Text Recognition in Natural Images using Multiclass Hough Forests

Gökhan Yildirim, Radhakrishna Achanta and Sabine Süsstrunk
School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

Keywords: Text Detection and Recognition, Hough Forests, Feature Selection, Natural Images.

Abstract: Text detection and recognition in natural images are popular yet unsolved problems in computer vision. In this paper, we propose a technique that attempts to detect and recognize text in a unified manner by searching for words directly without reducing the image into text regions or individual characters. We present three contributions. First, we modify an object detection framework called Hough Forests (Gall et al., 2011) by introducing "Cross-Scale Binary Features" that compares the information between the same image patch at different scales. We use this modified technique to produce likelihood maps for every text character. Second, our word-formation cost function and computed likelihood maps are used to detect and recognize the text in natural images. We test our technique with the Street View House Numbers* (Netzer et al., 2011) and the ICDAR 2003† (Lucas et al., 2003) datasets. For the SVHN dataset, our algorithm outperforms recent methods and has comparable performance using fewer training samples. We also exceed the state-of-the-art word recognition performance for ICDAR 2003 dataset by 4%. Our final contribution is a realistic dataset generation code for text characters.

1 INTRODUCTION

Text detection and recognition are popular and challenging problems in the computer vision community. State-of-the-art Optical Character Recognition (OCR) softwares robustly convert scanned documents into text. However, they do not perform well on text in natural images (de Campos et al., 2009). In natural images, text regions might have complex backgrounds and in general have no structure. In order to overcome these issues, previous research has mostly addressed text detection and recognition separately.

Text detection describes methods that isolate text regions from background clutter. In order to separate text and background, various methods have been proposed that can be summarized under two categories: connected-component based and texture-based.

In connected-component based methods, the color or intensity of text is assumed to be consistent. In order to find these consistent areas, mathematical morphology operations (Ezaki et al., 2004) and maximally stable extremal regions (Neumann and Matas, 2011) can be used. Connected-component based methods perform well if the consistency assumptions are met. However, they might not work on images with complex background and/or non-standard font styles.

In texture-based methods, text and non-text regions are distinguished by classification, such as Support Vector Machines (SVM) (Kim et al., 2003) and adaptive boosting with Haar-like structures (Chen and Yuille, 2004). Texture-based methods are robust in the presence of image clutter. However, they still have difficulty in detecting small, low contrast, or blurred text.

Text recognition in natural images is challenging due to practical difficulties, such as non-uniform backgrounds, geometric transformations, and the variety of font types and sizes. In the literature, we find structural and statistical approaches for solving the text recognition problem. In (Netzer et al., 2011), features are selected by clustering using K-means and other unsupervised techniques, and the characters (digits) are classified with SVM. In (de Campos et al., 2009), individual characters that are cut from natural images are recognized using a variety of different features and classified with nearest neighbours, SVM and multiple kernel learning. In (Newell and Griffin,

*http://ufldl.stanford.edu/housenumbers

†http://algoval.essex.ac.uk/icdar/Datasets.html

This work was supported by the Swiss National Science Foundation under grant number 200021_143406 / 1

Yildirim G., Achanta R. and Süsstrunk S..
Text Recognition in Natural Images using Multiclass Hough Forests.
DOI: 10.5220/0004197407370741
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2013), pages 737-741
ISBN: 978-989-8565-47-1

2011), multi-scale histograms of oriented gradients followed by a nearest neighbor-classifier are used to recognize characters. In (Neumann and Matas, 2011), directional Gaussian filters are applied to generate features for a character; these features are then used to train an SVM for recognizing a character. In (Neumann and Matas, 2011; Wang et al., 2011; Mishra et al., 2012), in addition to individual character recognition, lexicon priors for English language are employed for robust word recognition. As the recognition operation is not perfect, in practice it is required to utilize other information, such as word formation, lingual statistics, and size limitations to correct for false positives and negatives and to form a word.

In this paper, we present three contributions. First, we replace binary features in multiclass Hough forests with cross-scale binary features and to achieve better recognition rates. Multiclass Hough forests show promising results in object detection and recognition in (Gall et al., 2011; Razavi et al., 2011) but they have not yet been employed for text detection and recognition. Second, we propose a word formation method, which is more suitable and accurate for likelihood maps. Using our features and word formation method, we exceed the best performance for the SVHN dataset and outperform state-of-the-art word recognition performance of the ICDAR 2003 dataset by 4%. Finally, we present a realistic dataset generator for text characters.

The rest of the paper is as follows: In Section 2, we summarize the multiclass Hough forests and introduce cross-scale binary features. In Section 3, we describe our word-formation cost function, which attempts to jointly solve detection and recognition problems. In Section 4, we introduce our synthetic dataset and the performance of cross-scale binary features. In Section 5, we present and discuss our results for text detection and recognition on different datasets. In Section 6, we summarize our algorithm and related observations.

2 HOUGH FORESTS

The generalized Hough transform (Ballard, 1981) is a technique for finding the position and parameters of arbitrary shaped objects using local information in a voting scheme. In this paper, we use multiclass Hough forests (Gall et al., 2011), which is an efficient method to compute the generalized Hough transform for multiple types of objects by adopting random forests.

A Hough forest consists of binary random trees. In a random tree, a randomly selected portion of image patches are recursively split into smaller subsets by using binary features as separation criteria. One contribution of our paper is a new feature set called "Cross-Scale Binary Features" (see Section 2.2), which gives better recognition performance than the binary features from (Gall et al., 2011). We use our features to train and test the Hough forests.

2.1 Binary Features

In (Gall et al., 2011), the binary features are the comparison results of two randomly selected pixels. These pixels can be selected from different representations (intensity, gradient, blurred etc.) of the same image. The general structure to represent a binary feature is as follows:

$$f = \begin{cases} 1, & \text{if } \quad P^l(\mathbf{x_1}) > P^l(\mathbf{x_2}) + \tau \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Here, f is the binary feature, P^l is image patch P in l^{th} representation type, $\mathbf{x_1}$ and $\mathbf{x_2}$ are image coordinates and τ is an offset value.

2.2 Cross-Scale Binary Features

In our work, we modify the structure in (1) by introducing cross-scale feature comparison. We generalize the feature set by allowing the comparison among two representations l_1 and l_2 as follows:

$$f = \begin{cases} 1, & \text{if } P^{l_1}(\mathbf{x_1}) > P^{l_2}(\mathbf{x_2}) + \tau \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

In our method, each representation corresponds to a blurred version of the image processed with averaging filters with different kernel sizes. Thus, every feature effectively compares the mean values of two randomly positioned rectangles with random dimensions. A representation of a cross-scale feature on an image patch is shown in Figure 1.

Cross-scale binary features defined in (2) are the superset of the features in (1). Cross-scale features help us to exploit potential local structural information and to compactly group similar patches of different objects. Performance comparison of binary and cross-scale binary features can be found in Section 4.

2.3 Training and Testing Hough Forests

In the training step, we use 24×24 pixel images and densely sample them with 8×8 pixel patches. We

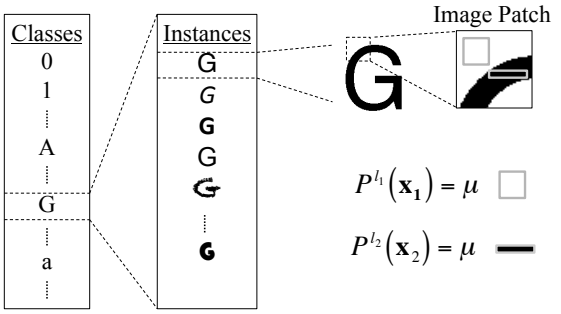


Figure 1: Representation of a cross-scale binary feature, μ represents the average pixel value for a given rectangular region.

then recursively split the image patches using a cross-scale binary feature that is selected from a pool of features by minimizing metric functions described in (Gall et al., 2011). Feature selection and splitting operations eventually lead to compact clusters of image patches. The separation procedure terminates when one of the two criteria is met: the maximum allowed tree depth is achieved or the total number of patches is less than a certain value. A leaf is then declared and the number of image patches for each class label and relative patch position is stored for testing as a "Hough vote".

In the testing step, multiclass Hough forests transform a test image into likelihood maps for each text character by computing and accumulating Hough votes of each test image patch. An example for the letter "a" is illustrated in Figure 2.

Figure 2: Generalized Hough transform for character "a" by using Hough forests (the image is taken from the ICDAR 2003 dataset).

For detection and recognition, we calculate Hough images for all 62 characters (digits, capital and small letters) and downsize images to achieve 10 different scales. We use integral images for efficient feature computation and distribute different scales to different processing cores.

3 WORD FORMATION

Letters can resemble each other either locally or globally. For example, the upper part of the letter "P" could be detected as a letter "D" within a smaller scale. Depending on the font style, the letter "W" can be confused with two successive "V"s or "U"s and vice versa. Therefore, instead of recognizing characters individually, we use a pictorial structure model (Felzenszwalb and Huttenlocher, 2005), which is similar to the one in (Wang et al., 2011) and produces lexicon priors with the help of an English word list[3]. These priors help us to reduce the search space and to robustly recognize the actual word. We recognize a word when the following function is locally minimized in the image:

$$w = \underset{n,\forall c_i}{\operatorname{argmin}} \Big(-\sum_{i=1}^{n} \mathbf{V}(L_i) + \sum_{i=1}^{n-1} \mathbf{C}(L_i, L_{i+1}) \Big) \qquad (3)$$

Here, w is the recognized word, L_i is the coordinate quadruplet (scale, character, 2-D position), $\mathbf{V}$ is the likelihood value of having that character at that scale and position, and $\mathbf{C}$ is the cost function for two adjacent letters. The cost function includes consistency terms for position, scale, capitalization, and lexicon priors. In order to find the most likely word hypothesis that minimizes the cost function, we perform a grid-like search on Hough maps. An example search is illustrated in Figure 3. The word search is initialized with a grid over all likelihood maps (red crosses). The word with minimum cost and part of the dictionary is the recognition output. The red circles depict $\mathbf{V}(L_1), \mathbf{V}(L_2)$, and $\mathbf{V}(L_3)$ and the red connections are $\mathbf{C}(L_1, L_2)$ and $\mathbf{C}(L_2, L_3)$.

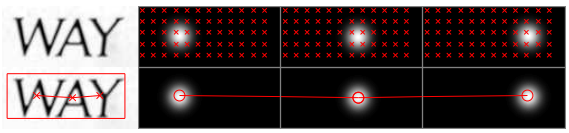


Figure 3: From left to right: word image, likelihood maps for "W", "A", and "Y". The first row shows the beginning of the word search and the second row shows the minimized path between likelihood maps.

4 EXPERIMENTS

Training Dataset. We create a dataset by using a subset of 200 computer fonts under random affine transformations[4]. In addition, we put other characters around the main character using the probabilities of occurrence in an English word (the word list of Section 3 is used). Finally, we blend these images with random non-text background to simulate a character

[3] http://www.mieliestronk.com/wordlist.html

[4] http://ivrg.epfl.ch/research/text_detection_recognition

in a natural scene.

We train our recognizer using over 1000 training images per class (10 trees in total with a maximum depth of 20). As the trees are independent from each other, they are trained in parallel. The same property is used when we compute the likelihood maps for text characters.

Feature Evaluation. We conduct five experiments for different number of features to compare the performance of cross-scale and normal binary features. If the number of features is equal to one, it means that there is no intelligent feature selection, thus totally random features and trees. The recognition results and error bars for each case in Figure 4 show that our features yield a better recognition performance. In addition, as the number of features increases, the differences between our features and binary features tend to increase. This is an expected result because our features are actually a superset of the binary features in (Gall et al., 2011).

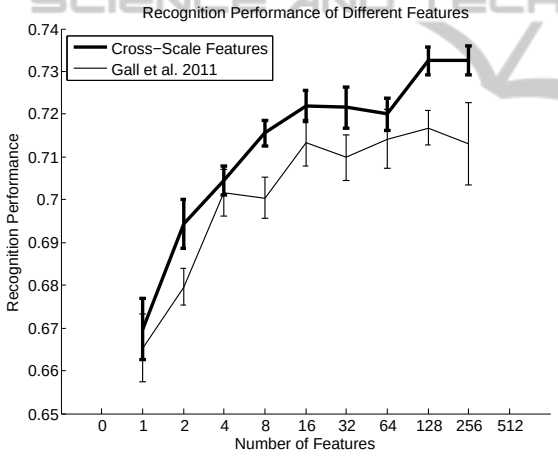


Figure 4: Recognition performance of two different types of features using our synthetically generated dataset.

5 RESULTS

Individual character recognition is done using the SVHN dataset, which consists of over 600'000 images of the ten digits with a 32×32 pixel size[5]. The performance is evaluated on the same test set as in (Netzer et al., 2011), which consists of over 26000 images of digits. As we can see from Figure 5, our technique has better performance than the K-means approach described in (Netzer et al., 2011). In addition, as the number of training samples decreases, we achieve the same recognition performance using approximately one-half of the training samples used for K-means. The cross-scale features that we use in training increase the distinction effectiveness of the Hough trees and create an accurate classifier with fewer training samples.

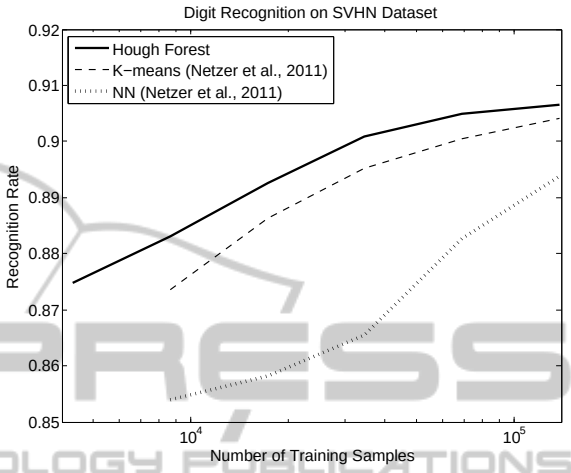


Figure 5: SHVN digit recognition dataset results.

We also test our algorithm on cropped words in the ICDAR 2003 database. As in (Wang et al., 2011) and (Mishra et al., 2012), we ignore words with non-alphanumeric characters and words that are shorter than three letters, giving us a total of 827 words. Note that the proper nouns and brand names that appear in the dataset are also in our search space. The recognition results are presented in Table 1.

Table 1: Cropped word recognition results (in %) from the ICDAR 2003 database.

Method	ICDAR2003	Time
Hough Forest	**85.7**	3 minutes
(Mishra et al., 2012)	81.78	-
(Wang et al., 2011)	76	15 seconds

Our framework is similar to that of (Mishra et al., 2012). The performance increase, however, is due to the exhaustive word search in the likelihood maps.

We also apply our algorithm to the full ICDAR 2003 images, some results are given in Figure 6. For the images in Figure 6(a),(b),(c), and (d), we are able to recognize the words correctly despite the non-standard font styles, distracting factors around the characters and/or complex backgrounds. The occluded word "Park" in Figure 6(b) is recognized correctly, because of the patch-based voting operation in Hough forests. In Figure 6(e), we incorrectly recognized the words "for" and "our" as "colour". This error might be due to the complex background, or to the

[5]We downsized the images to 24×24 to reduce the computational cost.

total word formation cost, which is lower for a combined word than individual words. In Figure 6(f), the word "Oxfam" was missed due to the resemblance of the letter "f" to the letter "t".

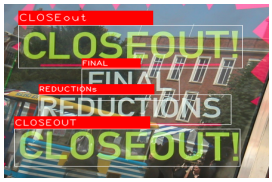
Computing Hough votes and searching for words can require significant computational power. However, due to the nature of Hough forests and the local word search operation in the image, the whole operation is highly parallelizable both in the training and testing stages.

6 CONCLUSIONS

We present a new method for text detection and recognition in natural images. We introduce cross-scale binary features and show that using these features improves the recognition performance. We train Hough forests using images generated by our realistic character generator code. We recognize the words in natural images using these features and our word-formation cost function. We test our algorithm on two available datasets. In individual character recognition, we show that our algorithm has a better recognition performance and can operate at same performance using fewer training samples. In cropped word recognition, we exceed the recognition performance of the most recent algorithm by 4%.

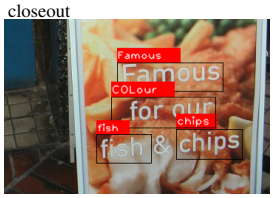
(a) australia

(b) sports, centre, wivenhoe, partk, conference, center, car, parks



(c) closeout, final, reductions, closeout

(d) yamaha



(e) famous, COLOUR, fist, chips

(f) OXFAM,bookshop

Figure 6: Some results of our algorithm on ICDAR 2003 dataset images (correctly and incorrectly recognized words are written in small and capital letters, respectively).

REFERENCES

Ballard, D. H. (1981). *Pattern Recognition*, 13(2):111–122.

Chen, X. and Yuille, A. (2004). Detecting and reading text in natural scenes. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 366–373.

de Campos, T. E., Babu, B. R., and Varma, M. (2009). Character recognition in natural images. In *Proc. of the International Conference on Computer Vision Theory and Applications*, pages 273–280.

Ezaki, N., Bulacu, M., and Schomaker, L. (2004). Text detection from natural scene images: towards a system for visually impaired persons. In *Proc. of the International Conference on Pattern Recognition*, volume 2, pages 683–686.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79.

Gall, J., Yao, A., Razavi, N., Gool, L. V., and Lempitsky, V. (2011). Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2188–2202.

Kim, K. I., Jung, K., and Kim, J. H. (2003). Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1631 – 1639.

Lucas, S., Panaretos, A., Sosa, L., Tang, A., Wong, S., and Young, R. (2003). ICDAR 2003 robust reading competitions. In *Proc. of the International Conference on Document Analysis and Recognition*, pages 682–687.

Mishra, A., Alahari, K., and Jawahar, C. V. (2012). Top-down and bottom-up cues for scene text recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2687–2694.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Neumann, L. and Matas, J. (2011). A method for text localization and recognition in real-world images. In *Proc. of the Asian Conference on Computer Vision*, volume 3, pages 770–783.

Newell, A. J. and Griffin, L. D. (2011). Multiscale histogram of oriented gradient descriptors for robust character recognition. In *Proc. of the International Conference on Document Analysis and Recognition*, pages 1085–1089.

Razavi, N., Gall, J., and Gool, L. J. V. (2011). Scalable multi-class object detection. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1505–1512.

Wang, K., Babenko, B., and Belongie, S. (2011). End-to-end scene text recognition. In *Proc. of the International Conference on Computer Vision*, pages 1457–1464.