Scene Text Recognition in the Wild

Chen-Yu Lee
chl260@ucsd.edu

Phuc Xuan Nguyen
pxn002@ucsd.edu

Figure 1. Examples of character detection and word recognition.

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 novel character detector using more representative information in the training step and support vector machine based classifier. The second is a more general word recognition system by setting up the dual of a Quadratic Programming problem. We show a significant improvement in accuracy on Street View Text dataset by 10

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] showe that the similarity among characters could be used to improve scene text recognition. They use bottom-up design by starting with 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 tradition 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 rely 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 bounding 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 ene-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 SVM

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. Multi-class Support Vector Machine (SVM) [**?**] is 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 Re-scoring 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 re-score those candidate bounding boxes base on their 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 probability value. At testing time, all candidate bounding boxes for each character will be re-scored by multiplying the probability score of normal distribution. Confident score of all candidate bounding boxes with unreliable aspect ratios would be suppressed by the probability value.

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.

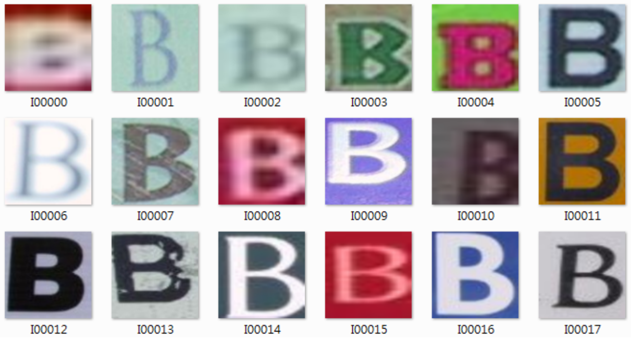


Figure 2. Examples of truncated training data. Notice that the internal geometric information still preserved after eliminate information on both sides.

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

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

where A is a cost matrix. The construction of the cost matrix is described in the next section. Given the primal problem, the dual of this problem is derived as followed,

$$\begin{array}{rcl} d^* & = & max - \frac{1}{2}(s+\lambda)^T (A + \text{diag}(\lambda))^\dagger (s+\lambda) \\ s.t. & & A + \text{diag}(\lambda) \succeq 0 \\ & & s + \lambda \in \Re(A + \text{diag}(\lambda)) \end{array}$$

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

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

The dual of the SDP is

Figure 3. Illustration of character detection and word recognition.

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

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 z as thesholds to minimize the original cost function.

3.2. Construction of cost matrices

We construct the cost matrix from three factors: collisions, bigram probability, and the color similarities. Mathematically, the cost matrix is defined as,

$$A = -\alpha B + \beta C + \gamma S$$

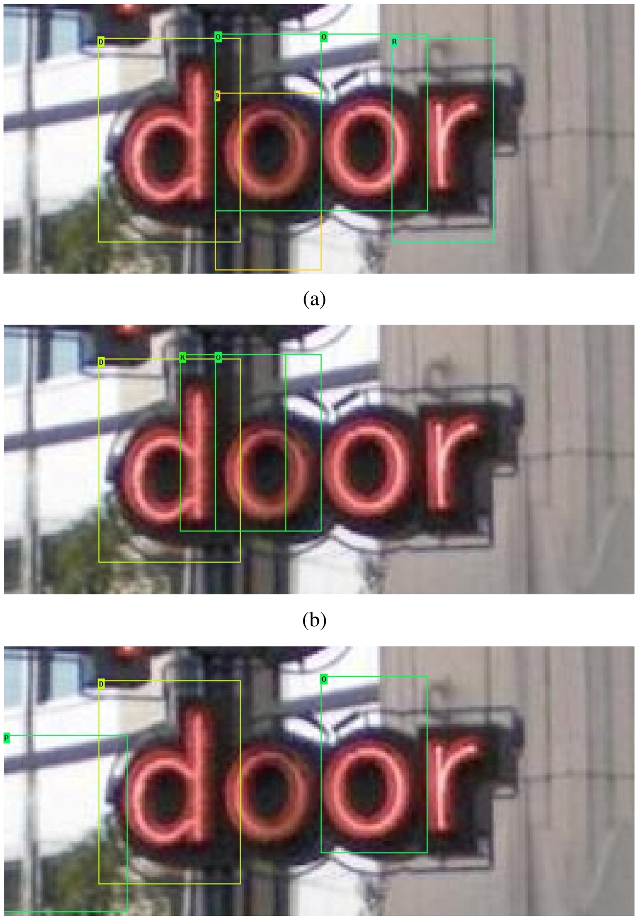
where B, C, and S matrices represent bigram probability, collision, and color similarity, while α, β, and γ are the weighting factors. Collision was factored in the cost matrix as the intuition provides that correct bounding boxes often have little overlaps with another. Figure ? provides an example where the collision distinction helps in recognition. Mathematically, the collision matrix is constructed as

$$C_{ij} = \frac{area_i \cap area_j}{area_i + area_j}$$

We construct a bigram model from our lexicon, consisted of 427 words. We want to dampen the bigram effects if the bounding boxes are far away from another. So we multiply the bigram score with the inverse of the Euclidean distance between the bounding box. The bigram matrix is mathematically defined as

$$B_{ij} = \begin{cases} \frac{P(l_i|l_j)}{d_{ij}} & x_i > x_j \\ 0 & otherwise \end{cases}$$

where $P(l_i|l_j)$ is the probability that the letter l_j is followed by l_i and d_{ij} is the Euclidean between the bounding boxes. The similarity matrix is constructed by computing the pairwise χ^2 distance the color histogram of each bounding box. We use 15 bins for each color channel.



(a)

(b)

(c)

Figure 4. Examples describing the effects of the factors in the cost matrix. Picture (a) shows an example where the bigram would break the tie between both plausible explanation for the words. The bigram would weight the chance of the letter 'D' followed by the letter 'O' high than the digit '9'. Example (b) shows the effects of minizing the overlaps would lead the algorithm to choose 'DO' over 'DK' or 'KO'. Example (c) shows the benifits of the similarity matrix. The bounding box containing the letter 'D' is closer in term of color histogram distance to the bounding box containing the letter 'O' than the letter 'P'.

4. Experiment

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 signs and have high degree of variability in appearance and resolution. There are total 647 cropped word images from the SVT testing set.

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.

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. This character detector is trained on ICDAR 2003 dataset, char74K, and our synthetic data using a one-againt-one SVM classifier with an RBF kernel, where the synthetic data contains about 10,000 images for 62 classes using 40 fonts and for each image we add some amount of Gaussian noise, and apply a random affine deformation.

We then perform sliding window based character detection for multi-scale and multi-aspect ratio for every location in the input image. The candidate bounding boxes obtained by SVM 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 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

Using the character detection and recognition described in previous section, we evaluate our method on the cropped region of the SVT-WD data set. We empirically find that $\alpha = .4$, $\beta = .4$, and $\gamma = .2$ give the best performance. We achieve the accuracy of 52.2% on the test set. While our result is worse than the original method, we did not rely much on the lexicon. Our algorithm only leverages from the bigram built from the provided lexicons. Even so, we train our bigram model on the whole lexicon of 427 words, not just 50 words as the original method describes. Due to the restriction on time, we are not able to fully search over our parameter and fully discover the weighting factors in the cost matrices. As another metric for our evaluation, we compute the Levenshtein's distance between the prediction and the ground truth. The average edit distance is 2.484. The average length of the words in the lexicon is 5.818.

Figure 5. Experiment results on SVT-WORD dataset.

4.4. 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 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 ene-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.