fix this problem, a second iteration is performed along the edge pixels. Here, the discarded pixels are marked as strokes if more than a significant portion of these pixels have non-zero stroke width value from the first iteration. Finally, we obtain a map with potential strokes. To detect both bright and dark strokes, this algorithm is executed twice, in both the positive and negative gradient direction.

Character level grouping: In this stage similar strokes widths are grouped into characters using a modified connected component algorithm. This algorithm ensures grouping of two neighboring pixels if their stroke width ratio is in the range $[3, \frac{1}{3}]$.

Geometric filtering: Detected character regions which do not satisfy certain geometric properties related to aspect ratio, median stroke width and size of the connected components are discarded.

Text line grouping: Characters which have similar stroke widths, letter widths, height and spaces between letters and words are grouped to obtain text lines. A text line must have minimum three characters to suppress false detections.

For a detailed version of this algorithm we refer the reader to [6]. A visual example describing the steps in SWT is shown in Fig.2. SWT code is not publicly available and we implemented our version of the algorithm for this paper.

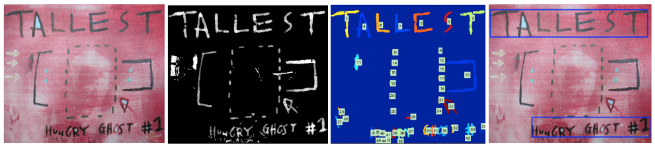

Fig. 2. Left to right: 1. The input image. 2. Stroke Width Transform Image. 3. Connected components and geometric filtering. 4. Final Detections (blue boxes). Best viewed in color.

3. THE PROPOSED APPROACH TO LEARN TEXT ATTENTION MAPS

The performance of SWT significantly depends on the quality of edges extracted from images. Typically, highly textured edges from trees, brick walls and other natural structures reduce the precision of SWT detector as it is prone to false positive detections in those regions. To overcome this problem, we develop an edge subset selection procedure which reliably detects *text edges*. Given a set of edges $\mathcal{E}$ in an image, we want to select a subset of edges $\mathcal{E}'$ which improves the SWT detector. Mathematically we want to obtain

$$\arg\max_{\mathcal{E}'} q_{SWT}^{\mathcal{E}'} \quad \text{S.T} \quad \mathcal{E}' \subseteq \mathcal{E} \tag{1}$$

where $q_{SWT}^{\mathcal{E}}$ is a quality measure of SWT detector using edges $\mathcal{E}$.

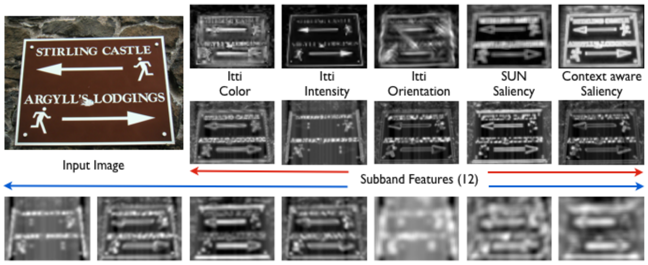

Fig. 3. Example of visual attention features computed in an image.

3.1. Learning

The best $\mathcal{E}'$ would correspond to a subset of edges which belong to text. As humans are adept at text detection, biologically motivated low level visual attention features which mimic human attention provide a useful prior for text boundary detection. Therefore, to approximate (1), we propose a learning based algorithm which estimates a mapping from these multiple low level saliency maps to text regions in an image for removing distracting edges.

3.2. Features

The following low level features are used in our algorithm:

Itti and Koch Saliency map: This early saliency model [9] is motivated by linear filtering and center surround operations and biologically motivated normalization provides intensity, color and contrast channels which we use in our model. This approach was primarily motivated for rapid analysis of visual scenes.

Context Aware Saliency Map: This approach [10] builds a mathematical model to the principles of human visual attention supported by psychological evidence which includes local global scale saliency, multi-scale saliency enhancement, immediate context inclusion, center prior and high level factors. This approach extracts salient objects together with parts of the discourse that surrounds them that can shed light on the meaning of the image.

Steerable pyramid features: The local energy of steerable pyramid filters [11] are correlated to visual attention. We use the features extracted from the pyramid subbands in four orientations and three scales similar to [12]. This combination provides 12 attention maps.

SUN Saliency map: Saliency Using Natural statistics [13] provides a map utilizing top-down and bottom-up information. This approach uses self information of visual features and pointwise mutual information between features and target during target search process.

We examined the utility of other saliency maps [12, 14, 15, 16, 17] which are effective in predicting human eye movements in natural images, however their text specificity was not suitable for our model, primarily attributed to the center bias prominent in these saliency maps. In total we have 17 attention maps and an example of the different extracted features are shown in Fig.3.

Given an image, we want to learn a binary map which highlights image regions which have high probability of text using features signifying visual attention. This map is called a text attention map, obtained by training a classifier to understand a mapping from attention features to text regions in images. Given a training set, we use SWT to extract character regions in all the images. Using the ground truth labels, we obtain sufficient true and false positive character regions.

A subset of pixels from true positive character regions are selected for training the text class. We also note that non-text class consists of equal number of pixels from false positives and true negatives. This procedure ensures the training set for non-text class consists of sufficient examples where SWT usually provides false

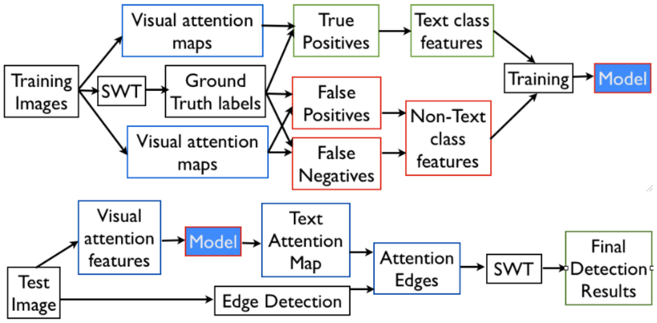

Fig. 4. Block diagram of the training(top) and test (bottom)modules of the visual attention based learning paradigm.

positives enabling the text attention map to correct SWT mistakes. Next, we learn a model to predict these text and non-text regions using visual attention maps at these selected pixel locations. Given a new image, this model (classifier) generates a corresponding text attention map by classifying every pixel as text or non-text and SWT based text detector only concentrates on regions classified as text. It offers dual benefits of lower computation time and higher precision. Further, the edges contained in these text attention maps approximate the edge subset selection problem (1). A block diagram of our framework is illustrated in Fig.4.

4. EXPERIMENTS AND RESULTS

We perform two separate experiments to validate the effectiveness of the derived text attention maps. In the first part we compare the text attention maps to eye fixation data in the MIT eye tracking dataset [12]. In the second experiment, we aim to improve the detection performance of stroke width transform algorithm.

4.1. Dataset and Setup

The ICDAR 2003 [18] text detection dataset is used to evaluate our algorithm. The dataset consists of 258 training images and 251 test images with challenging text present in various fonts, sizes, backgrounds, transparency, non-planar surfaces and reflections with word level ground truth annotations. As SWT is originally designed to capture text line groups, words in a sentence are combined to obtain text line level annotations for training and testing. During training, we run SWT and obtain true and false positive character regions. Next, all the training maps are resized such that the largest dimension consists of 200 pixels while maintaining the original aspect ratio. From every resized map, we randomly sample 12% of true positive locations for the text class and 7.4% of false positives and 0.3% of true negatives for non-text class. This gives about 40000 training samples per class (equal false positives and true negatives for non-text class). After training a model according to Sec. 3, for every test image (after resizing it in the same manner) we classify each pixel and obtain a text attention map by thresholding every pixel whose posterior probability of belonging to text class>0.35. This conservative threshold ensures most of the text regions are preserved in the map allowing some false non-text regions too. In the following stage, the SWT algorithm operates only on these regions for text detection. In practice we obtain a connected component canny edge map and every connected component which has more than 80% attention edges is selected for SWT based text detection. Algorithms 1 and 2 provide a step-by-step rundown of our training and test setup.

4.2. Comparison to Human Fixations

The proposed approach to obtain text attention maps was motivated to mimic the manner in which humans viewed text images. To test that theory, we collected a set of text images from MIT eye tracking dataset [12] and compared the text attention map generated by our algorithm to the gaussian smoothed human fixation map. Fig. 5 illustrates a few example images with their corresponding human and text attention map. We notice that our text attention maps significantly correlate well with human attention maps for the specific class of text images.

```
Data: Input Images $\{\mathcal{I}_i\}$ and binary ground truth labels $\{\mathcal{L}_i\}$,
      $i \in [1, N]$
Result: Classifier Model $\mathcal{C}$
initialization $tp$=0.12, $fp$=0.074, $fn$=0.003;
for $i$=$1 \rightarrow N$ do
    $\mathcal{F}_i$ = features($\mathcal{I}_i$);
    $\mathcal{SW}_i$ = Stroke Width Image($\mathcal{I}_i$);
    $\mathcal{T}_i$ = Binary Mask($\mathcal{SW}_i$): Binary mask of character regions;
    $\mathcal{TP}_i$=$\mathcal{F}_i(\mathcal{T}_i \bigodot \mathcal{L}_i)$: True Positives;
    $\mathcal{FP}_i$=$\mathcal{F}_i(\mathcal{T}_i \bigodot (1-\mathcal{L}_i))$: False Positives;
    $\mathcal{TN}_i$=$\mathcal{F}_i((1-\mathcal{T}_i) \bigodot (1-\mathcal{L}_i))$: True Negatives;
    $\mathcal{TP}_i^{sub}$ = Rand. Subset($\mathcal{TP}_i$) S.T $|\mathcal{TP}_i^{sub}| = \lfloor(|\mathcal{TP}_i|tp)\rfloor$;
    $\mathcal{FP}_i^{sub}$ = Rand. Subset($\mathcal{FP}_i$) S.T $|\mathcal{FP}_i^{sub}| = \lfloor(|\mathcal{FP}_i|fp)\rfloor$;
    $\mathcal{TN}_i^{sub}$ = Rand. Subset($\mathcal{TN}_i$) S.T
    $|\mathcal{TN}_i^{sub}| = \lfloor(|\mathcal{TN}_i|fn)\rfloor$;
end
train$^+$ = $\bigcup_i \mathcal{TP}_i^{sub}$;
train$^-$ = $(\bigcup_i \mathcal{FP}_i^{sub}) \cup (\bigcup_i \mathcal{TN}_i^{sub})$;
$\mathcal{C}$ = Classifier(train$^+$,train$^-$);
```

Algorithm 1: Training algorithm

```
Data: Test Image $\mathcal{I}$, Edgemap $\mathcal{E}$, Classifier $\mathcal{C}$, Connected
      Component Edges $\mathbf{C}^E$
Result: Text Attention Map $\mathcal{A}$, Attention Edges $\mathcal{E}'$ Detections $\mathcal{D}$
initialization $\mathcal{E}' = \emptyset$;
$\mathcal{F}$ = features($\mathcal{I}$); posterior = $\mathcal{C}(\mathcal{F})$;
for $i,j \in [row,col]$ do
    $\mathcal{A}(i,j) = \begin{cases} 1 & \text{posterior} > 0.35 \\ 0 & \text{else} \end{cases}$
end
for each $c \in \mathbf{C}^E$ do
    if $\frac{\sum_{p \in P} c(p)\mathcal{A}(p)}{|c|} > 0.8$ then
        $\mathcal{E}' = \mathcal{E}' \cup c$
    end
end
$\mathcal{D}$ = SWT($\mathcal{E}'$) : Stroke Width Transform on $\mathcal{E}'$
```

Algorithm 2: Testing algorithm

4.3. Text Detection Results

The output of text detection algorithm are a set of rectangles denoting text lines. These rectangles are matched to the ground truth rectangles representing text lines. A match score m, between two rectangles is determined as the intersection area divided by the union area. This quantity is 1 for identical rectangles and 0 for non-overlapping ones. For a given rectangle t the best matching rectangle m_b in a set of rectangles $\mathcal{T}$ is defined by $m_b(t, \mathcal{T}) = \max\{m(t,t')|t' \in \mathcal{T}\}$. This leads us to the definintions of Precision and Recall as Precision $= \frac{\sum_{t_e \in \mathcal{E}} m_b(t_e,\mathcal{G})}{|\mathcal{E}|}$ and Recall $= \frac{\sum_{t_g \in \mathcal{G}} m_b(t_g,\mathcal{E})}{|\mathcal{G}|}$. Here, $\mathcal{G}$ and $\mathcal{E}$ are the sets of ground truth and estimated rectangles respectively. The precision and recall are combined to a single quantity called f measure which is defined as $f = \frac{1}{\frac{\alpha}{\text{Precision}} + \frac{1-\alpha}{\text{Recall}}}$. Typically α is set to 0.5.

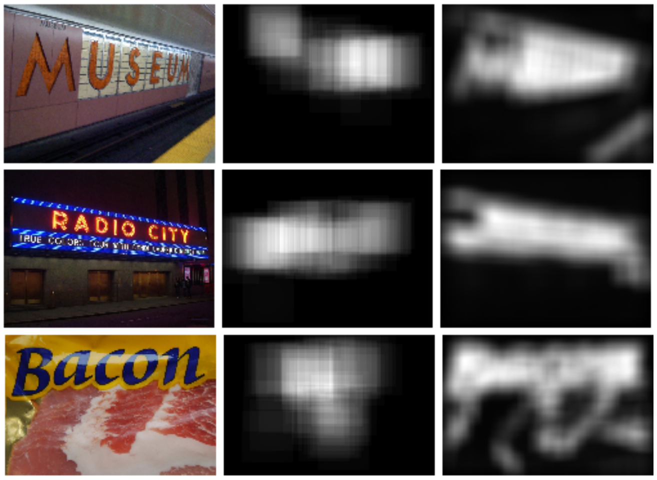

Fig. 5. Left column shows the input image, center column corresponds to human fixation map and the right column illustrates the proposed text attention map. The text attention maps are similar to human fixation map on text centric images. Note that eye fixations only includes foveal or central vision and peripheral vision is not captured. Therefore, as row 1 and 3 only have a single word, eye tracking results are biased towards the center of the word and therefore does not entirely overlap with our text attention map. Moreover, the text attention maps reliably localize the text regions.

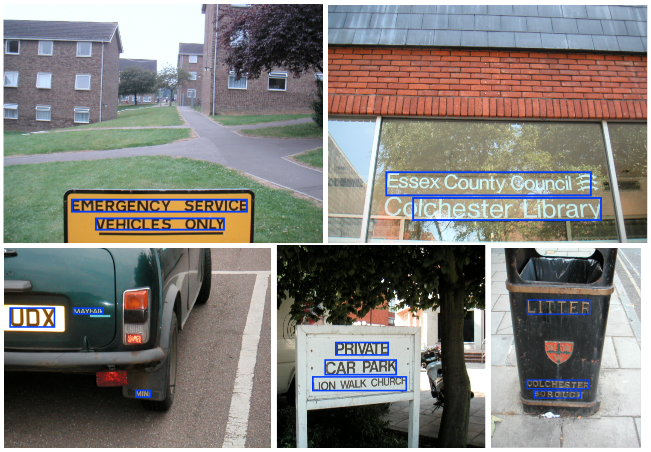

Fig. 6. Example detections (blue boxes) in images from ICDAR dataset. Best viewed in color.

	Precision	Recall	f Measure	Median Edges
SWT	0.613	0.721	0.664	12723
Our Method	0.720	0.727	0.724	19745

Table 1. Comparison of the performance of our algorithm and SWT

First, in the training phase we used three classifiers: SVM with Radial Basis Function(RBF) Kernel [19], Lib-linear SVM [20] and Linear Discriminant Analysis based classifier (LDA) [21]. SVM with RBF kernel was able to learn a better model to predict text regions on a validation set, than the linear classifiers as it obtained 86.3% accuracy compared to 78.3% and 77.1% by Lib-linear SVM and LDA respectively. This validation is a significant step as it provides evidence that bottom up visual attention based features can be used to understand text regions in images. Further, in the test stage, SVM with RBF kernel is used to compare our approach to SWT.

In the test phase, the proposed approach using SVM+RBF kernel obtains significantly better precision than baseline SWT and therefore f measure of our algorithm outperforms baseline SWT by 9.04% as indicated in Table 1. The text attention map is also able to remove a significant portion of false positive edges (about 55% from Table 1) and our text attention coupled with SWT is 30% faster than baseline SWT. Fig.6 shows some example detections obtained from our algorithm. The proposed approach is able to reject textured regions such as trees and bricks and is able to reliably detect text even in the presence of reflection and background clutter. Fig.7 highlights some visual examples where the proposed approach provides better detection results than SWT. The derived attention maps for these images indicate that edges corresponding to low contrast background regions (especially bricks) are ignored by the text attention maps leading to improved detection accuracy. Finally, Fig.8 shows an example where our approach fails to detect the text region in the image. The attention map in Fig.8 ignores the text region as it blends in with the surrounding background which caused the missed detection.

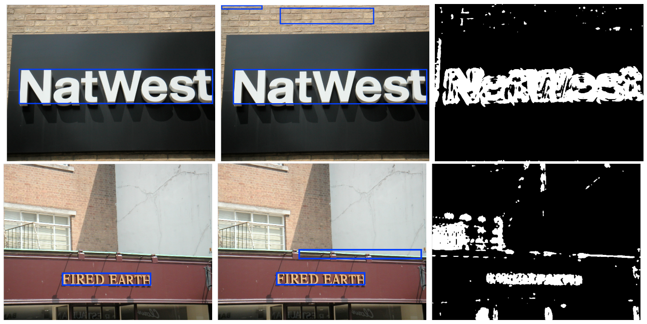

Fig. 7. Illustrates two example scenarios where our algorithm (left) outperforms SWT (center). The text attention maps (right) clearly ignores regions where SWT detects false positives. The detections are shown in blue rectangles. Best viewed in color.

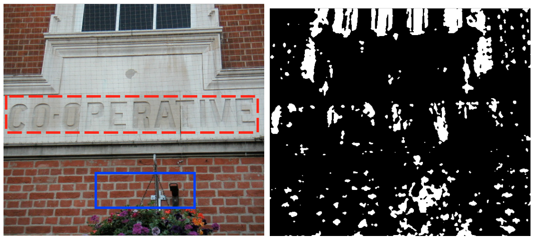

Fig. 8. An example image (left) where the proposed algorithm fails and the corresponding attention map (right). In this image the background is very similar to text region, hence, the text attention map fails to localize the text region. The missed detections are shown in red rectangles and the false positive detections in blue rectangles. Best viewed in color.

5. CONCLUSION

We have proposed a novel learning based framework to obtain text attention maps for images. These text attention maps prune the search space for SWT based detection algorithm. This approach significantly improves the precision of the SWT detector and also reduces the computation time. However, in regions where the text blends with the background our approach fails to detect the text. In addition, our attention maps resemble human attention maps in text images without multiple distractor elements. In the future we want to explore the possibility of adapting a learnt visual attention model to provide text attention maps instead of learning it ab initio.

6. REFERENCES

[1] Y. Zhong, H. Zhang, and A.K. Jain, “Automatic caption localization in compressed video,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 4, pp. 385–392, 2000.

[2] Q. Ye, Q. Huang, W. Gao, and D. Zhao, “Fast and robust text detection in images and video frames,” *Image and Vision Computing*, vol. 23, no. 6, pp. 565–576, 2005.

[3] F. Imamoglu Konuskan, “Visual saliency and biological inspired text detection,” *Masters Thesis, Technical University Munich & California Institute of Technology*, 2008.

[4] P. Shivakumara, T.Q. Phan, and C.L. Tan, “A laplacian approach to multi-oriented text detection in video,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 2, pp. 412–419, 2011.

[5] H. Chen, S.S. Tsai, G. Schroth, D.M. Chen, R. Grzeszczuk, and B. Girod, “Robust text detection in natural images with edge-enhanced maximally stable extremal regions,” in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 2609–2612.

[6] 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.

[7] M. Cerf, E.P. Frady, and C. Koch, “Faces and text attract gaze independent of the task: Experimental data and computer model,” *Journal of vision*, vol. 9, no. 12, 2009.

[8] Q. Sun, Y. Lu, and S. Sun, “A visual attention based approach to text extraction,” in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 3991–3995.

[9] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 11, pp. 1254–1259, 1998.

[10] S. Goferman, L. Zelnik-Manor, and A. Tal, “Context-aware saliency detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 10, pp. 1915–1926, 2012.

[11] E.P. Simoncelli and W.T. Freeman, “The steerable pyramid: A flexible architecture for multi-scale derivative computation,” in *Image Processing, 1995. Proceedings., International Conference on*. IEEE, 1995, vol. 3, pp. 444–447.

[12] T. Judd, K. Ehinger, F. Durand, and A. Torralba, “Learning to predict where humans look,” in *Computer Vision, 2009 IEEE 12th international conference on*. IEEE, 2009, pp. 2106–2113.

[13] L. Zhang, M.H. Tong, T.K. Marks, H. Shan, and G.W. Cottrell, “Sun: A bayesian framework for saliency using natural statistics,” *Journal of Vision*, vol. 8, no. 7, 2008.

[14] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[15] X. Hou, J. Harel, and C. Koch, “Image signature: Highlighting sparse salient regions,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 1, pp. 194–201, 2012.

[16] J. Harel, C. Koch, and P. Perona, “Graph-based visual saliency,” *Neural Information Processing Systems*, 2007.

[17] S. Karthikeyan, V. Jagadeesh, and BS. Manjunath, “Learning top-down scene context for visual attention modeling in natural images,” *IEEE International Conference on Image Processing*, 2013.

[18] S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, “Icdar 2003 robust reading competitions,” in *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, 2003, vol. 2, pp. 682–687.

[19] C.C. Chang and C.J. Lin, “Libsvm: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 27, 2011.

[20] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin, “Liblinear: A library for large linear classification,” *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[21] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 711–720, 1997.