SCENE TEXT RECOGNITION IN MULTIPLE FRAMES BASED ON TEXT TRACKING

Xuejian Rong[1], *Chucai Yi*[2], *Xiaodong Yang*[1] *and Yingli Tian*[1,2]

[1]The City College, [2]The Graduate Center, City University of New York
xrong000@citymail.cuny.edu, cyi@gc.cuny.edu, {xyang02, ytian}@ccny.cuny.edu

ABSTRACT

Text signage as visual indicators in natural scene plays an important role in navigation and notification in our daily life. Most previous methods of scene text extraction are developed from a single scene image. In this paper, we propose a multi-frame based scene text recognition method by tracking text regions in a video captured by a moving camera. The main contributions of this paper are as follows. First, we present a framework of scene text recognition in multiple frames based on feature representation of scene text character (STC) for character prediction and conditional random field (CRF) model for word configuration. Second, a feature representation of STC is employed from dense sampled SIFT descriptors and Fisher Vector. Third, we collect a dataset for text information extraction from natural scene videos. Our proposed multi-frame scene text recognition is more compatible with image/video-based mobile applications. The experimental results demonstrate that STC prediction and word configuration in multiple frames based on text tracking significantly improves the performance of scene text recognition.

Index Terms— Scene text recognition, text tracking, feature representation of scene text character, Fisher Vector, video dataset of scene text

1. INTRODUCTION

Text signage in natural scene serves as significant visual indicators for navigation and notification. With the popularity of mobile devices, many mobile applications incorporate the functionality of text information retrieval from camera-based images/videos of natural scenes. Especially, in the applications of assistive navigation, reading aid, vision-based geo-locating, as well as scene understanding, scene text information plays a very important role. However, accurate extracting scene text information remains a challenging task to be addressed. In most previous work, scene text extraction is performed on a single natural scene image. Given a scene image, two main steps are normally carried out to extract text information, which are scene text detection to obtain image regions containing text characters or text strings [4, 16, 17, 19, 20], and scene text recognition to obtain readable text codes from the detected text regions [8, 12, 13, 14, 18]. In this paper, we focus on the second step, i.e., scene text recognition. Due to the cluttered background and multiple text patterns, scene text recognition even from a detected text region is still a challenging problem. Some optical character recognition (OCR) systems have been released for real applications. But most of them are designed for the scanned documents with relatively clean background or segmented STCs. On the other hand, in a number of real-world applications of text information retrieval, the raw data captured from natural scene is in the form of video frames rather than a single scene image. This means that frame relationships are ignored in the traditional single image based scene text recognition methods. According to our literature review, there is no similar work of scene text recognition in video.

In this paper, we propose a novel method of scene text recognition from a video of natural scene. In our method, a text recognition method is combined with a tracking algorithm to improve the recognition accuracy. We carry out text detection on the first frame of a video, and generate an initial bounding box of each text region. Then we apply an object tracking algorithm to obtain the bounding boxes of the detected text regions in succeeding frames. Next, STC predictor and CRF model are applied to the tracking boxes of text regions to recognize text information. Fig. 1 illustrates a flow chart of our proposed framework. The main contributions of this paper are summarized as follows: (1) a novel framework of tracking-based scene text recognition in multiple frames; (2) a feature representation of STC using SIFT descriptor and Fisher Vector; (3) a self-collected dataset of videos containing text information from natural scenes. Our dataset will be released to the public through our research website.

In text recognition, we first solve the STC prediction, which is essentially a problem of multi-class classification. Many previous publications attempted to design discriminative feature representations of STC [12, 13, 18]. To rectify the error of STC prediction using lexical prior, CRF model is usually adopted to configure text word from predicted STCs [8, 12, 13, 14].

In our framework, multi-object tracking is combined with STC prediction and word configuration to improve the performance of scene text recognition. Multi-object tracking

aims to follow different targets in the entire video sequences. In general, multi-object tracking has to robustly maintain data association and synchronize the trajectory. This still remains an open issue due to the several reasons. First, the number of possible discrete object trajectories over time is quite large. Second, the estimation of dependencies between discrete trajectories is NP-complete [1]. Third, the inter-object occlusions result in evidence lost and appearance change. Text information in our framework is adopted to mitigate the above difficulties.

The rest of this paper is organized as follows. Section 2 introduces text detection and tracking used to extract text regions from multiple frames. Section 3 describes Fisher Vector based STC representation. Multiple frames based word configuration using CRF model is presented in Section 4. Section 5 demonstrates our collected dataset and experimental results to validate our proposed method. We conclude this paper in Section 6.

2. TEXT DETECTION AND TEXT TRACKING

This section describes the detailed algorithms we applied for scene text detection and tracking of text regions.

2.1. Scene Text Detection

In our framework, text detection is applied to extract text regions from the first frame of a video. The maximal stable extremal region (MSER) detector [9] is first used to extract connected components that are probably text characters in a scene image. The adjacent character grouping algorithm [17] is then applied to group connected components in similar size and linear alignment together to obtain text string candidates. A binary classifier is trained to distinguish the true positive text strings from those false alarms generated by non-text background. This binary classifier is learned by structural modeling of text string fragments using gradient distribution, stroke consistency, and edge density [20]. The true positive text strings are then further merged into text regions. Bounding boxes of the detected text regions serve as the input to scene text tracking.

Text string in a text region covers one or more words or phrases. Thus bounding boxes of words and phrases are used as the initial object locations in the following tracking process. Furthermore, since a text string consists of a group of text characters in the form of MSER connected components, the bounding box of each text character can be also obtained in the detection process. We use the relative positions of character members in a word to rectify the text tracking. The combination of MSER detector, adjacent character grouping, and structural classification obtains good performance of text detection.

2.2. Scene Text Tracking

To simultaneously track several STCs belonging to the same word, multi-object tracking is applied to scene text regions. In scene text scenario, we can avoid some challenges of multi-object tracking by the three constraints. First, we do not need to estimate the STC trajectories in the same word independently because we can instead estimate the trajectory of the whole word at first as a hint. Second, the STCs in the same word are well aligned and have relatively low dependencies with each other. Third, the relative locations of characters are stable. So the inter-object occlusions rarely happen as long as the whole word is clearly captured.

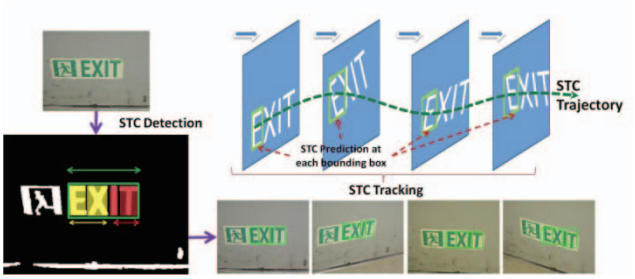

Fig. 1. Flowchart of the proposed framework. Initial text regions are extracted by grouping adjacent connected components in similar size and horizontal alignment in the MSER map. Each scene text character is independently tracked with the multi-object tracking method. A trajectory is then estimated by merging the tracked STC bounding boxes in each frame. The STC prediction scores along this STC trajectory is used to improve text recognition.

We adopt the tracking by detection method in our framework, i.e., each STC is treated as an independent object model that is detected and tracked in continuous multiple frames [10, 15]. An online tracking model in [5] is able to handle the variations of lighting and appearance. Compared to other tracking methods, e.g., the first-order Markov chain model which predicts object location in next frame from that in the current frame, the tracking by detection methods successfully solve the re-initialization problem even when a target has been accidentally lost in some frames, and the excessive model drift problem due to similar appearances of some STCs.

In each frame, MSER detector searches for the globally optimal hypothesis for the location of each STC. The detection output is then used as a constraint to optimize the trajectory search. Optimized trajectory estimation is then fed back to guide text detection in subsequent frames and reduce the effects of motion blur. The two processes are iteratively performed to track STC bounding boxes.

3. SCENE TEXT CHARACTER REPRESENTATION

3.1. Low-Level Feature Extraction

Given an image patch cropped from a bounding box of a video frame, if it contains a STC in the complete structure, STC prediction can be performed to obtain its category label and prediction score. Scene text recognition in detected image regions is based on the discriminative feature representation of STC. Scene text recognition involves 62

STC categories, i.e., 10 digits and 26 English letters in both upper and lower cases. For the multi-classification problem among the 62 categories, we present a Fisher Vector based STC feature representation.

In an image patch containing an STC, dense sampling and SIFT descriptor is applied to extract the low-level features. The image patch is first resized into a square with the side length S, which is a power of 2 and no smaller than 64. The gradient magnitude and orientation are computed at each pixel in this image patch. We then divide the image patch into $5 \times 5 = 25$ overlapping blocks. Each block has the size $(S/2) \times (S/2)$, and we slide it from top-left to horizontal and vertical directions. The stride distance between the centers of two neighboring blocks is $S/8$. Each block generates a 128-dimensional SIFT descriptor. The next step is to aggregate the SIFT descriptors from an image patch into a discriminative vector representation. While several coding and pooling schemes based on the bag-of-words (BOW) model [7] can be applied in this step, we employ the Fisher Vector as the feature representation. Temporal features are not suitable for scene text recognition, because text always appears as a static object, unlike human actions in temporal domain. In addition the extraction of temporal features will lower the efficiency of the whole framework.

3.2. Fisher Vector Representation

To generate more discriminative feature representation, we employ the Fisher Vector to represent each STC. Fisher Vector provides a feature aggregation scheme based on the Fisher kernel which takes the advantage of both generative and discriminative models. Fisher Vector describes each feature descriptor using the deviation with respect to the parameters of a generative model.

Fisher Vector employs the Gaussian mixture model (GMM) as the generative model $U_\lambda(x) = \sum_{k=1}^{K} \pi_k u_k(x)$, and u_k is the kth Gaussian component:

$$u_k(x) = \frac{1}{2\pi^{\frac{D}{2}} |\Sigma_k|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-\mu_k)' \Sigma_k^{-1} (x-\mu_k)\right\}, \quad \forall k : \pi_k \geq 0, \ \sum_{k=1}^{K} \pi_k = 1. \tag{1}$$

where the feature descriptor $x \in \mathbb{R}^D$; K is the number of Gaussian components; π_k, μ_k, and Σ_k correspond to the mixture weight, mean vector, and covariance matrix, respectively. We assume Σ_k to be a diagonal matrix with the variance vector σ_k^2. The parameters $\lambda = \{\pi_k, \mu_k, \Sigma_k, \ k = 1, \dots, K\}$ of GMM are estimated based on a large set of training SIFT descriptors by the Expectation Maximization (EM) algorithm to optimize the Maximum Likelihood (ML).

For a set of descriptors $X = \{x_1, \dots, x_N\}$ extracted from a STC patch, the soft assignment of descriptor x_i to component k is defined as:

$$\gamma_i^k = \frac{\pi_k u_k(x_i)}{\sum_{j=1}^{K} \pi_j u_j(x_i)}. \tag{2}$$

The Fisher Vector representation of X is $\Psi(X) = \{\rho_1, \tau_1, \dots, \rho_K, \tau_K\}$, where ρ_k and τ_k are the D-dimensional gradients with respect to the mean vector μ_k and the standard deviation σ_k of the kth Gaussian component:

$$\rho_k = \frac{1}{N\sqrt{\pi_k}} \sum_{i=1}^{N} \gamma_i^k \left(\frac{x_i - \mu_k}{\sigma_k}\right), \tag{3}$$

$$\tau_k = \frac{1}{N\sqrt{2\pi_k}} \sum_{i=1}^{N} \gamma_i^k \left[\frac{(x_i - \mu_k)^2}{\sigma_k^2} - 1\right]. \tag{4}$$

Compared to Bag-of-Words (BOW) based representations, Fisher Vector has the following merits: (1) BoW is a particular case of Fisher Vector, i.e., the gradient to the component weights of GMM. The additional gradients with respect to the means and variances in Fisher Vector provide extra distribution information of descriptors in the low-level feature space. (2) The Fisher Vector can be computed upon a much smaller visual vocabulary which significantly reduces the computational cost. (3) Fisher Vector performs quite well with simple linear classifiers which are efficient in both training and testing.

We follow the two normalization schemes introduced in [11], i.e., *L2* and power normalization. The *L2* normalization is used to remove the dependence on the proportion of class-specific information contained in a patch, in other words, to cancel the effect of different amount of foreground and background information contained in different images. The power normalization is proposed due to the fact that as the number of Gaussian components increases, Fisher Vector becomes peaky around zero in a certain dimension. This negatively impacts the computation of feature distance. The power normalization $f(z) = \text{sign}(z)|z|^\alpha$ with $0 < \alpha \leq 1$ is applied to each dimension z in the Fisher Vector. We utilize $\alpha = 0.5$ (i.e., the Hellinger kernel) to compute the signed square-root. In our representation, we first apply the power normalization and then the *L2* normalization.

In order to de-correlate the data to make it fitted more accurately by a GMM with diagonal covariance matrices, we apply a PCA on the SIFT descriptors to reduce them from $D = 128$ to 32. The number of Gaussian components K is empirically determined as 50. So each image patch of STC is represented as a feature vector with 3200 dimensions. SVM learning model is employed to generate a max-margin hyper plane in this feature space to classify the 62 STC categories. This hyper plane is defined as the STC predictor. The LIBLINEAR is used to implement SVM training and testing. Given an STC cropped from image frame, the STC predictor is able to compute its category from one of the 62 candidates as a label and output a 62-dimensional prediction scores as the probability of each category.

4. WORD CONFIGURATION IN CONDITIONAL RANDOM FIELD

Due to the cluttered background noise and multiple STC patterns in natural scene, the accuracy of STC prediction is limited. It only takes account of the appearance of a single STC in an image patch, but ignores the context information of neighboring STCs. In addition, it only performed one-by-one single STC prediction by ignoring the possible constraints of STC combinations in the lexicon model for work recognition. Moreover, some STCs tend to be classified to wrong categories in a similar structure, e.g., letter "O" and digit "0". Therefore the STC prediction can only be considered as preliminary results of word recognition in the character level. To rectify STC prediction and obtain recognized words compatible with dictionary, CRF model [6, 23] is used for word configuration based on the resulting SVM scores of STC prediction. CRF model is a discriminative undirected probabilistic graphical model $\langle V, E\rangle$ and encodes both the relationships between observations and category labels of nodes as well as the relationships between neighboring nodes in the graphical model. In the notations $\langle V, E\rangle$, V denotes the node set and E denotes edge set.

In CRF, each STC is defined as a random variable V_i, represented by a node. STC prediction assigns a category label to each node, which can also be considered as assigning an observation to each random variable. We set the total number of STC predictions, which is also the total number of nodes as $|V|$. The cost function of a CRF model is defined as Eq. (5).

$$L(V=c)=\sum_{i=1}^{|V|}L_i(V_i=c_i)+\lambda\sum_{(i,j)\in E}L_{ij}(V_i=c_i,V_j=c_j) \quad (5)$$

where V is the set of all nodes, c represents their corresponding category label, L_i is the cost function of single node to measure the suitability of category labels obtained from STC prediction, L_{ij} is the cost function of neighboring nodes to measure the compatibility of neighboring category labels obtained from STC predictions.

Fig. 2 illustrates the CRF model $\langle V, E\rangle$ of an image patch cropped from a video frame. Each STC is defined as a random variable node in CRF, and each pair of neighboring STC nodes is mutually connected by an edge. The involved cost functions of this graphical model are defined as:

$$\begin{aligned} L_i(V_i=c_i)&=1-Score(V_i)\\ L_{i,j}(V_i=c_i,V_j=c_j)&=1-Freq(c_i,c_j)\end{aligned} \quad (6)$$

where $L_i(V_i=c_i)$ is unary cost of a node, $L_{i,j}$ is pairwise cost of neighboring nodes, $Score(V_i)$ represents the STC prediction scores at node X_i, and $Freq(c_i,c_j)$ represents the frequency of the bigram lexicon c_i and c_j. By minimizing the cost function in Eq. (5), CRF model can improve the accuracy of word recognition on the basis of STC prediction.

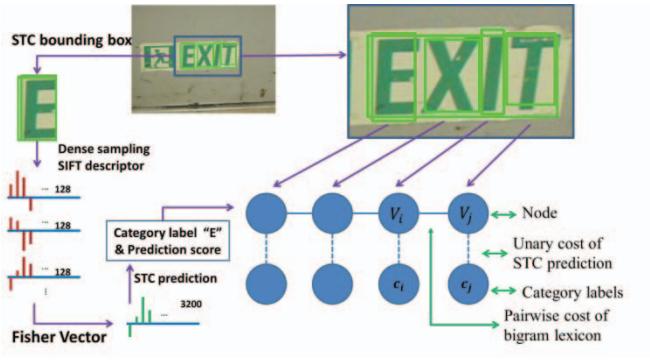

Fig. 2. On the left, STC is extracted by detection and tracking, and then transformed into Fisher Vector feature representation. SVM-based STC predictor is applied to obtain its category label and prediction score. On the right, in a tracking frame of scene text, each STC bounding box is defined as a node in CRF model. Unary cost of STC prediction and pairwise cost of bigram lexicon is defined in this graphical model.

5. EXPERIMENTS AND DISCUSSIONS

5.1. Datasets

To evaluate STC prediction and generate the bigram statistics in lexicon model, three public benchmark datasets and one self-collected dataset are used in our experiments: CHARS74K [3], ICDAR2003 [21], ICDAR2011 [22], and our collected dataset named as Video Text Reading Dataset (VTR). All of them are composed of image patches of single characters. In CHARS74K dataset, there are three types of text characters, including image-based characters cropped from natural scene image, hand-written characters, and computer-generated font characters. The first type is compatible with our STC prediction task. Numbers of samples in the 62 categories are similar. ICDAR2003 dataset contains 509 scene images and 2268 word-level text regions. The text regions are further divided into 11615 image patches of characters. All the 62 categories are also included in these patches, but the numbers of character patches between different categories are imbalanced. ICDAR 2011 dataset contains 229 scene images with 848 ground truth text regions in total. In our experiments, only CHARS74K samples are adopted to train the STC predictor. All three datasets are used to make statistics of the bigram frequency in lexical analysis.

Since our framework works on multiple frames, we collect a video dataset of text information from natural scene, which consists of 50 videos including both indoor and outdoor environments. These videos are captured by a moving and shaking camera, which results in some motion blur. Each video sample in this dataset has 30 frames per second and lasts around 15 seconds, so it contains about 450

frames, where the target signage is shot from about -45°~45° view angles. We uniformly sample 100 frames from each video as the effective frames in our experiments. The following experiments extract the STC trajectories in the 100 effective frames to perform multi-frame scene text recognition.

5.2. Experiment Design and Evaluation Measurement

We first perform an experiment of STC prediction on CHARS74K dataset to validate the effectiveness of the feature representation based on Fisher Vector. Then we perform the experiments of word recognition and compare the recognition results between using a single frame and multiple frames. In the multi-frame scene text recognition, we compare two schemes of fusing STC prediction scores.

To evaluate the performance of scene text recognition, we use different measurements for STC prediction and word recognition. In STC prediction, we measure the accuracy in a testing set by CH-RATE, which denotes the accuracy rate of STC prediction. In word recognition, difference between two words is measured by the edit distance. Whenever a word is updated by inserting a new character, deleting a character or modifying a character, its edit distance from the original word is increased by 1. However, it is too strict to only compute the accuracy of perfectly recognized words with edit distance 0 (ED0) to ground truth. Therefore, we also measure word recognition of the edit distance 1 (ED1) to ground truth. For the recognized words have 1 edit distance from, we still can recognize them in most cases.

5.3. STC prediction based on Fisher Vector

In CHARS74K dataset, 1860 STC samples are used to evaluate the performance of STC prediction under different feature representations. 930 samples are used for training and the other 930 samples are used for testing. This setup conforms to the standard training and testing splits in the benchmark evaluations of STC prediction as in [3].

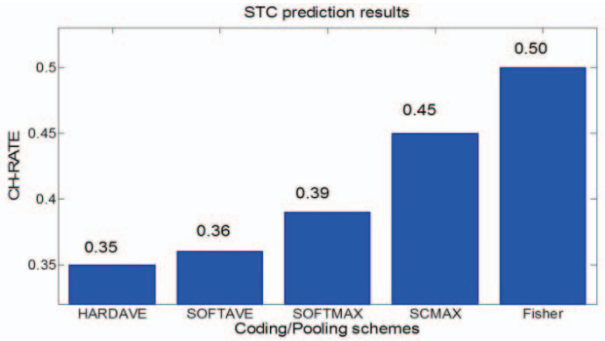

Fig. 3. STC prediction results in Chars74K, including all kinds of coding and pooling schemes. HARD: hard-assignment coding; SOFT: soft-assignment coding; SC: sparse coding; AVE: average pooling; MAX: max pooling. It shows that the Fisher Vector obtains much better performance under SIFT-based local feature descriptor than that in [18]. Other local feature descriptors like HOG under Fisher Vector will be evaluated in future work.

In our experiments of evaluating STC prediction, Fisher Vector is compared with multiple coding and pooling schemes under BOW models (see Fig. 3). They are applied to the same STC samples and identical local feature descriptor (SIFT). The coding and pooling schemes in [18] are input to SVM with χ^2 non-linear kernel, which involves much more computational cost than the linear kernel used in our method. As shown in Fig. 3, Fisher Vector significantly outperforms the coding and pooling schemes under BOW model by 5%.

5.4. Scene Text Recognition Using a Single Frame and Multiple Frames

To validate the effectiveness of multiple frames in scene text recognition, we first carry out the experiment by using a single frame on our VTR dataset -- Name here. Each sample in the dataset consists of multiple frames with text regions, and we could randomly choose one frame and perform scene text recognition. However, the image quality of the video frames is largely different, so it is unreasonable to use only one frame to evaluate text recognition in a sample. In our experiments, 10 frames, i.e., $1/10$ of the total number of effective frames in a video, are randomly selected to evaluate scene text recognition. Each STC bounding box generates a vector of prediction scores in 62 dimensions, since we define 62 categories of STCs. We compute the prediction scores of the corresponding STCs in the same trajectory from the randomly selected frames, and then calculate their mean as the result of scene text recognition in a single frame.

In comparison, we evaluate multi-frame text recognition. In text tracking, STC bounding boxes obtained from text detection are tracked in succeeding frames. We observe that although the motion blur due to camera shaking is minimized, not all STC bounding boxes in video frames are correctly tracked, and not all STC categories are correctly predicted. Therefore, we further fuse the STC prediction scores in each frame along its trajectory to improve the recognition accuracy. We propose two fusion methods. The first fusion method employs Majority Voting model, which makes statistics of category labels of STC prediction in all frames and choose the one in the highest frequency as the final result. For each STC trajectory, we generate a vector of predicted category labels from the frames. Then the highest-frequent label is computed as the result of STC prediction. All STC prediction results are then cascaded into word recognition result.

The second fusion method employs CRF model to fuse multi-frame STC prediction scores under the lexical constraints. CRF model is learned from ground truth text regions in CHARS74K, ICDAR2003, and ICDAR2011 datasets. We perform scene text detection and STC prediction in those ground truth regions, and use the prediction scores and bigram lexicons to train CRF model. Then the CRF model is applied to the STC prediction labels

and scores in multiple frames of text tracking. Fig. 4 displaces some example results of text tracking and word recognition from multiple frames.

Table 1. The accuracies of word recognition of single frame-based and multiple frames based methods on VTR dataset.

	CH-RATE	ED0	ED1
Single Frame	0.640	0.333	0.583
Multi-Frame (Majority)	0.680	0.389	0.611
Multi-Frame (CRF)	0.713	0.389	0.722

The experimental results in Table 1 demonstrate that multi-frame scene text recognition significantly improves the performance of STC prediction and word recognition in comparison with single-frame recognition. Majority voting suppresses the minority frames that generate incorrect STC prediction, so it obtains better performance of STC prediction and word recognition than single-frame method. Furthermore, CRF brings in lexical prior knowledge of bigram STCs. It further improves the performance of STC prediction and word recognition. Since the bigram lexical prior is calculated from ground truth words of three other datasets, and the size of our self-collected text video dataset is not large enough, CRF obtains the same performance on ED0 as majority voting. But we infer that CRF will give better performance as size increasing of text video dataset.

Fig. 4. Some example results of word recognition from multiple frames of text tracking. Right column shows recognized words.

6. CONCLUSION

We have proposed a scene text recognition method using multiple frames. Scene text detection is used to extract bounding boxes of text strings from the first frame as initial location of a text region to be tracked. Text tracking further tracks the bounding box to the subsequent frames of a video. STC prediction is then applied to each tracked bounding box. We employ a CRF model to configure text words, where the output score of STC prediction is used as node features and the lexical frequency of neighboring STCs are used as edge features. The combination of text extraction and tracking is able to improve efficiency in practical applications. The experimental results validate the effectiveness of our proposed tracking based scene text recognition in multiple frames. Our future work will focus on the design of more robust fusion methods to incorporate STC prediction scores of multiple frames to further improve the performance of word-level text recognition. We will continue to improve the feature representation of STC by exploring other local feature descriptors as well as coding and pooling schemes.

7. REFERENCES

[1] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *CVPR*, 2011.

[2] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *CVPR*, 2000.

[3] T. De-Campos, B. Babu, and M. Varma. Character recognition in natural images. In *VISAPP*, 2009.

[4] B. Epshtein, E. Ofek, and Y. Wexler. Detecting Text in nature scenes with stroke width transform. In *CVPR*, 2010.

[5] H. Grabner, C. Leistner, H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008.

[6] J. Lafferty, A. McCallum, F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[7] L. Liu, L. Wang, and X. Liu. In defense of soft-assignment coding. In *ICCV*, 2011.

[8] A. Mishra, K. Alahari, and C. Jawahar. Top-down and bottom-up cues for scene text recognition. In *CVPR*, 2011.

[9] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, 2002.

[10] K. Okuma, A. Taleghani. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004.

[11] F. Perronnin, J. Sanchez, and T. Mensink. Improving Fisher kernel for large-scale image classification. In *ECCV*, 2010.

[12] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang. Scene text recognition using part-based tree-structured character detection. In *CVPR*, 2013.

[13] J. Weinman, E. Leanred-Miller, and A. Hanson. Scene text recognition using similarity and a lexicon with sparse belief propagation. *IEEE TPAMI*, 2009.

[14] K. Wang, B. Bbenko, and S. Belongie. End-to-end scene text recognition. In *ICCV*, 2011.

[15] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *IJCV*, 2007.

[16] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *CVPR*, 2012.

[17] C. Yi and Y. Tian. Text string detection from natural scenes by structure-based partition and grouping. *IEEE TIP*, 2011.

[18] C. Yi, X. Yang, and Y. Tian. Feature representations for scene text character recognition: a comparative study. In *ICDAR*, 2013.

[19] X. Yin, K. Huang, and H. Hao. Robust text detection in natural scene images. *IEEE TPAMI*, 2013.

[20] Z. Ye, C. Yi, and Y. Tian. Reading labels of cylinder objects for blind persons. In *ICME*, 2013.

[21] http://algoval.essex.ac.uk/icdar/Datasets.html

[22] http://robustreading.opendfki.de/wiki/SceneText

[23] http://sourceforge.net/projects/hcrf/