# Video Text Detection with Text Edges and Convolutional Neural Network

Ping Hu
University of Chinese Academy of Sciences
Beijing, China
huping13@mails.ucas.ac.cn

Weiqiang Wang
University of Chinese Academy of Sciences
Beijing, China
wqwang@ucas.ac.cn

Ke Lu
University of Chinese Academy of Sciences
Beijing, China
luk@ucas.ac.cn

## Abstract

*Text and captions in videos provide useful information for content analysis and understanding. In this paper, we present an approach to detecting video text in a coarse-to-fine strategy. In the coarse phase we propose an efficient method to detect multi-scale candidate text regions with high recall. Then the candidate text regions are segmented and sent to the fine phase where a convolutional neural network(CNN) is applied to generate a confidence map for each candidate text region. Finally, the candidate text regions are further refined and partitioned into text lines by projection analysis. The CNN classifier in the fine phase enables feature sharing and robustly identifies text regions. The coarse phase sharply reduce the number of windows needed to be scanned by the CNN. The combination endows the proposed method with both efficiency and robustness when detecting video text. It was verified by experiment results on two publicly testing datasets and a dataset created by us.*

## 1. Introduction

Due to the rapid development of internet and multimedia technology, the tremendous increment of videos and images brings urgent need for efficient multimedia indexing and retrieving. For this task, methods are needed to automatically analyze the semantic information from the videos and images. Because of the close correlation to the video content, texts in video can play an important role in these method. Although numerous methods have been proposed, extracting texts from videos and images without constraints remains interesting and challenging due to the complexity of text appearance and background. In general the related techniques can be summarized into three steps: text detection, text segmentation and characters recognition. Text detection is an important step involving determining the existence of text regions and marking the corresponding position [14].

Different from the scene text, text in video shows several specifical features like, uniform color, high contrast to the background and so on. In this paper we focus on the detection of overlay text in videos. We propose a coarse-to-fine scheme for video text detection. On the first phase, we propose a edge-based method to quickly detect potential text regions with high recall, then three heuristic rules (minimum hight, minimum width and minimum area ratio of candidate pixel to its bounding box) are applied to eliminate false positives. On the second phase, a trained CNN classifier is adopted to assign a confident score to every pixel in the candidate region, and finally the confidence map of candidate regions are partitioned into text lines by projection analysis. The efficiency of the proposed method is evaluated with two publicly testing datasets and a 300-video-frame dataset proposed by us.

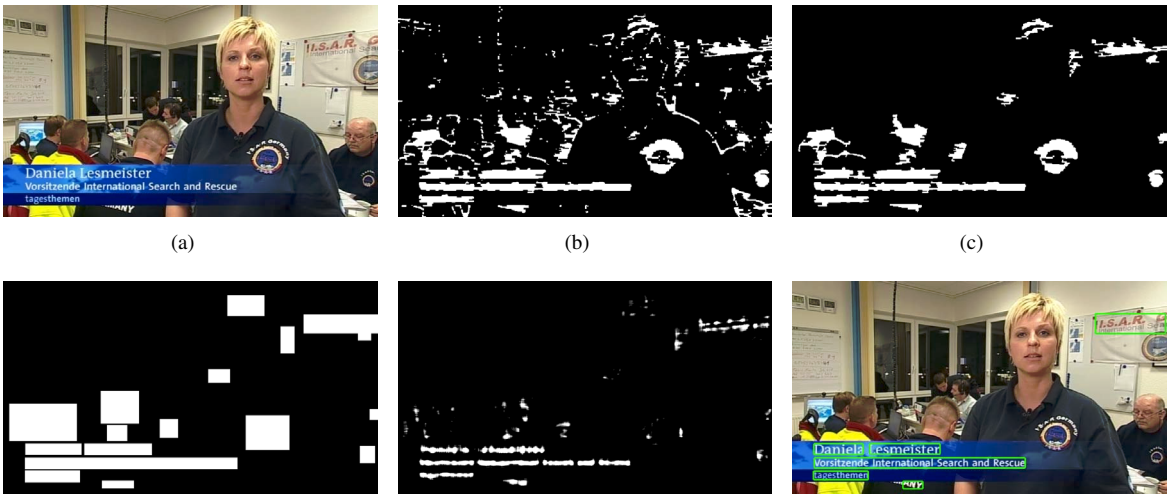
The rest of the paper is organized as follows. Section 2 presents the related works. In section 3, the proposed method is described in details. Section 4 reports the experimental results. Finally, the conclusion is drawn in Section 5.

## 2. Related Works

In many previous works, edge [12, 9], texture[7], color [6], corner [4], stroke width feature [17] are the most commonly adopted feature for video text detection.

Shivakumara et al. [9] proposed a combination of edge profiles and additional edge features to eliminate the false positives. Pan et al. [7] adopt a Waldboost classifier based on gradients features and horizontal and vertical projections to localize candidate regions, then a polynomial classifier with histogram of oriented gradient, LBP, DCT, Gabor filter, and wavelets is used to further verify. Wang et al. [10] take use of color clustering to separate the image into color layers and heuristic rules together with connected component analysis are used to localize text regions. In [16], text regions are extracted on the basis of Harris corner under the constraints of several heuristic rules. Zhao et al. [17] propose stroke unit connection operator to localize seed stroke

978-1-4799-6100-9/15/$31.00 ©2015 IEEE




(a) (b) (c)

(d) (e) (f)

Figure 1. Workflow of the proposed method. (a) the input image. (b) the candidate regions based on edges. (c) candidate regions after the false positive elimination. (d) binary map after connected component analysis. (e) confidence map. (f) Detection result

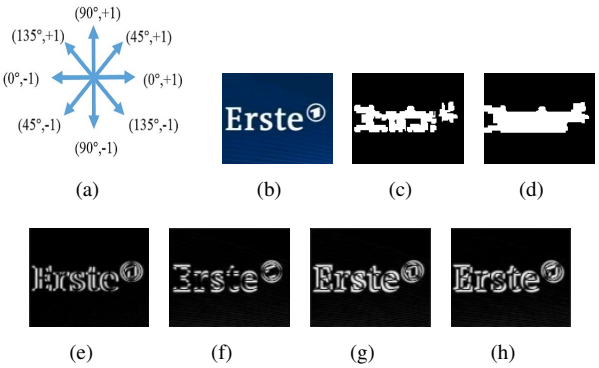
units, and the stroke shape distribution is extracted to train a SVM classifier.

Different from the sophisticated models and hand-engineering features, some approach attempts to directly learn the necessary feature from data have been proposed. Based on sparse representation, Zhao et al. [15] train two dictionaries, one for text and the other for natural scene, to distinguish text region from background. In [5, 13] mid-level features learned from data are used to generate discriminative patch representation. Saidane et al. [8] train a convolutional neural networks (CNN) to extract text pixels from cropped text regions. Wang et al. [11] use CNN with unsupervised pre-training to detect and recognize texts. This kind of method is always incorporated with sliding-window model.

Inspired by the previous work, the proposed method incorporates the efficiency of edge feature extraction and the robustness of CNN into a coarse-to-fine scheme to achieve good performance.

## 3. The proposed Method

The proposed method comprises two phases: the coarse candidate regions detection and the fine CNN text line localization. After building the image pyramid, the coarse stage localizes potential text regions in each layers of the pyramid. Then a CNN classifier scans candidate regions and generates confidence map in the fine stages. Candidate region are finally partitioned into text lines by projection analysis.




(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2. Example of coarse text detection. (a) edge gradient direction represented by$(\theta, \lambda)$. (b) the input image. (c) the detected text pixels. (d) the linked candidate regions. (e) the response of $R_{0\circ}$. (f) the response of $R_{90\circ}$. (g) the response of $R_{45\circ}$. (h) the response of $R_{135\circ}$.

### 3.1. Coarse Candidate Regions Detection

Generally video text have three prominent characteristics. First, text regions show strong contrast to the background. Second, text stroke shows various orientations and nearly constant width. Third, text structure shows compact spatial distribution. To quickly localize the potential regions, we perform the coarse detection by extracting pixel who has strokes of all four orientations in its neighborhoods.

To extract strokes of different directions, we first extract the edges of each frame with Sobel operator, and calculate the strength and direction of gradient for each edge pixel. The gradient directions can be quantized into 8 directions represented by two parameters $(\theta, \lambda)$. $\theta$ represents the orientation and $\lambda$ represents the polarity of the orien-

tation as shown in Figure 2(a). We disregard the polarity of orientation, the gradient directions are finally assigned into four directions $(0°, 45°, 90°, 135°)$. The gradient responses map of four direction are denoted as $R_\theta$ $(\theta = \{0°, 45°, 90°, 135°\})$ and shown in Figure 2(e)(f)(g)(h). We treat the four kinds of responses to be strokes in four directions.

Based on the three characteristics of video text, we define a pixel belong to the text region if it has all four kind of stroke pixels in its neighborhoods,

$$\phi(x,y) = \prod_{\theta=\{0°,45°,90°,135°\}} \sum_{i=-\sigma,j=-\sigma}^{i=+\sigma,j=+\sigma} R_\theta(x+i, y+j) \tag{1}$$

where $(x, y)$ is the position of the pixel in the image. $\sigma$ is the size of the neighborhood. For a given threshold $\lambda$, if $\phi(x, y) > \lambda$ the pixel at is labeled as text pixel. Then, we adopted a morphological dilation operation to link the candidate pixels into candidate regions. The example of detected text pixels are shown in Figure 2(c), and the example of linked region is shown in Figure 2(d)

These candidate regions has a high detection rate but relatively low precision, because background regions with complex texture may also be labeled as candidate regions. To reduce the number of regions which are sent to the next stage, we apply some constraints. A region is eliminated if:

- Height is lower than $MinHeight$
- Width is lower than $MinWidth$
- Area ratio of potential regions to its bounding box is lower than $MinRatio$

An example of all detected potential regions is shown in Figure 1(b), the result after false positive elimination is shown in Figure 1(c). The patches that are sent to next stage are shown in Figure 1(d).

## 3.2. Fine Text Line Detection

The patches within the bounding box of candidate regions are sent to this stage one after another and scanned by the CNN classifier to generate the confidence maps. Then the candidate patches are partition into text lines by projection analysis of their confidence maps.

### 3.2.1 Convolutional Neural Network

The convolutional neural network(CNN) has been used in a number of computer-vision tasks and greatly improve the performance. It is a novel architecture that can learn high-level feature and enables efficient feature sharing. In this section, we describe the CNN classifier used in this paper.

Similar to the application in [11, 1, 3], the network in this paper include two convolutional layers with average pooling steps. The number of filters for each layer are $n_1 = 96$ and $n_2 = 128$ respectively. The input is fixed 32-by-32 gray-scale images.

As applied in [11], we train the first layer in a unsupervised way. We randomly extracts 8-by-8 patches from 32-by-32 training patches to be contrast normalized and ZCA whitened into input vectors $x^{(i)} \in \Re^{64}$. Then, a variant of K-means described in [1] is used to learn a set of filters $D \in \Re^{n_1 \times 64}$. For a preprocessed input vector $x \in \Re^{64}$, the response $z$ of the first layer is computed as,

$$z = \max\{0, |D^\top x| - \alpha\} \tag{2}$$

where $\alpha = 0.5$.

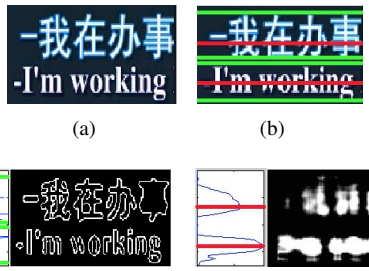
Given an input patch of 32-by-32, we can compute a 25-by-25-by-96 first layer response map. Then, we perform average pooling with 5-by-5 window to reduce its dimensions to 5-by-5-by-96. A 4-by-4-128 convolutional layer and a 2-by-2 average pooling layer stacked upon the first layer to obtain a 128-dimension feature vector. The output feature is sent to a SVM to generate a confident score. The output feature is fully connected to the classification layers and the parameters are trained by back-propagating the SVM classification error.

We use CNN classifier to compute each candidate a confidence map. When sent to the CNN classifier, the area of candidate regions are expanded to ensure the output confidence map is the same size as we detected. The confidence map of candidate patches in Figure 1(d) is shown in Figure 1(e)

### 3.2.2 Localization of Text Line

We observe that the projection analysis of confidence map can detect the center of every text line segmentation accurately rather than the boundaries. To localize the text accurately, we adopt projection analysis of Canny's edge map to determine the boundaries of text line.

With the confidence map, we first eliminate the false positives regions if the ratio of number of its pixels with positive confidence to its area is too low. Before the projection analysis, we set the negative score in the confidence map to be 0. Subsequently, for the confidence map and Canny's edge map of a candidate patch, the horizontal projection analysis is performed to locate the peaks and valleys with non-maximum suppression respectively. For every segments decided by two adjacent valleys, if there is a peak between them, it is selected as text line. Then a vertical projection is performed on the confidence map to relocate the text objects. An example is shown in Figure 3.




(a) (b)

(c) (d)

Figure 3. Example of text line segmentation. (a) input patch. (b) the detected boundaries(in green) and centers(in red) of text line. (c) Canny edge map and the valleys of the projection. (d) Confidence map and the peaks of the projection.

Table 1. Comparison on the Microsoft common test set

| Method | Recall | Precision | F-measure |
|---|---|---|---|
| Shivakumara et al. [9] | 0.92 | 0.90 | 0.91 |
| Zhao et al. [15] | 0.94 | 0.98 | 0.96 |
| Yang et al. [12] | 0.93 | 0.94 | 0.93 |
| Wang et al [11]. | 0.92 | 0.95 | 0.93 |
| Our Method | 0.96 | 0.97 | 0.96 |

Table 2. Comparison on the TV news dataset with pixel-based evaluation

| Method | Recall | Precision | F-measure |
|---|---|---|---|
| Yang et al. [12] | 0.86 | 0.81 | 0.83 |
| Our Method | 0.92 | 0.90 | 0.91 |

## 4. Experimental and Results

To train the CNN classifier, we take training data from ICDAR training set, the English subset of the Chars74k dataset and a video caption dataset created by us. There are 10000 positive and 20000 negative samples of 32-by-32 are used in the training step. Some training examples are shown in Figure 4(b). All Experiments in this paper are performed with Matlab 8 on a PC with an Intel Quad 2.83GHz CPU and 2 GB ram.

The evaluation of the proposed text detection method is performed on two public datasets and the dataset created by us. The caption video dataset created by us contains 300 typical frames collected various sources, including television series, movies, cartoon films and lectures. Some examples of the caption dataset and their detection result are shown in Figure 4(a). The two public datasets are: Microsoft common text set [2], and TV news dataset[1], which are collected from German TV news program. To provide a comparison to the existing method we adopt the evaluation method of [15] for the Microsoft common text set. The result is illustrate in Table 1. On the Microsoft common text set, the proposed method spends about 11 seconds to detect

[1] *http://www.yovisto.com/labs/VideoOCR/*


(a)

(b)

Figure 4. (a) Examples of our dataset and their detection reasult. (b) Examples of the training set.

Table 3. Comparison on the proposed dataset with pixel-based evaluation

| Method | Recall | Precision | F-measure |
|---|---|---|---|
| Wang et al. [11] | 0.96 | 0.92 | 0.93 |
| Our Method | 0.97 | 0.95 | 0.96 |

single frame on average, while the method in [11] spends more than 40 seconds. The experiment results shows that our method is more efficient and accurate.

Because the images in the Microsoft set are of low resolution and quality, we evaluate the proposed method on the TV news dataset and the dataset created by us, the pixel-wise detection result are shown in Table 2 and 3. The result show that our method is capable of handling a wide range of videos robustly.

## 5. Conclusion

In this paper we present an efficient and robust system based on a coarse-to-fine scheme to detect video text. In the coarse phase, the edges in different gradient orientation are used to detect the candidate text regions with a high recall but low precision. In the fine phase, a CNN classifier is adopted to compute each candidate region a confidence map based on which the candidate region are partitioned into text lines via projection analysis. This structure endows the proposed method the both efficiency and robustness. The experimental results validate our method.

## 6. Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61271434, No. 61232013, No. 61175115.

## References


[1] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *ICDAR*, pages 440–445, 2011. 3

[2] X.-S. Hua, W.-Y. Liu, and H.-J. Zhang. An automatic performance evaluation protocol for video text detection algorithms. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(4):498–507, 2004. 4

[3] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolution neural network induced mser trees. In *ECCV*, pages 497–511. 2014. 3

[4] X. Huang and H. Ma. Automatic detection and localization of natural scene text in video. In *ICPR*, pages 3216–3219, 2010. 1

[5] C.-Y. Lee, A. Bhardwaj, W. Di, V. Jagadeesh, and R. Piramuthu. Region-based discriminative feature pooling for scene text recognition. In *CVPR*, pages 4050–4057, 2014. 2

[6] X. Liu and W. Wang. Extracting captions from videos using temporal feature. In *ACMMM*, pages 843–846, 2010. 1

[7] Y.-F. Pan, C.-L. Liu, and X. Hou. Fast scene text localization by learning-based filtering and verification. In *ICIP*, pages 2269–2272, 2010. 1

[8] Z. Saidane and C. Garcia. Robust binarization for video text recognition. In *ICDAR*, volume 2, pages 874–879, 2007. 2

[9] P. Shivakumara, T. Q. Phan, and C. L. Tan. Video text detection based on filters and edge features. In *ICME*, pages 514–517, 2009. 1, 4

[10] K. Wang and J. A. Kangas. Character location in scene images from digital camera. *Pattern Recognition*, 36(10):2287–2299, 2003. 1

[11] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *ICPR*, pages 3304–3308, 2012. 2, 3, 4

[12] H. Yang, B. Quehl, and H. Sack. A framework for improved video text detection and recognition. *Multimedia Tools and Applications*, 69(1):217–245, 2014. 1, 4

[13] C. Yao, X. Bai, B. Shi, and W. Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *CVPR*, pages 4042–4049, 2014. 2

[14] Q. Ye and D. Doermann. Text detection and recognition in imagery: A survey. *TPAMI*, (99), 2014. 1

[15] M. Zhao, S. Li, and J. Kwok. Text detection in images using sparse representation with discriminative dictionaries. *Image and Vision Computing*, 28(12):1590–1599, 2010. 2, 4

[16] X. Zhao, K.-H. Lin, Y. Fu, Y. Hu, Y. Liu, and T. S. Huang. Text from corners: a novel approach to detect text and caption in videos. *Transactions on Image Processing*, 20(3):790–799, 2011. 1

[17] Y. Zhao, T. Lu, and W. Liao. A robust color-independent text detection method from complex videos. In *ICDAR*, pages 374–378, 2011. 1