

Scene Text Recognition in the Wild

Chen-Yu Lee

University of California, San Diego

chl260@ucsd.edu

Abstract

This paper demonstrates an algorithm to tackle the problem of reading text in uncontrolled “natural” photos. In contrast to the traditional OCR problem, for which the focus was on scanned pages, scene text presents a number of challenges more commonly associated with general object recognition, including different viewpoints, sizes/scale, locations, fronts, and styles (neon, graffiti). There are two main contributions of this work: The first is the use of multi-ratio of width and height of sliding window based character detector. The second is how to extract more representative information in the training step for the character classifier. We show a significant improvement in accuracy on Street View Text dataset by 10% and provide a novel approach to achieve a better character detector.

1. Introduction

For most people, a visual display of information is the fastest and most direct way to receive external information, through activities such as billboards and neon signs. However, for the visually impaired, it cannot be conveyed via the visual way to obtain information. In this paper, scene text recognition technique can be used in natural environments, so that the visually impaired person can access to the environmental information in texts.

Scene text recognition could also help improve map services. House number and store name recognition helps improve address geocoding. In most countries, very few addresses have associated geographic coordinates. The geocodes of the majority of addresses is therefore usually computed by interpolating the geocodes of neighboring addresses. Such interpolation introduces errors which may be significant, and may result in poor user experience. With the vehicle’s location, the house number and the store name, a better map service of the building of interest can be computed in a straightforward way.

Smith and Learned-Miller [11] show that the similarity among characters could be used to improve scene text recognition. They use bottom-up design by starting with



Figure 1. Examples of character detection and word recognition.

hand segmentations of each character in the form of a rectangular bounding box, then the similarity expert performs on all pairs of bounding boxes within one word to higher the confident score of boxes that are the same in both signal and label spaces. This approach required highly rectified candidate bounding boxes (segmented by hand in [11]) to obtain reliable similarity output.

Neumann and Matas [7, 8] use traditional pipelines for text recognition that their system first performs text localization by sequential selection from the set of Extremal Regions (ERs), and then use character segmentation approach to produce input for traditional Optical Character Recognition (OCR) engine. While this algorithm relies on stability of both text localization and segmentation procedure.

Netzer et al., [6] detect and read house numbers from street level photos using unsupervised feature learning method. They show a robust feature learning approach that does not need to use any existing image descriptors and achieve high accuracy on their benchmark. The experiment setting is similar to our goal while they conduct text recognition on total 10 classes (from 0 to 9) and need training dataset that contains around 73,000 digits.

Mishra et al., [5] propose a Conditional Random Field model to perform word recognition from candidate bound-

ing boxes of each character obtained in similar method as our work. They impose top-down cues obtained from a lexicon-based prior and the optimal word representation is obtained by minimizing the energy function corresponding to the random field model.

While progress has been made on cropped word recognition recently, the main challenge still lies on end-to-end word recognition on the full image due to a great amount of unpredictable false positive objects. Our work and experiment setting are based on [12] that we propose a more robust and reliable character detector to achieve a better scene text recognition on both cropped word and whole word recognition on full image.

2. Character Detection

Each step of character detection will be discussed in detail.

2.1. Character Detection with Random Fern

Like many object recognition approaches, the first step in our algorithm is to detect potential locations of characters in the input image. In order to detect characters in different fonts and view points, we perform multi-scale and multi-aspect ratio for each character via sliding window classification. However, there are 62 categories (26 lowercase, 26 uppercase, and 10 digits) in our problem, we need to choose a suitable classifier in order to handle this large amount of categories efficiently. Random Ferns [1, 9, 10] are an robust classifier to deal with multi-class problem. In our implementation, we use Histogram of Gradient (HOG) descriptor [2]

This sliding window detection method produces many possible locations with different scales and aspect ratios. We call them candidate bounding boxes. However, some of those candidate boxes are not useful for recognizing words in the next step. We eliminate those bounding boxes using the following method.

2.2. Candidate Pruning and NMS

English alphabet tends to have certain aspect ratios for each character. For example, character ‘I’ is usually thinner than ‘W’ and ‘l’ is thinner than ‘m’. We then adopt this aspect ratio heuristic to prune out those candidate bounding boxes with unreliable aspect ratios using normal distribution model:

$$AC(l_i) = \exp\left(\frac{-(\mu_{a_j} - a_i)^2}{2\sigma_{a_j}^2}\right)$$

where μ_{a_j} and σ_{a_j} are the mean and variance of the aspect ratio (computed from training data) for character j for a window l_i with aspect ratio a_i and $AC(l_i)$ is the aspect ratio confidence score. At testing time, all candidate bounding



Figure 2. Examples of truncated training data. Notice that the internal geometric information still preserved after eliminate information on both sides.

boxes for each character will be examined by this model, and then for those candidates with low aspect ratio confidence scores will be eliminated.

We then apply Non-Maximum Suppression (NMS) for each character to address the issue of multiple overlapping detection for each instance of a character. Notice that NMS is performed after aspect ratio pruning because wide candidates may contain other thin candidates while thin candidates are true locations and we do not want NMS first eliminate those true but thin candidates.

3. Word Recognition

The character detection step provides us with a large amount of potential candidate bounding boxes for each character. Our next step is to find words from those candidates and produce reliable word recognition output.

3.1. Problem setup

Let $x = (x_1, \dots, x_n)$ be the candidates bounding boxes and $s = (s_1, \dots, s_n)$ be the scores associated with them. We want to find the optimal configuration that minimizes the following cost function

$$\begin{aligned} x^* &= \operatorname{argmin} -s^T x + x^T A x \\ s.t. \quad & x_i(x_i - 1) = 0 \forall i = 1 \dots n \end{aligned}$$

Given the primal problem, the dual of this problem is derived as followed,

$$\begin{aligned} d^* &= \max -\frac{1}{2}(s + \lambda)^T (A + \operatorname{diag}(\lambda))^\dagger (s + \lambda) \\ s.t. \quad & A + \operatorname{diag}(\lambda) \succeq 0 \\ & s + \lambda \in \mathbb{R}(A + \operatorname{diag}(\lambda)) \end{aligned}$$

The semidefinite programming equivalent of this problem is expressed as followed,

$$\begin{aligned} \max_t \quad & \\ \text{s.t.} \quad & \begin{bmatrix} A + \text{diag}(\lambda) & -\frac{1}{4}(s + \lambda) \\ -\frac{1}{4}(s + \lambda) & -t \end{bmatrix} \succeq 0 \end{aligned}$$

The dual of the SDP is

$$\begin{aligned} d^* &= \text{argmin}_{tr}(AZ) - \frac{1}{2}s^T z \\ \text{s.t.} \quad & \text{diag}(Z) = \frac{1}{2}z \\ & \begin{bmatrix} Z & z \\ z^T & 1 \end{bmatrix} \succeq 0 \end{aligned}$$

We treat this SDP problem as a relaxation of the primal problem. After obtaining the solution for the SDP, we search across the values of the elements of z to minimize the original cost function.

3.2. Word Re-scoring and NMS

The final step is our algorithm is to perform word level re-scoring and NMS [12] in order to capture global information for each word output by pictorial structure.

4. Experiment

This section describes detailed evaluation of our method.

4.1. Dataset

We use the Street View Text (SVT) [12] dataset in our experiments. The SVT dataset contains images taken from Google Map Street View. The words in SVT images come from local business sign and have high degree of variability in appearance and resolution. There are total 647 cropped word images from the SVT testing set.

4.2. Character Detection

There are many descriptors can be used for character recognition, and we choose the HOG feature as our descriptor that it outperform the others in [3] because it preserves better geometric information. We densely compute HOG features with a cell size of 8×8 using 8 bins after resizing each image to 48×48 windows.

We then perform sliding window based character detection for multi-scale and multi-ratio of width and height for every location in the input image. The candidate bounding boxes obtained by Random Fern classifier are then pruned by normal distribution model with a probability less than 0.4. NMS is performed on the remaining bounding boxes to avoid wrong elimination. These two pruning steps are

Approach	Accuracy
Previous work [12]	0.56 (364 words)
Multi-scale + Multi-ratio	0.60 (390 words)
Multi-scale + Multi-ratio + TT	0.66 (427 words)

Table 1. Cropped Word Recognition Accuracy: A comparison of proposed method to PLEX in [12]. We improve the accuracy by 10% in SVT-WORD dataset.

simply but efficient to discard noisy candidate while still preserve most true positive candidates for the next word recognition step.

4.3. Cropped Word Recognition

We evaluate cropped word recognition on the SVT-WD dataset assuming a perfect text detector is available. Pictorial structure in [12] is used to recognize words using a lexicon of about 50 words for each image. Figure 4 shows the output by our system compared with the previous work in [12]. Table 1 shows the word recognition accuracy is improved using multi-scale and multi-aspect ratio approach, and the result could be further improved by training on truncated training (TT) data total 10% more.

4.4. Word Detection and Recognition on Full Image

Our main experiment of end-to-end scene text recognition is performing word detection and recognition on the full image. A bounding box is counted as a match if it overlaps a ground truth bounding box by more than 50% and the words match. Figure 3 shows the precision and recall rates of our system. Notice that F-score is improved using proposed character detector.

4.5. Results and Discussion

We evaluate our methods on the challenging SVT dataset, which contains words in different viewpoints, sizes/scale, locations, fronts, and styles with low resolution or low contract conditions. We observe that our proposed method outperform [3] in cropped word recognition by 10%.

We study the effect of multi-ratio of width and height. Original work in [12] uses fixed aspect ratio for sliding window approach. However, we observe that different characters tend to have different ratios between width and height. In order to leverage this feature, multi-ratio of width and height for sliding window is performed in our algorithm, and the word recognition accuracy is higher by 4% as in Table 1. The reason could be that our multi-ratio approach can better capture true character bounding boxes in natural image with high variability.

We also analyze the effect of truncated training data. Random Fern classifier is train on square training data that

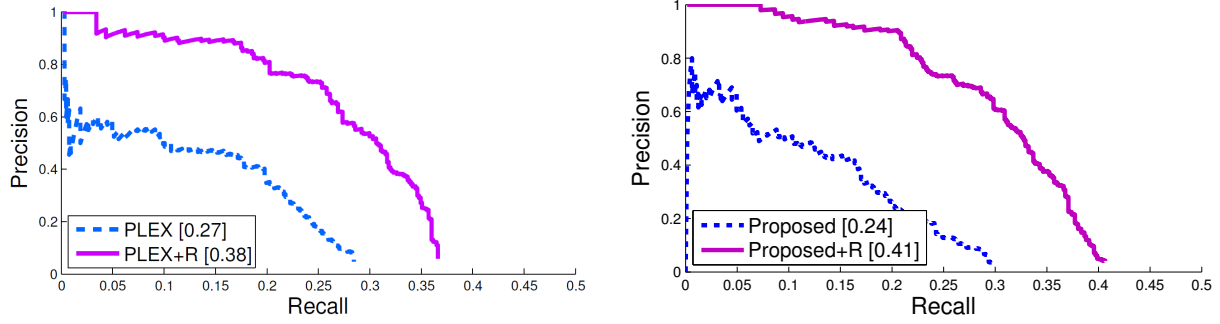


Figure 3. Precision and recall of end-to-end word detection and recognition methods on the SVT dataset. F-score are shown in brackets. Notice that the proposed character detector captures more precise and reliable characters and lead to better result on full image.



Figure 4. Experiment results on SVT-WORD dataset. Results from [12] are shown in the left column and our work are shown in the right column. Notice that our proposed algorithm works for words with tightly connected characters because our Random Fern classifier is train on truncated training data.

contain some noise on both sides in [12]. However, in many real street view cases, characters are tightly connected and it is difficult to contain all possible noise in training step. Therefore, we train our character detector on truncated training data by discarding 0.3 width information on both sides to capture the internal geometric information. The training error is higher than before by using truncated training data but surprisingly the word recognition accuracy is higher by 6% more and achieve total 10% higher (Table 1) in cropped word recognition accuracy than in [12]. The result shows that we only need the internal information to

capture characters in natural street view image and we can avoid the labor of collecting all possible combinations for the noise in both sides in training step. In the reminder of the paper we will refer to the truncated training procedure as “TT”

5. Conclusion

Our work establish a new approach to improve traditional sliding window based character detector and a novel way to select useful training data by truncating each character on both sides and further extract more representative

geometric information for character classifier. We show that we can outperform original work in [12] for both cropped word recognition and end-to-end scene text recognition on full images, and contribute to better text recognition in the natural images.

References

- [1] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *IEEE International Conference on Computer Vision*, 2007.
- [2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.
- [3] T. E. deCampos, B. R. Babu, and M. Varma. Character recognition in natural images. In *Proceedings of the International Conference on Computer Vision Theory and Applications*, Lisbon, Portugal, February 2009.
- [4] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. <http://people.cs.uchicago.edu/~pff/latent-release4/>.
- [5] A. Mishra, K. Alahari, and C. V. Jawahar. Top-down and bottom-up cues for scene text recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
- [6] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.
- [7] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *ACCV 2010: Proceedings of the 10th Asian Conference on Computer Vision*, 2010.
- [8] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
- [9] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [10] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [11] D. L. Smith, J. Feild, and E. Learned-Miller. Enforcing similarity constraints with integer programming for better scene text recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [12] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011.



Figure 5. Experiment results on SVT-WORD dataset. Results from [12] are shown in the left column and our work are shown in the right column. Notice that our proposed algorithm works for words with tightly connected characters because our Random Fern classifier is train on truncated training data