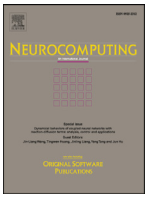
Neurocomputing 333 (2019) 284–291

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Elite Loss for scene text detection

Xu Zhao[a,b], Chaoyang Zhao[a,b,*], Haiyun Guo[a,b], Yousong Zhu[a,b], Ming Tang[a,b], Jinqiao Wang[a,b]

[a] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, No.95, Zhongguancun East Road, Beijing 100190, China
[b] University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Received 21 August 2018
Revised 4 December 2018
Accepted 26 December 2018
Available online 29 December 2018

Communicated by Dr. Kaizhu Huang

Keywords:
Scene text detection
Elite Loss
Object detection

ABSTRACT

Many scene text detection approaches generate foreground segmentation maps to detect the text instances. In these methods, usually all the pixels within the bounding box regions of the text are equally treated as foreground during the training process. However, different from the general object segmentation problem, we argue that not all the pixels across the text bounding box region contribute equally for locating the text instance. Specifically, some in-box not-on-stroke pixels even degrade the detection performance. Moreover, for the segmentation based methods with a regression step applied to predict the corresponding bounding box on each pixel, not all the pixels need to be fully trained to predict foreground texts. Therefore, in this paper, we propose Elite Loss, which is intended to down-weight the contributions of the in-box not-on-stoke pixels while paying more attention to the on-stoke pixels. Furthermore, we design a segmentation-based method to validate the effectiveness of the proposed Elite Loss. Extensive experiments demonstrate that our methods achieve the state-of-the-art results on all three challenging datasets, with the F-score of 0.855 on ICDAR2015, 0.425 on COCO-Text, and 0.819 on MSRA-TD500.

1. Introduction

Scene text detection has attracted more and more attention for its important role in many computer vision tasks. It aims at localizing the text with bounding boxes of words or text lines. Various methods have been proposed to tackle this problem [1–5]. The main challenges of scene text detection are the large variety of texts in scales, layouts, fonts, and orientations, as well as the cluttered background which is easily confused with text. Traditional text detection approaches are mostly bottom-up [6–8]. Lots of hand-craft features are designed to distinguish text from background regions. But these methods perform poorly on complex scenes.

Recently, many deep learning based methods are proposed to detect texts. Some methods are evolved from general object detection methods like SSD [9] or Faster RCNN [10]. They utilized the reference boxes to detect text, such as TextBoxes [11], TextBoxes++ [12], SegLink [1], RRPN [5], R^{2}CNN [13], RRD [14] FSTN [15], etc. These methods shows some improvement over traditional approaches, but they cannot deal with the multi-oriented texts well, as discussed in [4].

Another category of deep learning based methods are inspired by FCN [16], which is widely used in various segmentation tasks. Zhang et al. [17] and He et al. [18] used FCN to locate raw text regions, but they need complicated post-processing steps. Lyu et al. [19] uses corners and position-sensitive segmentation maps to segment each text instances. Liu et al. [20] proposed MCN, in which they designed the Markov Clustering Network to group the segmented foreground pixels into text instances. Recently, some methods such as EAST [2] and Direct Regression [4,21] are proposed. They regress a bounding box on each pixel's location, in an end-to-end way. We call these methods *segmentation based methods* because they detect text in a segmentation-like style.

The segmentation based methods detect the text instances by pixel-wise predictions. However, due to the lack of fine-grained text segmentation annotations, they usually assume all the pixels inside the text's bounding box as foreground. This is different from the general object segmentation task, where the sub-regions in the target segmentation map have consistent counters with the original objects. We consider that the use of the raw ground truth in the segmentation step is a trade-off caused by the lack of ground truth segmentation maps of text strokes. However, this leads to the optimization error in the backward propagation process of the

[*] Corresponding author at: National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, No.95, Zhongguancun East Road, Beijing 100190, China.
E-mail address: chaoyang.zhao@nlpr.ia.ac.cn (C. Zhao).

https://doi.org/10.1016/j.neucom.2018.12.009
0925-2312/© 2019 Elsevier B.V. All rights reserved.

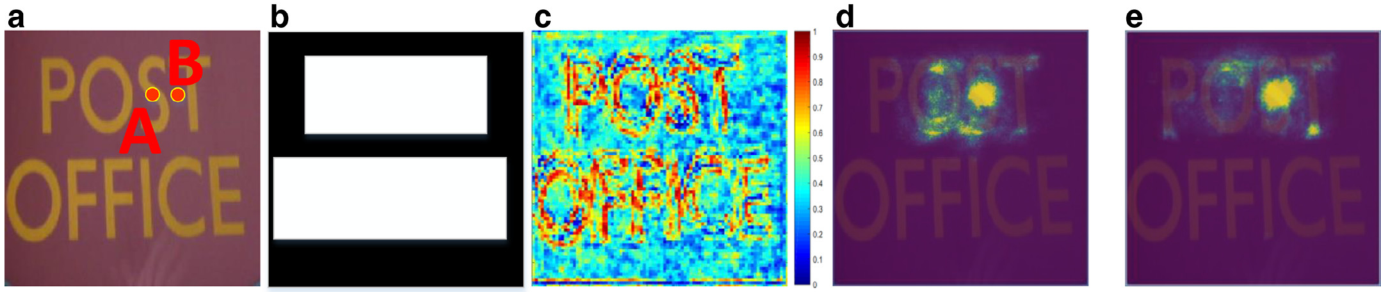


Fig. 1. (a, b):A training image and its label map. (c): The output score map of a segmentation based method after the first iteration. Those pixels on the text stroke have larger confidence on the feature map, indicating they are easier to be learned and better to be located. (d,e): The effective receptive field of the pixel on the score map at the location of point A, after training (d) without Elite Loss (e) with Elite Loss.

networks. Fig. 1(c) shows an output score map of a segmentation based detector [2] after the first iteration when it is being fine-tuned from ImageNet-pretrained models. It can be seen that the detector has high activations on the stroke areas and relatively low activations on the smooth areas outside the strokes. Obviously, the pixels on the in-box outside-stroke areas cannot be easily distinguished from the outside-box background pixels for the similar appearance. This means that pixels on the strokes capture the most distinctive characteristics of the text and thus they are easy to learn, comparing to the pixels on the smooth outside-stroke region. For example, from Fig. 1 (a–c), it can be seen that the pixel A has a stronger response at its corresponding location on the score map than the pixel B after the first iteration.

This paper takes example for the category of segmentation based methods which are with an additional regression step [2,4], to further inspect the above phenomenon. These methods perform the classification and bounding box regression on each pixel of the output segmentation maps to detect the text instances. Each pixel outputs one text instance bounding box if it is classified as positive. We can conclude that the pixels of the score map are the basic elements for the detection task and act independently of each other. In this paper, we call these basic elements as *predicting units* to better describe their roles. These methods treat all the predicting units of the same corresponding text instance equally. In this case, the total training loss of all the predicting units of the text instance may be easily dominated by those in-box not-on-stroke predicting units, for they usually have high loss value. However, the essential on-stroke predicting units are less considered, for they usually have small loss value. Meanwhile, since many of the not-on-stroke predicting units tend to have similar appearances with predicting units on the background (e.g., the unit at the location of pixel B in Fig. 1), forcing the network to distinguish the hard not-on-stroke predicting units as foreground may lead to many false detections on the background region.

From another view, for the predicting units of the same corresponding text instance, their tasks are duplicated. That is they are all able to find the same text instance independently. Thus it is unnecessary to cost much on those not-on-stroke but hard predicting units. The detector should focus on predicting units that represent the text instance better to learn a robust text detector.

It is worth to note that the not-on-stroke but *hard* predicting units are not the same as the *hard samples* chosen from all training samples in the hard example mining [22] or bootstrapping [23] procedure that should be considered more to benefit the performance. The hard predicting units here are the positive samples which are relatively hard but unnecessary to learn. These not-on-stroke but hard predicting units can also be regarded as units with noise labels in a sense, because their appearance is similar to the background and it is unreasonable to consider them the typical positive samples during training.

As analyzed above, we consider the on-stroke predicting units capture the instinct characteristics of text regions, so we call them as *elite* predicting units. In this paper, we propose a new loss reweighting strategy to train the detector. We re-weight the classification losses of the prediction units to automatically lower the not-on-stroke but hard predicting units' contributions to the loss during training. This helps the detector to focus on learning better features to correctly classify the elite on-stroke predicting units on the foreground. We call this reweighed loss as *Elite Loss*, for the reason that it focuses more on those elite predicting units (i.e., the on-stroke prediction units), and pays less attention to the noisy predicting units, i.e. not-on-stroke predicting units. The Elite Loss is flexible in its specific forms and it is effective. It improves our self-built baseline detector significantly and reaches the new state-of-art on various benchmarks. In Fig. 1, we use the method of [24] to show the difference about effective receptive fields of the same pixel of score map at the location of point *A* with and without the Elite Loss. We can conclude that the Elite Loss makes the effective receptive field more concentrated. This is beneficial to the pixel-wise classification task, since the surrounding noises are largely suppressed.

The contributions of this paper are listed as follows:

- We propose the Elite Loss in segmentation based text detection networks which have a regression step, by down-weighting the contributions of the in-box not-on-stoke pixels to improve the training performance.
- To demonstrate the effectiveness and the flexibility of Elite Loss, we design two specific forms of Elite Loss and evaluate them in the task of text detection.
- With the Elite loss integrated into the segmentation based text detector which has a regression step, we achieve state-of-the-art results on various datasets.

2. Related work

Task specific loss functions. For the general object detection, Focal Loss [25] is proposed to handle the class-imbalance problem by focusing hard examples and down-weight the easy examples. Differently, Elite Loss is to handle the imprecise labels of text instance's predicting units. It down-weights the not-on-stroke examples that are unnecessary or even may degrade the detector's performance. For the robust estimation task, Huber Loss [26] also down-weights the hard-to-learn samples, which is regarded as outliers for Huber Loss. We also consider these in-box hard samples outside the strokes would harm the performance. But Elite Loss is for classification loss functions while Huber Loss is for regression functions.

Scenetext detection. Scene Text Detection has been studied for a long time. Traditional text detection methods are mainly based on connected components, such as Stroke Width Transform

(SWT) [6] and Maximally Stable Extremal Regions (MSER) [27,28], or sliding-window [29,30] and use a bottom-up strategy, with complex post-processing steps.

In recent years, many deep learning based text detection methods are proposed. Some methods are evolved from general object detection methods, like SSD [9] or Faster RCNN [10]. Representative methods are TextBoxes [11], TextBoxes++ [12], RRD [14], SegLink [1], RRPN [5], R^{2}CNN [13]. These methods set some anchor boxes to detect objects, which is not suitable for detecting multi-oriented text instances, as discussed in [4]. Therefore they usually need complex designing to detect multi-oriented texts, leading to high computation complexity.

Other methods are based on FCN [16]. They usually implement the text detection by classifying each pixel into text or background, like [17,18]. But most of them need complex post-processing steps to locate each text instance. Wu et al. [31] introduce the border class and classify each pixel into three classes to reduce the complexity of post-processing steps. Deng et al. [32] predict the linking relationships between each pixel with its neighborhood pixels, apart from the two-class classification task. Liu et al. [20] design the Markov Clustering Network to group the segmented foreground pixels into text instances. Lyu et al. [19] predict position-sensitive segmentation maps rather than two-class maps. They also detect text corners and then combine the two tasks to get the final text boxes. Recently, methods like EAST [2] and DirectRegression [21,33] are proposed, which combine two-class pixel-wise classification and regression step to predict the corresponding bounding box for each positive pixel. The segmentation based text detection methods can detect multi-oriented texts better. The proposed Elite Loss is designed for the segmentation based methods with a regression step. It can alleviate the problem caused by the lack of fine-grained segmentation annotations, as mentioned in the introduction section.

3. Elite Loss for scene text detection

The *Elite Loss* is proposed for the segmentation based detectors, in which each pixel is a predicting unit. It is intended to force the training on the on-stroke predicting units and discard the in-box not-on-stroke predicting units which are unnecessary and may harm the detector. We first present the definition of Elite Loss, which is evolved from the existing classification loss function for text detection, in Section 3.1. Then Section 3.2 gives a discussion on the specific rules on distinguishing the predicting units that should be paid more attention and those should be down-weighted. Sections 3.3 and3.4 give two forms of Elite Loss based on different distinguishing rules.

3.1. Definition of Elite Loss

Elite Loss aims to re-weight the regular predicting units that involved in the segmentation loss computation process.

The current state-of-the-art segmentation based text detectors usually regard each predicting unit equally during training. Taking the cross-entropy loss as an example, the segmentation loss function of one image can be expressed as:

$$LI = \sum_{u=1}^{U_n} l(p_u, g_u = -1) + \sum_{t=1}^{T} \sum_{u=1}^{U_{t,p}} l(p_u, g_u = +1) \tag{1}$$

where T is the number of text objects, U_n is the number of predicting units that are matched to none of the text objects, $U_{t,\,p}$ is the number of predicting units that are matched to the text object indexed by t, $p_u \in [0,\,1]$ is the confidence score of the predicting unit u, which is predicted by the network, $g_u \in \{-1, +1\}$ is the class label for u, and $l(\cdot)$ is the cross-entropy loss.

As discussed above, we consider the in-box not-on-stroke predicting units of the text instance as less important ones and down-weight them to encourage the network to pay more attention to the elite on-stroke ones. By assigning a weight coefficient to each predicting units, we get the Elite Loss (or more exactly, Elite Cross Entropy Loss):

$$LI = \sum_{u=1}^{U_n} w_u l(p_u, g_u = -1) + \sum_{t=1}^{T} \sum_{u=1}^{U_{t,p}} w_u l(p_u, g_u = +1). \tag{2}$$

In Elite Loss, within the in-box regions, the elite predicting units should have higher weights than the non-elite ones. Moreover, the out-of-box predicting units should have full weight (w_u=1), because all these predicting units are correctly labeled in a reasonable way and they should be fully trained.

3.2. Selection of Elite predicting units

The text detection datasets usually have no fine-grained text stroke segmentation labels. Thus weak-supervised methods for distinguishing the text strokes and outside-stroke areas are needed. We can either directly define the text stroke regions based on heuristic rules, or locate them adaptively through the confidence on the output segmentation maps.

Hence, we propose two forms of the Elite Loss according to two different kinds of elite weight generation. One is the Adaptive Elite Loss and the other is the Heuristic Elite Loss.

3.3. Adaptive Elite Loss

Because the not-on-stroke predicting units usually share similarities with the background, they tend to have low confidence scores. Thus the confidence score of each predicting unit, output by the network during the training process, is a good indicator of elite or non-elite. Therefore, we design the Adaptive Elite Loss using the predicted confidence score as the weight w_u of each predicting unit's loss l_u:

$$w_u = \begin{cases} p_u^{\lambda} & \text{if } g_u = 1, \\ 1 & \text{otherwise.} \end{cases} \tag{3}$$

where $w_u \in [0,\,1]$ is the elite weight to be multiplied with l_u and λ controls the extent of down-weighting for the non-elite predicting units. The predicting units without being matched to any ground truth (i.e. the in-box not-on-stroke ones) have the full weight, for they have no regression target if classified as positive.

We can down-weight the contributions of non-elite predicting units through Eq. (3) directly. However, this operation does harm to the text instances of whom all the assigned predicting units showing bad confidence scores. To avoid this problem, we normalize the weights by the maximum value of the confidence score of all predicting units that are matched to the same text object:

$$p_u^* = \max_i (p_i | Target(u) = Target(u_i)), \tag{4}$$

$$w_u = \begin{cases} (\frac{p_u}{p_u^*})^{\lambda} & \text{if } g_u = 1, \\ 1 & \text{otherwise.} \end{cases} \tag{5}$$

where *Target*(x) represents the index of text box which the predicting unit x is assigned to. Eq. (5) guarantees that the elite predicting units have large enough weight.

Moreover, when the maximum confidence score of all predicting units for some text instance is too low, the confidence score is less relevant to the “eliteness” of these predicting units. To guarantee that that text object is not lost, we set a barrier based on Eq. (5):

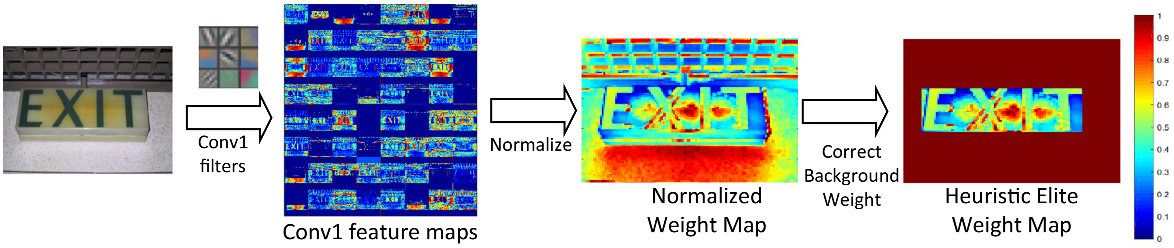


Fig. 2. The generation process of the heuristic form of Elite Loss.

$$w_u = \begin{cases} (\frac{p_u}{p_u^*})^\lambda & \text{if } g_u = 1,\ p_u^* > \theta\ , \\ \theta & \text{if } g_u = 1,\ p_u^* <= \theta, \\ 1 & \text{otherwise.} \end{cases} \tag{6}$$

where $\theta \in [0,\ 1]$ is the barrier threshold. Text objects with maximum predicting unit confidence less than θ have all predicting units with elite weight of θ. The reason why we set their weights with θ rather than 1 is to keep the distribution of focused samples dominated by elite predicting units, which usually share the same characteristics.

Discussion. The Adaptive Elite Loss does not perfectly locate the stroke areas. Intuitively, as discussed above, the pixels on the strokes of the text share the same characteristics. Only these pixels should be classified as elite predicting units. Moreover, the pixels on the in-box background regions, i.e. the not-on-stroke pixels, are to some extent easily-confused with that on the out-box background regions, so it should be classified as background. However, on the other hand, this rule might not be the best choice. Training the detector with some background pixels near the strokes may add the robustness of the model as a form of data augmentation. During the training process, more pixels near the strokes should be easy and easy to train, for they share similar receptive fields. In Adaptive Elite Loss, these pixels' weights are adaptively up-scaled during the training process. Detailed experimental results in Section 5.2 show that the background pixels around the strokes indeed improve the performance.

3.4. Heuristic Elite Loss

Matthew D. Zeiler [34] demonstrated that the lower layers of the convolutional neural network respond to the low-level features like edge/color conjunctions or corners. Based on this popular work, we design the Heuristic Elite Loss utilizing the output feature map of *conv1* layer of ResNet-50, donating C. The generation process of Heuristic Elite Loss is shown in Fig. 2.

Specifically, we conduct the following steps in the normalization stage to transform the multi-channel feature maps into the single-channel elite weight map, which are formalized as Eqs. (7)–(9). Firstly, we normalize each channel of the feature map to the range of [0,1], by dividing the maximum value of the corresponding channel (Eq. (7)). Then we generate the raw weight map by calculating the maximum value at each location across all channels. We re-scale the value of raw weight map by mapping the range [0.5, 1] to [0, 1] (Eq. (8)). Finally, we set the weight of out-of-the-box pixels as 1 for we only down-weight the in-box not-on-the-stroke pixels (Eq. (9)). The formulas of this procedure are:

$$C^*_{c,i,j} = Sigmoid\left(\frac{C_{c,i,j}}{\max_{i,j} C_{c,i,j}} - 0.5\right), \tag{7}$$

$$N_{i,j} = 2 \cdot Relu\left(\max_c C^*_{c,i,j} - 0.5\right), \tag{8}$$

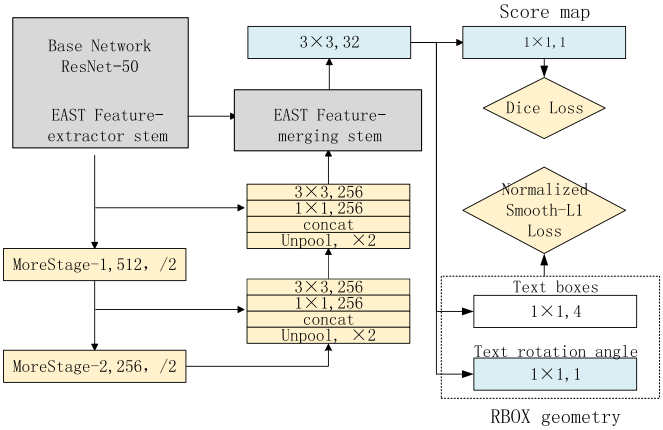


Fig. 3. The architecture of EAST+.

$$w_{i,j} = \begin{cases} N_{i,j} & \text{if } g_{i,j} = 1, \\ 1 & \text{otherwise.} \end{cases} \tag{9}$$

where C^* is channel-wise normalized feature map of C, N is the generated elite weight map, *Sigmoid* and *Relu* is the sigmoid and rectified linear operation of the neural network, and $w_{i,\ j}$ is the elite weight.

Fig. 2 shows that the elite map generated using this heuristic method generally reflects the phenomenon that the pixels on the strokes of the text have relatively larger weights than the pixels on the background in the bounding boxes, though the weight map is not perfectly consistent with the text texture and has large values at some in-box background locations.

4. Elite Loss on segmentation based text detector

We choose EAST [2] as the representative method of segmentation based text detectors and demonstrate the effectiveness of Elite Loss. EAST adopts FCN to generate a pixel-level text score map representing the presence of text and geometry maps encoding the word 's bounding boxes. In EAST, the pixels in the shrunken bounding boxes are regarded as foreground. As shown in Fig. 3 three modifications are conducted to enhance the original implementation and we call the new baseline as *EAST+*. Firstly, we choose ResNet-50 [35] as the base network. Secondly, when evaluating on the image with large-input size (i.e., 768 × 768 in the experiment section), we put 2 additional downsampling modules and 2 upsampling modules after the final stage of the base network to enlarge the receptive field. Thirdly, we adopt the dice loss instead of the balanced cross-entropy loss to train the text classifier, and further, we replace it with the object-size balanced dice loss. Finally, we utilize smoothed-L1 loss with a normalization term to regress the RBOX.

The Elite Loss can be easily integrated into Dice Loss. Dice Loss has the form of:

$$DL = 1 - \frac{\sum_i^N p_i g_i}{\sum_i^N p_i^2 + g_i^2} \tag{10}$$

We can find that Dice Loss also treats all the positive predicting units equally. To adjust the contribution of each predicting unit dynamically with their contributions to the detector, we use the Elite Loss to re-weight the units in Dice Loss, which we name as Elite Dice Loss:

$$DL = 1 - \frac{\sum_i^N p_i g_i w_i}{\sum_i^N p_i^2 w_i + g_i^2 w_i} \tag{11}$$

where w_i is the same as in Eq. (2)

Training details. We train EAST+ using ADAM optimizer over 4 GPUs for 16 epochs. We randomly crop image patches of fix size to form a larger mini-batch. The patch size is set differently according to the different text sizes of each dataset. And the learning rate starts from 1e-4 and it is multiplied by 0.94 after every 8 epochs. Besides the multi-scale data augmentation, we do not adopt any other form of data augmentation strategies unless noted.

The Elite Loss assigns a scaling factor for the loss of every positive predicting unit. It practically takes effect when the network is in the back-propagation stage. The training loss should not back-propagate through the computed weight, because the training process encourages to down-scale every unit's weight.

5. Experiment

In this section, we do extensive experiments on three public benchmark datasets to verify the effectiveness of our Elite Loss.

5.1. Datasets

The *ICDAR 2015* [36] dataset is from the Challenge 4 (Incidental Scene Text challenge) of the ICDAR 2015 Robust Reading Competition. The dataset includes 1000 images for training and 500 images for testing, with text labeled at the word level. Most text objects are rotated and some are blurred. The results are evaluated using the evaluation tools of ICDAR 2015 [36].

The *COCO-Text* [37] dataset is a large text detection dataset, consisting 43,686 images for training and 20,000 images for testing. All images are from MS-COCO dataset. The text objects vary in orientation and appearance with the background cluttered. We report the standard evaluation result of COCO-Text, including performance on English and non-English as well as legal and illegal texts.

The *MSRA-TD500* [38] dataset is the first standard dataset that focuses on oriented text. It has 300 images for training and 200 images for testing. All text objects are annotated at line-level, which means each annotated bounding box covers multiple words in the same line. The evaluation protocol is described in [38].

5.2. Ablation study

In Section 3, we give two forms of Elite Loss, which are Adaptive Elite Loss (AEL) and Heuristic Elite Loss (HEL). Table 1 compares these two forms of Elite Loss and some variations of them on MSRA-TD500. The texts of MSRA-TD500 are multi-oriented. Since MSRA-TD500's training set is too small to optimize the network, we add 400 images from HUST-TR400 dataset [39] into the training data, which is the common practice [2]. We train the detector with 512 × 512 sampled image crops and test it using the input image with the short side resized to 512. Other settings are the same as that in Section 4. The experimental results of this section are reported on the test set of the MSRA-TD500 benchmark.

Table 1
Comparing different forms of Elite Loss on MSRA-TD500. AEL: Adaptive Elite Loss. HEL: Heuristic Elite Loss.

Method	Recall	Precision	F-score
EAST+	0.6980	0.7910	0.7410
AEL+EAST+	0.7210	0.8330	0.7730
HEL+EAST+	0.7137	0.8093	0.7586
Reverse AEL + EAST+	0.6990	0.7830	0.7390
Dirichlet AEL + EAST+	0.7124	0.8348	0.7687
Canny HEL+EAST+	0.6780	0.7910	0.7300

Table 2
Varying λ and θ of Adaptive Elite Loss on MSRA-TD500.

θ	λ	Recall	Precision	F-score
0.001	2	0.6876	0.8094	0.7436
0.1	2	0.6930	0.8390	0.7590
0.3	2	0.6936	0.8292	0.7553
0.1	1	0.7210	0.8330	0.7730
0.1	0.5	0.7110	0.8200	0.7620
1	0	0.6980	0.7910	0.7410

Table 1 shows that both adaptive and heuristic forms of Elite Loss can improve the EAST+ by a significant margin. The AEL is much better than the HEL. We think there are two main causes. Firstly, the HEL is actually based on the hand-craft rule, which usually is a sub-optimal solution. For example, the weight map of HEL has large values on some smooth in-box regions in Fig. 2. Secondly, the AEL locates the elite predicting units in an adaptive way based on the output confidence score maps. This strategy is more reasonable and more promising to achieve better results. Thus we choose to use the *AEL* in the following experiments.

In Table 1, we also tried the contrary strategy that is aimed to focus on classifying the hard predicting units that are outside the strokes correctly, and down-weight the easy examples. This is similar to the OHEM [22] method which pays more attention to hard examples. By replacing the p with $1 - p$ and p^* with $1 - p^*$ in Eq. (6), we get the Reverse Adaptive Elite Loss. The precision of EAST+ with Reverse Elite Loss drops 0.01 point, compared with the EAST+. This demonstrates that forcing the not-on-stroke predicting units to be correctly classified comes with false alarms on the background. The Dirichlet Adaptive Elite Loss, which is the binarization of Adaptive Elite Loss with a threshold of 0.5, also leads to an improvement over the baseline EAST+. However, the recall of Dirichlet Adaptive Elite Loss is lower than the Adaptive Elite Loss. Thus the form of soft weights of Adaptive Elite Loss is more suitable to represent the importance of each pixel than the form of hard weights of the Dirichlet version. We also use the computed edge detection results by Canny edge detector as the predicting units' weight to form Canny Heuristic Elite Loss. According to Table 1, the result degrades seriously in both precision and recall, because only using the edge pixels limits the effective number of positive samples.

We also explore the effects of different settings of λ and θ of Eq. (6) on AEL's performance, as shown in Table 2. The hyper-parameter λ controls the extent of the down-weighting for the non-elite predicting units. When $\lambda = 0$, Elite Loss generates full-weight for all predicting units, which is the same as the original loss functions. With increasing λ, less elite predicting units are focused on, and non-elite predicting units are further down-weighted.

The hyper-parameter θ sets a barrier for those text instances with all corresponding predicting units hard to be trained. A considerably small value means that the "hard text instances" may be less considered, and a much higher value causes that, the elite loss will have less impact.

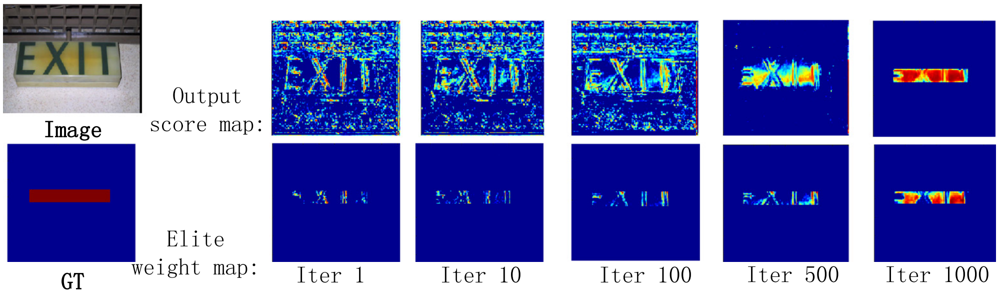


Fig. 4. The evolution of output score maps and elite weight maps during the training process of EAST+. The network tends to have high activations on stroke pixels.

From Table 2, it can be concluded that without the AEL (*i.e.* $\lambda = 0, \theta = 1$), the baseline EAST+ only achieves F-score of 0.741. With varying θ and λ, 0.032 points gain in F-score is achieved. It can be seen that with λ changing from 1 to 2, the recall drops sharply because less positive samples are considered in the training process. With θ changing from 0.1 to 0.001, the precision drops by 0.0296. The reason may be that the maximum weight for the predicting units of hard text instances is too high according to Eq. (6). Moreover, with θ changing from 0.1 to 0.3, the precision drops slightly when more predicting units of the hard text instances are considered.

Fig. 4 shows the evolution of the elite weight map of AEL during the training process. The train set consists of only one image to demonstrate the effect of AEL during training. We can conclude from Fig. 4 that the AEL focus the network's attention to the pixels that are easier to learn, which potentially helps to train a better text detector.

5.3. Comparison with state-of-the-arts

We evaluate the Adaptive-Elite-Loss-equipped EAST+ on various benchmarks to demonstrate its superiority. We train our model on the training set of each benchmark, and test the model on each corresponding test set. In this section, we use the best hyper-parameter setting of λ and θ tuned in Section 5.2, and keep them fixed through all the following experiments. That is $\lambda = 1$ and $\theta = 0.1$.

EAST+ runs at around 40 ms per image on an NVIDIA Titan X (Pascal) GPU with the input image size of 512 × 512, which is efficient. Since Elite Loss takes effect at training time, it does not cost additional time for testing the algorithm.

For ICDAR 2015, the network is trained on 512 × 512 resolution on the combination of ICDAR2015 and ICDAR2013 [41] training set, which has 1229 images in total. We evaluate the network using the image's original size. As seen from Table 3, the EAST+ with Elite Loss achieves the state-of-the-art performance with the F-score of 0.8228. EAST+ with Elite Loss outperforms the original EAST [2] and DirectRegression [4]. With the model pretrained on 7.2k images from MLT[1] dataset, we outperform FSTN [15], RRD [14], Lyu et al. [19], and Textboxes++ [12]. These methods use a subset of 160k images or all of the 860k images from SynthText [42] dataset for pretraining. With the multi-scale testing, we achieve the new state-of-the-art of 85.51 F-score. We note that the Elite Loss down-weights the effects of the in-box not-on-stroke predicting units, which are easily confused with the background. This leads the text detector to have high precision, which is important for the multi-scale testing to improve the recall with the side-effect of precision drop.

Table 3
Results on ICDAR 2015. EAST*+: EAST+ Pretrained using MLT dataset. MS: multi-scale testing. AEL: Adaptive Elite Loss.

Method	Recall	Precision	F-score
AEL + EAST*+(MS)	0.8209	0.8922	0.8551
AEL + EAST*+	0.8040	0.8826	0.8415
AEL + EAST+	0.7670	0.8845	0.8228
EAST+	0.8035	0.8291	0.8161
Lyu et al. [19] (MS)	0.7970	0.8950	0.8430
FSTN [15]	0.8000	0.8860	0.8410
RRD [14] (MS)	0.8000	0.8800	0.8380
TextBoxes++ [12](MS)	0.7850	0.878	0.829
R^2CNN [13] (MS)	0.7968	0.8562	0.8254
He et al. [21] (MS)	0.8000	0.8500	0.8200
DirectRegression [4] (MS)	0.8200	0.8000	0.8100
EAST [2] (MS)	0.7833	0.8327	0.8072
WordSup [40] (MS)	0.7703	0.7933	0.7816
RRPN [5]	0.8217	0.7323	0.7744
SSTD [33]	0.7300	0.8000	0.7700
MCN [20]	0.8000	0.7200	0.7600
SegLink [1]	0.7310	0.7680	0.7500
Yao et al. [3] (MS)	0.5869	0.7226	0.6477
Zhang et al. [17] (MS)	0.4309	0.7081	0.5358

Table 4
Results on COCO-Text. AEL: Adaptive Elite Loss.

Algorithm	Recall	Precision	F-score
AEL + EAST+	0.331	0.5957	0.4253
EAST+	0.340	0.5175	0.4103
TextBoxes++ [12] (MS)	0.5670	0.6087	0.5872
Lyu et al. [19] (MS)	0.324	0.619	0.425
EAST [2]	0.324	0.5039	0.3945
SSTD [33]	0.310	0.4600	0.3700
WordSup [40] (MS)	0.309	0.4520	0.3680
Yao et al. [3] (MS)	0.271	0.4323	0.3331
Baselines from [37]			
A	0.233	0.8378	0.3648
B	0.107	0.8973	0.1914
C	0.047	0.1856	0.0747

For COCO-Text, the network is trained on 512 × 512 resolution and test with the image's short length resized to 512. Note that we only use the legal text labels for training. As shown in Table 4, EAST+ outperforms the original EAST [2] by a large margin. The Adaptive Elite Loss further improves the F-score to 0.4253, which is comparable to the state-of-the-art. This indicates that Elite Loss enables the detector to focus on learning better features for the elite predicting units on the large and cluttered training dataset. Textboxes++ [12] outperforms other methods by a large margin. The reason may be that text instances are labeled as axis-aligned bounding boxes in COCO-Text, even for the highly inclined texts. Thus the general-object-detection-based method Textboxes++ performs better.

For MSRA-TD500, we adopt the same setting as in Section 5.2 except that we train on image patches of 768 × 768 and test on images with short side resized to 768. Table 5 shows that the larger test size only improves less than 0.01 F-score

[1] http://rrc.cvc.uab.es/?ch=8.

Table 5
Results on MSRA-TD500. AEL: Adaptive Elite Loss. MS: multi-scale testing.

Algorithm	Recall	Precision	F-score
AEL + EAST+ (MS)	0.771	0.873	0.819
AEL + EAST+	0.699	0.887	0.782
EAST+	0.726	0.809	0.765
He et al. [21] (MS)	0.910	0.810	0.860
RRD [14] (MS)	0.730	0.870	0.790
MCN [20]	0.79	0.88	0.83
FSTN [15]	0.771	0.876	0.820
Lyu et al. [19] (MS)	0.762	0.876	0.815
SegLink [1]	0.860	0.700	0.770
EAST [2] (MS)	0.873	0.674	0.761
Yao et al. [3]	0.753	0.765	0.759
RRPN [5]	0.820	0.680	0.740
He et al. [4] (MS)	0.770	0.700	0.740
Zhang et al. [17] (MS)	0.67	0.83	0.74
Yin et al. [43]	0.63	0.81	0.71

Table 6
Results of Elite Loss applied on DirectRegression [4] on MSRA-TD500. DirectRegression-ReIm: The DirectRegression implemented by ourself. AEL: Adaptive Elite Loss. HEL: Heuristic Elite Loss. MS: multi-scale testing.

Method	Recall	Precision	F-score
DirectRegression (MS)	0.7700	0.7000	0.7400
DirectRegression-ReIm (MS)	0.7913	0.7159	0.7517
HEL + DirectRegression-ReIm (MS)	0.7698	0.7593	0.7646
AEL + DirectRegression-ReIm (MS)	0.7793	0.7698	0.7718

(AEL+EAST+) compared to the results in Table 1. It can be seen Elite Loss contributes 0.017 on F-score. Moreover, with the high precision brought by the Elite Loss, multi-scale testing achieves 0.819 on F-score, which is comparable with the current state-of-the-art. Note that except random cropping, we do not adopt any other forms of data augmentation or model pre-training on other text detection datasets, which is used in [15,20,21].

Tables 3–5 show that our method achieves the best precision rate over all methods. This is achieved by down-weight of the contributions of hard not-on-stroke predicting units to the training loss, for these units are easily confused by the units in the background. Furthermore, since for each text instance we guarantee at least one predicting unit has the full-weight, the recall rate is less influenced. Moreover, this characteristic of Elite Loss makes the detector achieve the new state-of-the-art with multi-scale testing.

5.4. Experiments onother segmentation based methods

DirectRegression [4] is another segmentation based method with a regression step. The main difference between EAST [2] and DirectRegression is that EAST predicts text instances of all possible scales using the neural network, while DirectRegression uses the network to predict text instances of only one scale (texts with shorter edge of around 32 pixels) and adopts the multi-scale testing to detect text instances of all scales. Since the code of DirectRegression [4] has not been released, we implement DirectRegression by ourself. Then we apply Elite Loss on it to validate the generality of Elite Loss. As shown in Table 6, both forms of Elite Loss (AEL and HEL) improve the performance of DirectRegression. Again, AEL is better. This experiment shows that Elite Loss has good generality for other segmentation based text detection methods which have the regression step.

Discussion. For those segmentation based methods without a regression step, they have to design other procedure to identify the text instances from each other, including predicting the border pixels [14], predicting the linking relationship with the neighbors of each pixel [32], and adopting other hand-crafted post-processing steps [17]. But these methods require that all in-box pixels must be classified as foreground to form the connected areas to detect each text instance integrally. It can be concluded that for these methods none of the pixels should be down-weighted in the classification loss. Therefore, Elite Loss cannot be directly applied to these segmentation methods which do not involve a regression step.

6. Conclusion

In this paper, we propose the Elite Loss to address the problem that current segmentation labels are unsuitable for the network to learn, in segmentation based text detection methods. We found that for the segmentation based methods which have a regression step, each pixel location on the output feature map is an independent predicting unit. Instead of considering all the predicting units equally, the Elite Loss reweights them with their contributions to the detector's performance. It forces the detector to learn better features for the elite predicting units, which are usually on the strokes and capture the instinct characteristics of text regions better. The Elite Loss is flexible and effective and it can be easily integrated into current popular text detectors. We give two forms of Elite Loss, which are the heuristic form and the adaptive form. Extensive experiments on various datasets demonstrate the effectiveness of the Adaptive Elite Loss.

Acknowledgments

This work was supported by Natural Science Foundation of China under Grants 61772527, 61806200, and 61876086.

References

[1] B. Shi, X. Bai, S. Belongie, Detecting oriented text in natural images by linking segments, in: Proceedings of the CVPR, IEEE, 2017, pp. 3482–3490.

[2] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, J. Liang, East: an efficient and accurate scene text detector, in: Proceedings of the CVPR, IEEE, 2017, pp. 2642–2651.

[3] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, Z. Cao, Scene text detection via holistic, multi-channel prediction, CoRR. arXiv:1606.09002.

[4] W. He, X.Y. Zhang, F. Yin, C.L. Liu, Deep direct regression for multi-oriented scene text detection, in: Proceedings of the ICCV, IEEE, 2017, pp. 745–753.

[5] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, X. Xue, Arbitrary-oriented scene text detection via rotation proposals, IEEE TMM 20 (11) (2018) 3111–3122.

[6] B. Epshtein, E. Ofek, Y. Wexler, Detecting text in natural scenes with stroke width transform, in: Proceedings of the ICCV, IEEE, 2010, pp. 2963–2970.

[7] L. Neumann, J. Matas, Real-time scene text localization and recognition, in: Proceedings of the CVPR, IEEE, 2012, pp. 3538–3545.

[8] M. Jaderberg, A. Vedaldi, A. Zisserman, Deep features for text spotting, in: Proceedings of the ECCV, IEEE, 2014, pp. 512–528.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, SSD: single shot multibox detector, in: Proceedings of the ECCV, Springer, 2016, pp. 21–37.

[10] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, in: Proceedings of the NIPS, 2015, pp. 91–99.

[11] M. Liao, B. Shi, X. Bai, X. Wang, W. Liu, Textboxes: a fast text detector with a single deep neural network, in: Proceedings of the AAAI, 2017.

[12] M. Liao, B. Shi, X. Bai, Textboxes++: a single-shot oriented scene text detector, IEEE TIP 27 (8) (2018) 3676–3690.

[13] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, Z. Luo, R2CNN: rotational region CNN for orientation robust scene text detection, CoRR. arXiv:1706.09579.

[14] M. Liao, Z. Zhu, B. Shi, G.s. Xia, X. Bai, Rotation-sensitive regression for oriented scene text detection, in: Proceedings of the CVPR, IEEE, 2018, pp. 5909–5918.

[15] Y. Dai, Z. Huang, Y. Gao, Y. Xu, K. Chen, J. Guo, W. Qiu, Fused text segmentation networks for multi-oriented scene text detection, in: 2018 24th International Conference on Pattern Recognition (ICPR), IEEE, 2018, pp. 3604–3609.

[16] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE TPAMI 39 (4) (2017) 640–651.

[17] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, X. Bai, Multi-oriented text detection with fully convolutional networks, in: Proceedings of the CVPR, IEEE, 2016, pp. 4159–4167.

[18] T. He, W. Huang, Y. Qiao, J. Yao, Text-attentional convolutional neural network for scene text detection, TIP 25 (6) (2016) 2529–2541.

[19] P. Lyu, C. Yao, W. Wu, S. Yan, X. Bai, Multi-oriented scene text detection via corner localization and region segmentation, Proceedings of the CVPR, IEEE (2018) 7553–7563.

[20] Z. Liu, G. Lin, S. Yang, J. Feng, W. Lin, W. Ling Goh, Learning markov clustering networks for scene text detection, in: Proceedings of the CVPR, IEEE, 2018, pp. 6936–6944.

[21] W. He, X.Y. Zhang, F. Yin, C.L. Liu, Multi-oriented and multi-lingual scene text detection with direct regression, IEEE TIP 27 (11) (2018) 5406–5419.

[22] A. Shrivastava, A. Gupta, R. Girshick, Training region-based object detectors with online hard example mining, in: Proceedings of the CVPR, IEEE, 2016, pp. 761–769.

[23] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the CVPR, IEEE, 2014, pp. 580–587.

[24] J. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: the all convolutional net, in: Proceedings of the ICLR (workshop track),

[25] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in: Proceedings of the CVPR, IEEE, 2017, pp. 2980–2988.

[26] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning, Vol. 1, No. 10, Springer series in statistics, 2001.

[27] L. Neumann, J. Matas, A method for text localization and recognition in real–world images, in: Proceedings of the ACCV, Springer, 2010, pp. 770–783.

[28] X.C. Yin, X. Yin, K. Huang, H.W. Hao, Robust text detection in natural scene images, TPAMI 36 (5) (2014) 970–983.

[29] L. Neumann, J. Matas, Scene text localization and recognition with oriented stroke detection, in: Proceedings of the ICCV, IEEE, 2013, pp. 97–104.

[30] T. Wang, D.J. Wu, A. Coates, A.Y. Ng, End-to-end text recognition with convolutional neural networks, in: Proceedings of the ICPR, IEEE, 2012, pp. 3304–3308.

[31] Y. Wu, P. Natarajan, Self-organized text detection with minimal post-processing via border learning, in: Proceedings of the CVPR, IEEE, 2017, pp. 5000–5009.

[32] D. Deng, H. Liu, X. Li, D. Cai, Pixellink: Detecting scene text via instance segmentation, in: Proceedings of AAAI, 2018.

[33] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, X. Li, Single shot text detector with regional attention, in: Proceedings of the CVPR, IEEE, 2017, pp. 3047–3055.

[34] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Proceedings of the ECCV, 2014, pp. 818–833.

[35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the CVPR, IEEE, 2016, pp. 770–778.

[36] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V.R. Chandrasekhar, S. Lu, et al., Icdar 2015 competition on robust reading, in: Proceedings of the ICDAR, IEEE, 2015, pp. 1156–1160.

[37] A. Veit, T. Matera, L. Neumann, J. Matas, S. Belongie, Coco-text: Dataset and Benchmark for Text Detection and Recognition in Natural Images, arXiv:1601.07140.

[38] C. Yao, X. Bai, W. Liu, Y. Ma, Z. Tu, Detecting texts of arbitrary orientations in natural images, in: Proceedings of the CVPR, IEEE, 2012, pp. 1083–1090.

[39] C. Yao, X. Bai, W. Liu, A unified framework for multioriented text detection and recognition, TIP 23 (11) (2014) 4737–4749.

[40] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, E. Ding, Wordsup: exploiting word annotations for character based text detection, in: Proceedings of the ICCV, IEEE, 2017.

[41] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L.G.i. Bigorda, S.R. Mestre, J. Mas, D.F. Mota, J.A. Almazan, L.P. de las Heras, Icdar 2013 robust reading competition, in: Proceedings of the ICDAR, IEEE, 2013, pp. 1484–1493.

[42] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic data for text localisation in natural images, in: Proceedings of the CVPR, IEEE, 2016, pp. 2315–2324.

[43] X.C. Yin, W.Y. Pei, J. Zhang, H.W. Hao, Multi-orientation scene text detection with adaptive clustering, TPAMI 37 (9) (2015) 1930–1937.

Xu Zhao received the B.E. degree in 2014 from Dalian University of Technology of China. He is now pursuing a Ph.D. degree on pattern recognition and intelligence systems as a student at the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, since 2014. His research interests include object detection, scene text detection, image and video processing, and intelligent video surveillance.

Chaoyang Zhao received the B.E. degree and the M.S. degree in 2009 and 2012 respectively from University of Electronic Science and Technology of China. He received the Ph.D. degree in pattern recognition and intelligence systems from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, in 2016. He is currently an Assistant Professor in National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include object detection, image and video processing and intelligent video surveillance.

Haiyun Guo received her B.E. degree from Wuhan University in 2013 and the Ph.D. degree in pattern recognition and intelligence systems from the Institute of Automation, University of Chinese Academy of Sciences, in 2018. She is currently an assistant researcher in the National Laboratory of Pattern Recognition, Institute of Automation Chinese Academy of Sciences. Her current research interests include pattern recognition and machine learning, image and video processing, and intelligent video surveillance.

Yousong Zhu received the B.E. degree from Central South University, Changsha, China, in 2014. He is currently pursuing the Ph.D. degree in pattern recognition and intelligence systems with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His current research interests include object detection, video object detection, pattern recognition and machine learning, and intelligent video surveillance.

Ming Tang received the B.S. degree in computer science and engineering and M.S. degree in artificial intelligence from Zhejiang University, Hangzhou, China, in 1984 and 1987, respectively, and the Ph.D. degree in pattern recognition and intelligent system from the Chinese Academy of Sciences, Beijing, China, in 2002. He is currently a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His current research interests include computer vision and machine learning.

Jinqiao Wang received the B.E. degree in 2001 from Hebei University of Technology, China, and the M.S. degree in 2004 from Tianjin University, China. He received the Ph.D. degree in pattern recognition and intelligence systems from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, in 2008. He is currently a Professor with Chinese Academy of Sciences. His research interests include pattern recognition and machine learning, image and video processing, mobile multimedia, and intelligent video surveillance.