A framework of reading timestamps for surveillance video

J. Cheng [1], *W. Dai* [2]

[1] *Computer School, Hubei Polytechnic University, Huangshi 435000, Hubei, China,*

[2] *School of Economics and Management, Hubei Polytechnic University, Huangshi 435003, Hubei, China*

Abstract

This paper presents a framework to automatically read timestamps for surveillance video. Reading timestamps from surveillance video is difficult due to the challenges such as color variety, font diversity, noise, and low resolution. The proposed algorithm overcomes these challenges by using the deep learning framework. The framework has included: training of both timestamp localization and recognition in a single end-to-end pass, the structure of the recognition CNN and the geometry of its input layer that preserves the aspect of the timestamps and adapts its resolution to the data. The proposed method achieves state-of-the-art accuracy in the end-to-end timestamps recognition on our datasets, whilst being an order of magnitude faster than competing methods. The framework can be improved the market competitiveness of panoramic video surveillance products.

Keywords: surveillance video, timestamp localization, timestamp recognition.

Citation: Cheng J, Dai W. A framework of reading timestamps for surveillance video. Computer Optics 2019; 43(1): 72-77. DOI: 10.18287/2412-6179-2019-43-1-72-77.

Acknowledgments: This work was supported in part by National Natural Science Foundation of China (No.71473074), and Hubei Polytechnic University Scientific Research Projects (18xjz03C).

Introduction

The timestamp plays a critical role in video semantics analysis. In surveillance videos, the timestamp indicates event time. The information of timestamp of video and image can be stored in the timestamp channel and video/image players can choose whether the timestamp is overlaid on each frame/image according users' option. Another way is that a timestamp is superimposed into a frame/image. For the old analog videos timestamps have to be superimposed into videos; for the nowadays videos timestamps may purposely be superimposed into videos so that they cannot be easily changed, of course videos may have both encoded timestamp information and the superimposed timestamp. This paper considers the common case in which a timestamp has been superimposed on a surveillance video, so the algorithm presented in this paper does not need to use these encoded timestamps.

Figure 1 shows the two frames with timestamp from surveillance videos. Hence, it is highly desired to develop the algorithms for reading the superimposed digital video timestamp, independently of the timestamp encoded in the timestamp channel.

In this paper, we propose a novel end-to-end framework which simultaneously locates and recognizes timestamp in scene images. As the first contribution, we present a model which is trained for both timestamp localization and recognition in a single learning framework. The proposed method locates and recognizes timestamp in surveillance video real time.

As the second contribution, we show how the state-of-the-art object localization methods [22, 23] can be extended for timestamp localization and recognition, taking into account specifics of timestamp such as the exponential number of classes and the sensitivity to hidden parameters such as timestamp aspect and rotation. The method achieves state-of-the-art results on our datasets and the timecost is faster than the our previous proposed methods.

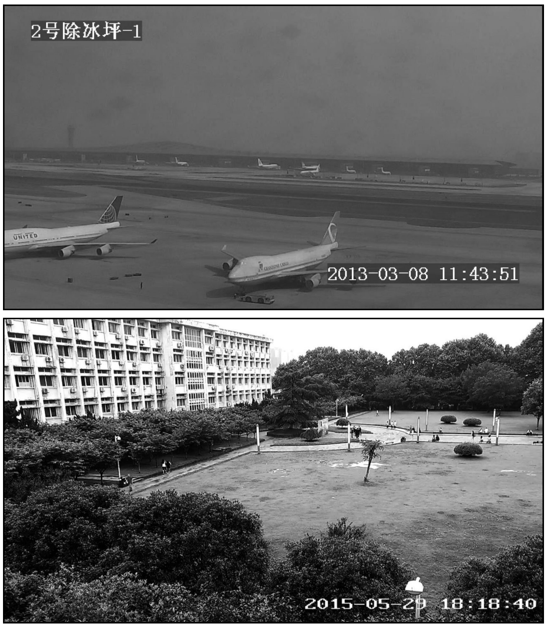

Fig. 1. Two frames with timestamp

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 presents the proposed method in details. The experimental results are presented in Section 4, followed by conclusion drawn in Section 5.

Related works

Timestamp localization

The timestamp localization is a very interesting problem in video analysis. Timestamp is a static region but the methods for detecting static regions cannot be used to detect the timestamps in surveillance videos because the scene objects appear as static regions too.

The methods for detecting static regions may be effective for some kinds of videos such as sports video, home video, and news videos because the scenes in these videos keep changing. However, the methods for detecting static regions are not applicable for surveillance videos because the surveillance cameras have little camera motion so that most of scene objects are static. The timestamp localization also can be considered as a scene text localization problem because the digits representing time and date form the text of timestamp. Jaderberg et al. [3] train a character-centric CNN [4], which takes a 24×24 image patch and predicts a text/no-text score, a character and a bigram class. The input image is scanned by the trained network in 16 scales and a text saliency map is obtained by taking the text/no-text output of the network. Given the saliency maps, word bounding boxes are then obtained by the run length smoothing algorithm. The method is further improved in [5], where a word-centric approach is introduced. First, horizontal bounding-box proposals are detected by aggregating the output of the standard Edge Boxes [6] and Aggregate Channel Feature [7] detectors. Each proposal is then classified by a Random Forest [8] classifier to reduce the number of false positives and its position and size is further refined by a CNN repressor, to obtain a more suitable cropping of the detected word image. Gupta et al. [9] propose a fully-convolutional regression network, drawing inspiration from the YOLO object detection pipeline [10]. An image is divided into a fixed number of cells (14×14 in the highest resolution), where each cell is associated with 7 values directly predicting the position, rotation and confidence of text. The values are estimated by translation-invariant predictors built on top of the first 9 convolutional layers of the popular VGG-16 architecture [11], trained on synthetic data. Tian et al. [12] adapt the Faster R-CNN architecture [13] by horizontally sliding a 3×3 window on the last convolutional layer of the VGG-16 [11] and applying a Recurrent Neural Network to jointly predict the text / non-text score, the y-axis coordinates and the anchor side-refinement. Similarly, Liao et al. [14] adapt the SSD object detector [15] to detect horizontal bounding boxes. Ma et al. [16] adapt the Faster R-CNN architecture and extend it to detect text of different orientations by adding anchor boxes of 6 hand-crafted rotations and 3 aspects. However, the existing text localization algorithms cannot get the satisfactory results.

Timestamp recognition

Timestamp recognition is a special case of the timestamp recognition problem. The timestamp recognition also can be considered as a scene text recognition problem because the digits representing time and date form the text of timestamp. Jaderberg et al. [5] take a cropped image of a single word, resize it to a fixed size of 32×100 pixels and classify it as one of the words in a dictionary. In their setup, the dictionary contains 90000 English words and words of the training and testing set. The classifier is trained on a dataset of 9 million synthetic word images uniformly sampled from this dictionary. Shi et al. [17] train a fully-convolutional network with a bidirectional LSTM using the Connectionist Temporal Classification (CTC), which was first introduced by Graves et al. [18] for speech recognition to eliminate the need for pre-segmented data. Unlike the proposed method, Shi et al. [17] only recognize a single word per image (i.e. the output is always just one sequence of characters), they resize the source image to a fixed-sized matrix of 100×32 pixels regardless of how many characters it contains and the method is significantly slower because of the LSTM layer.

Proposed framework

This section presents the methods of localizing and recognizing timestamps for each individual video. Two observations are obtained from the collected timestamps. The first one is that a timestamp consists of one/two lines of digits representing date and time within a rectangle surrounding the digits. The other is that the timestamp digits are in the same color. Hence, the digit color of a timestamp can be known through learning from the instances of its s-digits (s-digits are the digits on the second place of timestamps). Thus, all digits of the given timestamp can be extracted by using the learnt digit color. Based on the above discussion, a procedure for removing timestamps is formed. The proposed model localizes timestamp regions in a given scene image and provides timestamp transcription as a sequence of characters for all regions with timestamp. The model is jointly optimized for both timestamp localization and recognition in an end-to-end training framework.

Fully convolutional network

We adapt the YOLOv2 architecture [10] for its accuracy and significantly lower complexity than the standard VGG-16 architecture [11], as the full VGG-16 architecture requires 30 billion operations just to process a 224×224 (0.05Mpx) image [10]. Using YOLOv2 architecture allows us to process images with higher resolution, which is a crucial ability for timestamp recognition – processing at higher resolution is required because a 1Mpx scene image may contain timestamp which is 10 pixels high, so scaling down the source image would make the timestamp unreadable.

The proposed method uses the first 18 convolutional and 5 max pool layers from the YOLOv2 architecture, which is based on 3×3 convolutional filters, doubling the number of channels after every pooling step and adding 1×1 filters to compress the representations between the 3×3 filters [10]. We remove the fully-connected layers to make the network fully convolutional, so our model final layer has the dimension of

$$\frac{W}{32}\times\frac{H}{32}\times 1024,$$

where W and H denote source image width and height [10].

Region proposals

Similarly to Faster R-CNN [13] and YOLOv2 [10], we use a Region Proposal Network (RPN) to generate region proposals, but we add rotation γ_θ which is crucial for a successful timestamp recognition. At each position of the last convolutional layer, the model predicts k rotated bounding boxes, where for each bounding box γ we predict 6 features – its position γ_x, γ_y, its dimensions γ_w, γ_h, its rotation γ_θ and its score γ_p, which captures the probability that the region contains timestamp.

The bounding box position and dimension is encoded with respect to predefined anchor boxes using the logistic activation function, so the actual bounding box position (x, y) and dimension (w, h) in the source image is given as

$$x = \sigma(\gamma_x) + c_x\,, \tag{1}$$

$$y = \sigma(\gamma_y) + c_y\,, \tag{2}$$

$$w = \alpha_w \exp(r_w)\,, \tag{3}$$

$$h = \alpha_h \exp(r_h)\,, \tag{4}$$

$$\theta = r_\theta\,, \tag{5}$$

where c_x and c_y denote the offset of the cell in the last convolutional layer and α_w and α_h denote the predefined height and width of the anchor box α. The rotation $\theta \in (-(\pi/2), \pi/2)$ of the bounding box is predicted directly by r_θ.

We followed the approach of Redmon et al. [10] and found suitable anchor box scales and aspects by k-means clustering on the aggregated training set. Requiring the anchor boxes to have at least 60 % intersection-over-union with the ground truth led to $k = 14$ different anchor boxes dimensions.

For every image, the RPN produces $W32 \times H32 \times 6k$ boxes, where k is the number of anchor boxes in every location and 6 is the number of predicted parameters (x, y, w, h, θ and the timestamp score).

Bilinear sampling

Each region located in the previous stage has a different size and rotation and it is therefore necessary to map the features into a tensor of canonical dimensions, which can be used in recognition.

Faster R-CNN [13] uses the RoI pooling approach of Girshick [19], where a $w \times h \times C$ region is mapped onto a fixed-sized $W' \times H' \times C$ grid (7×7×1024 in their implementation), where each cell takes the maximum activation of the $(w / W) \times (h / H)$ cells in the underlying feature layer.

In our model, we instead use bilinear sampling to map a $w \times h \times C$ region from the source image into a fixed-height $(wH' / h) \times H' \times C$ tensor ($H' = 32$). This feature representation has a key advantage over the standard RoI approach as it allows the network to normalize rotation and scale, but at the same to persist the aspect and positioning of individual characters, which is crucial for timestamp recognition accuracy.

The transformation allows for shift and scaling in x- and y- axes and rotation and its parameters are taken directly from the region parameters.

Timestamp recognition

Given the normalized region from the source image, each region is associated with a sequence of characters or rejected as not timestamp in the following process. The main problem one has to address in this step is the fact, which timestamp regions of different sizes have to be mapped to character sequences of different lengths. Traditionally, the issue is solved by resizing the input to a fixed-sized matrix(typically 100×32) and the input is then classified by either making every possible character sequence (i.e. every word) a separate class of its own, thus requiring a list of all possible outputs in the training stage, or by having multiple independent classifiers, where each classifier predicts the character at a predefined position. Our model exploits a novel fully-convolutional network (see Table 1), which takes a variable-width feature tensor $\overline{W} \times H' \times C$ as an input ($\overline{W} = wH'/h$) and outputs a matrix $(\overline{W}/4) \times |\tilde{A}|$, where A is the alphabet (e.g. all English characters). The matrix height is fixed (it’s the number of character classes), but its width grows with the width of the source region and therefore with the length of the expected character sequence.

As a result, a single classifier is used regardless of the position of the character in the word (in contrast to Jaderberg et al. [20], where there is an independent classifier for the character “A” as the first character in the word, an independent classifier for the character “*A*” as the second character in the word, etc). The model also does not require prior knowledge of all words to be located in the training stage, in contrast to the separate class per character sequence formulation. The model uses Connectionist Temporal Classification (CTC) [17] to transform variable-width feature tensor into a conditional probability distribution over label sequences. The distribution is then used to select the most probable labelling sequence for the timestamp region. Let $y = y_1, y_2, \ldots, y_n$ denote the vector of network outputs of length n from an alphabet A extended with a blank symbol “–”.

In training, an objective function that maximizes the log likelihood of target labeling $p(w \,|\, y)$ is used. In every training step, the probability $p(w_{gt} \,|\, y)$ of every timestamp region in the mini-batch is efficiently calculated using a forward-backward algorithm similar to HMMs training and the objective function derivatives are used to update network weights, using the standard back-propagation algorithm (w_{gt} denotes the ground truth transcription of the timestamp region).

At test time, the classification output w^* should be given by the most probable path $p(w \,|\, y)$, which unfortunately is not tractable, and therefore we adapt the approximate approach of taking the most probable labelling. At the end of this process, each timestamp region in the image has an associated content in the form of a character sequence, or it is rejected as not timestamp when all the labels are blank. The model typically produces many different boxes for a single timestamp area in the image; we therefore suppress over-lapping boxes by a standard non-maxima suppression algorithm based on the timestamp recognition confidence, which is the $p(w^{*}|y)$ normalized by the timestamp length.

Training

The training dataset for evaluating the proposed timestamp localization and recognition algorithm consists of 300 video clips (704×704) and 300 video clips (1280×720) cropped from the surveillance videos. Each clip is about 20 second long with a working digital video timestamp.

We pretrain the localization CNN using the 600 video clips for 3 epochs. The recognition CNN is pretrained on the 600 video clips for 3 epochs, with weights randomly initialized from the $N(0,1)$ distribution. As the final step, we train both networks simultaneously for 3 epochs on the surveillance video dataset. For every video, we randomly crop up to 30 % of its width and height. We use standard Stochastic Gradient Descent with momentum 0.9 and learning rate 10^{-3}, divided by 10 after each epoch.

Experimental results

This section evaluates the proposed framework in two aspects. The proposed framework uses two Hikvision™ network cameras and its software is implemented using C++ on a workstation with Intel i7 3.10 GHz CPU and 8 GB memory. Two kinds of experiments are conducted to evaluate the framework. The first kind of experiments is on accuracy and computing time of s-digit localization and timestamp localization. The second is on accuracy of timestamp recognition for surveillance videos. Here experimental works are presented to verify that our algorithm is feasible and has good performance.

Dataset preparation and experiment setting

(1) Original video database

The dataset for evaluating the proposed timestamp localization and recognition algorithm consists of 1000 video clips (704×576) and 1000 video clips (1280×720) cropped from the surveillance videos. Each clip is about 40 second long with a working digital video timestamp.

(2) Synthetic video database with OpenCV Library

To demonstrate the proposed framework is robust to different kinds of video, we tried to generate another synthetic video database with the OpenCV function library in C and C++ coding language on visual studio 2010. For each video, we insert a superimposed working timestamp to frames for the synthetic video database generation. The synthetic video database includes 1000 video clips. Each clip is about 30 second long with a working digital video timestamp.

(3) TRECVID 2017 video database

We use i-LIDS airport surveillance video data from received 2017 video database to test the proposed framework. The data consist of about 150h of airport surveillance video data (courtesy of the UK Home Office). We tried to generate 1000 video clips from i-LIDS airport surveillance video. Each clip is about 20 second long with a working digital video timestamp.

(4) Evaluation standards

To evaluate the reading timestamps efficiency, the recall rate (R_r) and precision rate (R_p) are used, which are common standard in video and image related detection and classification research. The recall rate is the percentage of correctly located or recognized timestamps in videos among all video databases; a high recall rate can well prove the localization or recognition timestamps accuracy.

$$R_r = \frac{N_c}{N_c + N_m} \times 100\%, \tag{6}$$

$$R_p = \frac{N_c}{N_c + N_f} \times 100\%, \tag{7}$$

where N_c is the number of correctly located or recognized timestamps in videos; N_m is the number of missed located or recognized timestamps in videos; N_f is the number of falsely located or recognized timestamps in videos.

Experiments on timestamp localization

In this section, the proposed framework compared to our previous method in [21] for of timestamp localization. An experiment is done to evaluate the accuracy and computing time of timestamp localization using three video databases. The results are given in Table 1, Table 2 and Table 3. Total indicate the numbers of test videos; μ and σ are the means and the standard deviations of computing times of locating the timestamp for a batch of videos. The experiments results show that our method can achieve a very high accuracy more than the proposed method in [21] for timestamp localization. The experimental results also show that this method can accurately locate the timestamp in a very low cost of computing.

Experiments on timestamp recognition

Here we conduct the experiments to evaluate the accuracy of timestamp recognition using three video databases in section 4.1, and compared the results to our previous method in [21]. The results are given in Table 4, Table 5 and Table 6. The experiments results show that our method can achieve a very high accuracy more than the proposed method in [21] for timestamp recognition.

Conclusions and future work

A novel framework for timestamp localization and recognition was proposed. The model is trained for both timestamps localization and recognition in a single training framework.

The proposed model achieves state-of-the-art accuracy in the end-to-end timestamp recognition on our dataset, whilst being an order of magnitude faster than the previous methods in [21]. Our model showed that the state-of-the-art object localization methods [22, 23] can be extended for timestamp localization and recognition, taking into account specifics of timestamp, and still maintaining a low computational complexity.

Table 1. Accuracy and computing time of timestamp localization for original video database

Method	Resolution	Total	N_c	N_m	N_f	R_r (%)	R_p(%)	Computing time (second)	
								μ	σ
The method in [21]	704×576	1000	895	56	49	94.11	94.81	1.489	0.236
	1280×720	1000	912	41	47	95.70	95.10	2.368	0.443
Proposed	704×576	1000	976	13	11	98.69	98.89	0.985	0.152
	1280×720	1000	985	9	6	99.09	99.39	1.526	0.245

Table 2. Accuracy and computing time of timestamp localization for synthetic video database

Method	Total	N_c	N_m	N_f	R_r (%)	R_p(%)	Computing time (second)	
							μ	σ
The method in [21]	1000	876	68	56	92.80	93.99	1.356	0.338
Proposed	1000	965	21	14	97.87	98.57	0.865	0.263

Table 3. Accuracy and computing time of timestamp localization for TRECVID 2017 video database

Method	Total	N_c	N_m	N_f	R_r (%)	R_p(%)	Computing time (second)	
							μ	σ
The method in [21]	1000	886	56	58	94.06	93.86	1.556	0.456
Proposed	1000	981	11	8	98.89	99.19	1.085	0.358

Table 4. Accuracy of timestamp recognition for original video database

Method	Resolution	Total	N_c	N_m	N_f	R_r (%)	R_p(%)
The method in [21]	704×576	1000	886	68	46	92.87	95.06
	1280×720	1000	908	45	47	95.28	95.08
Proposed	704×576	1000	971	15	14	98.48	98.58
	1280×720	1000	978	11	11	98.89	98.89

Table 5. Accuracy of timestamp recognition for synthetic video database

Method	Total	N_c	N_m	N_f	R_r (%)	R_p(%)
The method in [21]	1000	873	75	52	92.09	94.38
Proposed	1000	958	25	17	97.46	98.26

Table 6. Accuracy of timestamp recognition for TRECVID 2017 video database

Method	Total	N_c	N_m	N_f	R_r (%)	R_p(%)
The method in [21]	1000	878	68	54	92.81	94.21
Proposed	1000	965	19	16	98.07	98.37

We also demonstrated the advantage of the joint training for the end-to-end task, by outperforming the ad-hoc combination of the state-of-the-art localization and state-of-the-art recognition methods [25, 27], while exploiting the same training data.

Last but not least, we showed that optimizing localization accuracy on timestamps bounding boxes might not improve performance of an end-to-end system, as there is not a clear link between how well a method fits the bounding boxes and how well a method reads timestamp. Future work includes extending the training set with more realistic effects, single characters and digits. This method can be improved the market competitiveness of panoramic video surveillance products. This technology can not only improve the economic ability of the enterprise, but also support the innovation and development of the enterprise.

References

[1] Karatzas D, Shafait F, Uchida S, Iwamura M, Bigorda LGi, Mestre SR, Mas J, Mota DF, Almazàn JA, de las Heras LP. ICDAR 2013 robust reading competition. Proc 12th International Conference on Document Analysis and Recognition 2013: 1484-1493. DOI: 10.1109/ICDAR.2013.221.

[2] Karatzas D, Gomez-Bigorda L, Nicolaou A, Ghosh S, Bagdanov A, Iwamura M, Matas J, Neumann L,

Chandrasekhar VR, Lu S, Shafait F, Uchida S, Valveny E. Proc 13th International Conference on Document Analysis and Recognition (ICDAR) 2015: 1156-1160. DOI: 10.1109/ICDAR.2015.7333942.

[3] Jaderberg M, Vedaldi A, Zisserman A. Deep features for text spotting. In Book: Fleet D, Pajdla T, Schiele B, Tuytelaars T, eds. Computer Vision – ECCV 2014. Cham: Springer; 2014: 512-528. DOI: 10.1007/978-3-319-10593-2_34.

[4] Lécun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998; 86(11): 2278-2324. DOI: 10.1109/5.726791.

[5] Jaderberg M, Simonyan K, Vedaldi A, Zisserman A. Reading text in the wild with convolutional neural networks. International Journal of Computer Vision 2016; 116(1): 1-20. DOI: 10.1007/s11263-015-0823-z.

[6] Zitnick CL, Dollár P. Edge boxes: Locating object proposals from edges. In Book: Fleet D, Pajdla T, Schiele B, Tuytelaars T, eds. Computer Vision – ECCV 2014. Cham: Springer; 2014: 391-405. DOI: 10.1007/978-3-319-10602-1_26.

[7] Dollar P, Appel R, Belongie S, Perona P. Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 2014; 36(8): 1532-1545. DOI: 10.1109/TPAMI.2014.2300479.

[8] Bosch A, Zisserman A, Munoz X. Image classification using random forests and ferns. IEEE International Conference on Computer Vision 2007: 1-8. DOI: 10.1109/ICCV.2007.4409066.

[9] Gupta A, Vedaldi A, Zisserman A. Synthetic data for text localisation in natural images. IEEE Conference on Computer Vision and Pattern Recognition 2016: 2315-2324. DOI: 10.1109/CVPR.2016.254.

[10] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016: 779-788. DOI: 10.1109/CVPR.2016.91.

[11] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint v6 2015. Source: ⟨https://arxiv.org/abs/1409.1556⟩.

[12] Tian Z, Huang W, He T, He P, Qiao Y. Detecting text in natural image with connectionist text proposal network. In Book: Leibe B, Matas J, Sebe N, Welling M, eds. Computer Vision – ECCV 2016. Cham: Springer; 2016: 56-72. DOI: 10.1007/978-3-319-46484-8_4.

[13] Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 2017; 39(6): 1137-1149. DOI: 10.1109/TPAMI.2016.2577031.

[14] Liao M, Shi B, Bai X, Wang X, Liu W. TextBoxes: A fast text detector with a single deep neural network. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17) 2017: 4161-4167.

[15] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC. SSD: Single shot multibox detector. In Book: Leibe B, Matas J, Sebe N, Welling M, eds. Computer Vision – ECCV 2016. Cham: Springer; 2016: 21-37. DOI: 10.1007/978-3-319-46448-0_2.

[16] Ma J, Shao W, Ye H, Wang L, Wang H, Zheng Y, Xue X. Arbitrary-oriented scene text detection via rotation proposals. IEEE Transactions on Multimedia 2018; 20(11): 3111-3122. DOI: 10.1109/TMM.2018.2818020.

[17] Shi B, Bai X, Yao C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 2016; 39(11): 2298-2304. DOI: 10.1109/TPAMI.2016.2646371.

[18] Graves A, Gomez F. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. Proc 23rd International Conference on Machine Learning 2006: 369-376. DOI: 10.1145/1143844.1143891.

[19] Girshick R. Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV) 2015: 1440-1448. DOI: 10.1109/ICCV.2015.169.

[20] Jaderberg M, Simonyan K, Vedaldi A, Zisserman A. Synthetic data and artificial neural networks for natural scene text recognition. NIPS Deep Learning Workshop 2014. Source: ⟨https://arxiv.org/abs/1406.2227⟩.

[21] Yu X, Cheng J, Wu S, Song W. A framework of timestamp replantation for panorama video surveillance. Multimedia Tools and Applications 2016; 75(17): 10357-10381. DOI: 10.1007/s11042-015-3051-1.

Authors' information

Jun Cheng (b.1980) received his Ph.D. degree from Central China Normal University in 2016. His major research interests include video surveillance, machine learning, and e-learning. He is a lecturer at the Computer School of Hubei Polytechnic University. He has published many papers in related journals, such as Multimedia Tools and Applications, International Journal of Information and Computer Security, etc.

Wei Dai (b.1981) received his Ph.D. degree from Wuhan University of Technology in 2012. His major research interests include neural network, machine learning, and electronic commerce. He is an associate professor at the School of Economics and Management of Hubei Polytechnic University. He has published many papers in related journals.

Received May 25, 2018. The final version – October 10, 2018.