2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Focus On Scene Text Using Deep Reinforcement Learning

Haobin Wang, Shuangping Huang* and Lianwen Jin
School of Electronic and Information Engineering, South China University of Technology
*Corresponding author Email: huangshuangping@gmail.com

***Abstract*—Scene text detection has been attracting increasing interests in recent years and a rich body of approaches has been proposed. These previous works of detecting scene text have been dominated by region proposals based approaches, which always generate too many text candidates relative to the number of ground truth bounding boxes. Only a few of those candidates are output as true predictions, and most of the other is fruitlessly involved in regression or classification predictions that consume a great amount of time and storage. Thus emerges the problem of low efficiency of generating text candidates. To address the issue, we propose a method for focusing on scene text gradually guided by an active model. The model allows an agent to take the whole image as the only region proposal in each episode when locating text and therefore significantly reduces the region proposals needed. The agent is trained by deep reinforcement strategy to learn how to estimate future returns of given states and sequentially make decisions to find scene text. Considering the characteristics of scene text, we additionally propose a flexible action scheme and a new reward scheme together with lazy punishment. The experiments on the ICDAR 2013 dataset shows that the proposed method achieve a promising performance while using region proposals as few as the ground truth bounding boxes.**

I. Introduction

Scene text is a kind of visual objects containing abundant information, and the therein high level semantics and environmental description clues play an important role in several practical applications such as camera instant translation [1], assistive smartphone applications for visually impaired people [2] and autonomous navigation, etc. Therefore, reading scene text has obtained rapidly increased attention. Scene text detection, as an important sub-task of understanding scene text, has been focused on by recent works [3]–[6]. Different from OCR text, detecting text in the wild is more challenging due to extremely complicated background, diverse text patterns, and so on.

Traditional methods of scene text detection are usually implemented with the help of various hand-crafted features that represent some specific properties of scene text. Combining with these manual features, sliding-window based methods [4], [7]–[9] and connected-components methods [10]–[13] are proposed. In addition, owing to the superior performances of convolution neural networks(CNNs) achieved in the field of computer vision, there has also emerged a developing trend of adopting CNN on scene text detection task. These methods directly learn effective features from training data rather than extract hand-crafted features based on fixed rules. Whether or not deep CNNs are involved in, all the aforementioned methods follow a similar process of generating text candidates according to various features and classifying the true positive predictions from those candidates.

Actually, the idea behind region proposal based methods is naive and the corresponding works [3], [6], [14]–[16] achieved state-of-the-art scene text detection performance. However, most of these existing methods generate too many region candidates, less to hundreds, more to thousands. Those region candidates either contain little texts, or cover the same texts, or keep a low level of intersection-over-union (IoU) towards the ground truth texts, all of which are involved in the subsequent regression or classification process. This leads to not only a lot of consumption of time or memory, but also vagueness of true predictions. Thus emerges the problem of low efficiency of generating text candidates.

To tackle the problem, we propose a method that locates scene text in an iterative manner. We cast this task as a Markov Decision Process (MDP) and adopt deep reinforcement learning to train an agent. In each episode, The agent starts from the only region proposal that is the whole image in our case. It iteratively takes the actions and tries to push the scene text region to be focused on in the following time steps. Finally, the agent succeeds to locate the target text after a few sequential decisions. In fact, the proposed method refers to the way human locate targets in sight, i.e., human visual psychological focusing mechanism, as described in [17]. Specifically, eye movements are iteratively made to fixate the more precise focus area from given search scenes during the process of the target acquisition model. After several eye movements with computing the activation of the search scene and adjusting fixation dynamically, the target is found and the process stops. From the above description, the entire image is treated as the only candidate area that may contain text, being gradually focused towards to text region through a searching path. The searching path consists of sequential actions and the resulting sub-regions. Thus, only one region proposal is used to locate a text.

Besides, considering the characteristics of scene text that varies in size and aspect ratio greatly, we also design a flexible action set combining attention mechanism, and a new dense reward scheme combining lazy punishment to further improve the scene text detection performance.

The rest of this paper is structured as follows: Section II briefly introduces the related works of both scene text detection and deep reinforcement learning. In Section III we

978-1-5386-3787-6/18/$31.00 ©2018 IEEE

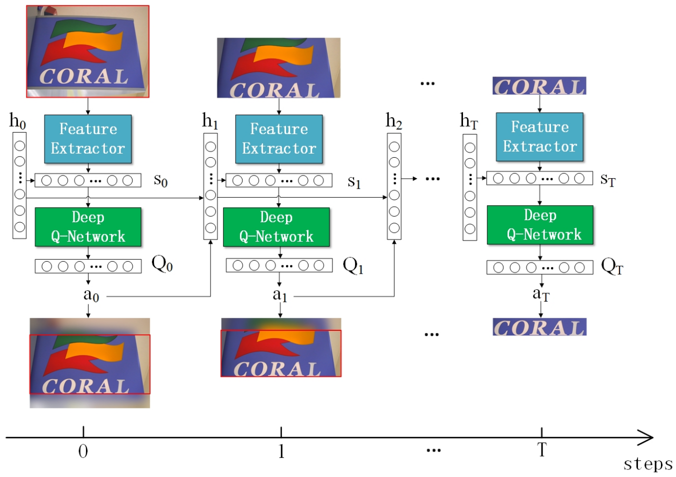

Fig. 1. The pipeline of each episode in the proposed method. In each time step t, the state representation s_t is composed by the feature vector extracted by the feature extractor and the history vector h_t recording all the actions in the current episode. When the target text is found or the maximum steps are run out, the episode stops.

describe the proposed method in details, including the flexible action scheme and the dense reward scheme combining the lazy punishment. In Section IV, the training algorithm and the network architectures are presented in details. Experimental evaluation is given in Section V. The paper is concluded in Section VI.

II. Related Work

A. scene text detection

Scene text detection has been an active research topic in recent years. Conventional methods depend on many hand-crafted features including color features [18]–[20], edge features [21], [22], texture features [23]–[26]. Based on various manual features, connected-components methods [10]–[13] and sliding-window methods [4], [7]–[9] have been proposed, which are the traditional mainstream methods. Connected-components methods filter text pixels out and group them into several components like strokes, and thus generate many text candidates. Among these methods, SWT [10] and MSER [11] are the representative ones. Sliding-window methods search for scene text by moving a multi-scale window through every entire image, and distinguish the windows if they contain text by a classifier. Recently, since deep learning based approaches achieve better results on object detection, there are also many scene text detection methods [3], [6], [14]–[16] based on recent advanced object detection systems such as Faster-RCNN [27] and SSD [28], and these methods advanced scene text detection technique. Compared with the conventional approaches, deep learning based methods learn the effective features by the networks and generate region proposals with relatively higher quality. However, the common drawback of the aforementioned methods is that they generate too many text candidates relative to the number of ground truth boxes, resulting in a low efficiency of generating region candidates.

B. deep reinforcement learning

Reinforcement learning trains an agent based on trail and error. Following the framework of Markov Decision Process, the agent first interacts with the environment and observes the current state, then selects an action considering the state and receives a reward as feedback from the environment. With the goal of maximizing the expected future return, the agent learns by itself with these reward signals.

Recently, there is a growing interest in reinforcement learning methods combined with deep neural networks, and these methods have been applied in many research areas including games [29]–[35] and Robotics [36]–[38]. One of the mainstream deep reinforcement learning methods is Deep Q-network(DQN) that was first proposed in [29]. The DQN has a goal of learning the optimal action-value function approximated by deep neural network, and is often used in the tasks whose action spaces are discrete.

The works supporting our idea are [39] and [40], both of which locate general objects using DQN. Caicedo et al. [39] provided an action set for the agent, which can deform the bounding boxes but easily turns the search process into a loop and leads to a huge search space, while Miriam Bellver et al. [40] adopted a top-down search mechanism with a fixed representation, which is unable to locate the objects with various shapes like scene text. In this paper, to adapt to the characteristics of scene text, we propose a different set of actions with a flexible attention mechanism, which focuses on the targets gradually in a top-down manner while deforming the visible regions simultaneously.

III. Focus on Scene Text Following the MDP Framework

We cast the scene text detection task as a Markov Decision Process(MDP), since the framework is suitable to model a discrete time sequential decision making process. In this task, we consider each image as the environment that the agent interacts with, and the aim of the agent is to focus on scene text in each image. During training, to learn to locate scene text, the agent stimulates to search for certain episodes in training images. In each episode, the agent makes decisions with the given state representations, and receives reward feedbacks following the defined reward scheme. During testing, the agent follows the learnt policy to search the texts for several episodes in each image.

The pipeline of each episode is shown in Figure 1. In an episode, the initial state s_0 is a feature vector extracted from a region proposal, which is the entire image in our case, and a history vector h_0 which is initialized to zero. In each of the following time steps, the agent will take the action with the highest expected future return evaluated by the Deep Q Network to transform the image to a new one. The new image should only contain a sub-region of the former one. Finally, the agent will receive a reward, then get to the next state by adding the taken action into the history vector and updating the feature vector to the one extracted from the new image.

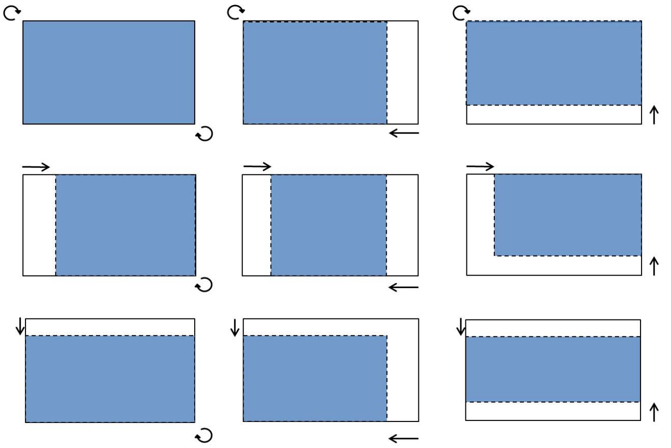

Fig. 2. The proposed action set with a flexible attention mechanism. The blue windows with dashed lines represent the next images after taking the corresponding actions. The arrows near the top left corners and the bottom right corners represent the moving directions of these two corners, and the circular arrows mean stopping.

An episode ends the target text region is found or the agent reaches the max step.

Our MDP framework is defined by the following components, namely, states, actions and reward scheme. Considering the characteristics of scene text, we formulate our MDP to become more suitable for our task, and achieve better results. The components of our MDP framework is detailed as follows.

A. state

The state is a vector containing two components, the feature vector of the current visible region and the history of all past actions. Since the agent deals with natural images in this task that are high-dimensional data, we use a pre-trained VGG16 [41] model to extract the information of the current region. The model is detailed in Section IV-B. Since there are many small text in natural images, as suggested in [40], the feature extractor takes as input each re-scaled visible region and the flattened output acts as the feature vector of the visible region in each time step.

Besides, we also concatenate the history of the past actions the agent has chosen, contributing to the current state representation. To be noted, the history vector records all the past actions in the current episode, which is different from that in [39] and [40]. This is done to help the agent make sense what have happened as well as summarize the number of steps in the current episode for the implementation of lazy punishment strategy. When the agent fails to locate targets in the predefined time steps, a punishment will be given by the reward scheme, which is detailed in III-C. The past actions in the history vector are represented in a one-hot manner. Assume that the given number of actions is n_a and the maximum steps in each search episode is h_{max}, the length of the history vector is $n_a * h_{max}$. We use $h_{max} = 25$ and $n_a = 9$ in the subsequent experimental setting.

B. action

Given the fact that the aspect ratio of scene text has a large variation due to the various numbers of the characters in a word, we design a new action set with a flexible attention mechanism integrated in to adapt to our targets. Considering that a horizontal rectangle is determined by its top left corner and bottom right corner, the action set is composed of combined actions of both corners. We only need to define the effective actions of these two corners, as is illustrated in Figure 2. The top left corner can give out three actions, namely, stopping, moving right and moving down; while the bottom right corner can choose to stop, move up or move left. With pair-wise combination of two actions emitted by the two corners, there are totally 9 independent actions to control the transformation of image. For example, when the top left corner moves down and the bottom right corner moves up, the visual region is flattened toward the center area; when the two corners choose to stop simultaneously, terminating action is triggered. In fact, all the deformable effects of 9 actions can be observed from the Figure 2, which provide comprehensive flexibility of simultaneous position and shape adjustment (e.g. aspect ratio).

With regard to the two works [39] and [40] on detecting general objects using deep reinforcement learning, they design actions to transform the visible regions in different ways. Caicedo et al. [39] proposed three kinds of actions including moving, changing the scale, and changing the aspect ratio, each of which is only involved in one of two elements (i.e., position and shape) at the same time. Moreover, search process guided by sequential actions that are decided by the agent is always carried out in the entire image. And thus the search space seems to be potentially enlarged. By comparison, we start from the whole image and gradually focus on the more accurate text area using the designed action set. After each action, the agent pays the attention only to the current sub-regions, limiting the search process conducted in the specific visible images. As a result, the searching space is reduced and its easier to converge. Miriam Bellver et al. [40] defined five sub-regions over each observed bounding box: four quarters distributed as over the box and a central overlapping region. Any sub-region being selected is equivalent to a movement action being executed. Here, each action implies the simultaneous change of position and shape (scale), however, the aspect ratio of the sub-regions in [40] is fixed, being restricted by the shape of the original images. Therefore, it is hard to adapt to the targets in different aspect ratios from the original images.

Further, we define the stride of each movement as below:

$$\beta_w = \beta * w \quad \beta_h = \beta * h \tag{1}$$

where $\beta \in \{0, 1\}$. w and h are the width and height of the current visible region respectively. In the follow-up experiments, we set $\beta = 1/6$.

The only action that doesn't transform the image is the trigger action . When the agent chooses to trigger, the current search episode is terminated and a new episode is started from the initial region proposal. Besides, the image at the end of each episode is covered to suppress the focused region and avoid the repetitive attractions towards the same targets.

The only action that doesn't transform the image is the trigger action . When the agent chooses to trigger, the current search episode is terminated and a new episode is started from the initial region proposal. Besides, the image at the end of each episode is covered to suppress the focused region and avoid the repetitive attractions towards the same targets.

C. dense reward scheme and lazy punishment

In [29], the agent receives a reward only when the game score is changed, and this leads to a sparse reward scheme in which most of the immediate reward signals are zero. In [39] and [40], given the positions of ground truth boxes, a denser reward scheme that considers the change of IoU after taking an movement action is defined. Compared to the sparse one in [29], the dense reward scheme provides more intensive information for the agent, and leads to a faster convergence. The reward functions for the movement actions r_m and the trigger action r_s are shown in equation (2) and (3) respectively, which are defined in [39] and [40].

$$r_m = sign(IoU(b', g) - IoU(b, g)) \in \{-1, 1\} \tag{2}$$

$$r_s = \begin{cases} +\lambda & IoU(i, g) \geq \delta \\ -\lambda & otherwise \end{cases} \tag{3}$$

where i and i' are the current and next visible image, respectively. g is the ground truth box for a scene text, and δ is set to 0.5, while λ is set to 3 in our setting.

Dense reward scheme is more suitable to detection task, however, introduces a new problem. When a dense reward scheme is adopted, the agent can receive a positive reward on every time step more easily, and therefore becomes lazy and fails to locate the targets before the specified steps are run out. In order to address the problem, we use equation (3) to measure r_s, which is the same as that in [39] and [40]. Additionally, we defines r_m in a different way as following:

$$r_m = \begin{cases} sign(IoU(i', g) - IoU(i, g)) & n < h_{max} \\ -\lambda & otherwise \end{cases} \tag{4}$$

Replacing equation (3) with (4) implies our lazy punishment strategy. Let's put it another way, when the number of steps the agent consumes is still within h_{max}, the reward depends on the change of IoU. Once the given steps are run out, the agent receive a negative reward regardless of the change of IoU.

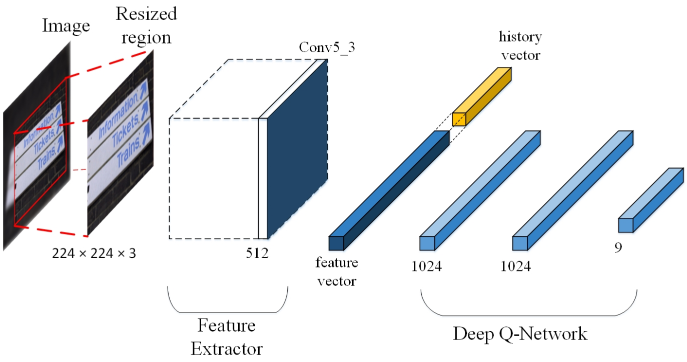

Fig. 3. The overall architecture of the proposed model. The model is composed of the feature extractor and the Deep Q-Network. In each time step, the feature vector of the visible region is extracted and then concatenated with the history vector to become the state of the visible region. The Deep Q-Network takes as input the state vector and predicts the expected future returns of the 9 actions.

IV. Training a agent with deep reinforcement learning

A. Q-learning

Following the MDP framework, The goal of the agent is to maximize the expected future return. We define the future return at time t as follow:

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} \tag{5}$$

where T is the time steps when the searching process is terminated within a episode. and γ is the discounted factor. In this task, we use Q-learning to train a agent with a goal of maximizing the expected future return, which is measured by the action-value function $Q(s, a)$ with the given state s and the chosen action a, and approximate the action-value function with a deep neural network, namely Deep Q-network. In each training iteration, the action-value function is updated using Bellman equation,

$$Q^*(s_t, a_t) = \mathbb{E}[r_t + \gamma \max_{a'} Q^*(s_{t+1}, a')] \tag{6}$$

where a' is the action with the highest Q value in state s_{t+1}.

B. network architecture

In our work, the agent has two components, the feature extractor and the deep Q-network, and the overall architecture is shown in Figure 3.

We refine the feature extractor to adapt to the scene text detection task. We train the feature extractor based on the pre-trained VGG16 [41] model so that it leads to the faster convergence. Since the features of VGG16 model pre-trained on ImageNet [42] are position-insensitive, we train it to become position-sensitive for this task with the method detailed in [43] and use the part from conv1_1 through conv5_3. Because we use the refined the feature extractor, there is no need to update the parameters of the feature extractor with much more training data during training the Deep Q-Network, leading to the faster optimization speed. In each time step, the visible region is resized to 224*224, and then is taken as the input of VGG-16, and we get a feature map from conv5_3, and the flattened feature map becomes the component of the state.

The Deep Q-network is a network including 2 fully-connected layers which are both followed by ReLU activation function and Dropout regularization. The Deep Q-network takes the state illustrated in Section 3.1 as input and outputs the expected future returns of all the actions defined in Section 3.2, and chooses the action with the highest expected future return.

C. training procedure

Since the deep reinforcement learning algorithm is based on trail and error, the agent learns the optimal policy by locating scene text on the training images for several epochs. During each training iteration, the agent chooses an action a_t with the current state s_t according to the ϵ-greedy scheme, receives the corresponding reward r_t given by the reward scheme described in Section III, and then observes a new visible region as well as obtain the new state s_{t+1}. Adopting a mechanism termed Experience Replay [29], we store the sample (s_t, a_t, r_t, s_{t+1}) generated at each time step in a replay memory D, and randomly select a batch of samples from it. Using the selected samples(s_j, a_j, r_j, s_{j+1}), we calculate the loss as follows:

$$L(\theta_j) = \mathbb{E}_{s_j,a_j \sim D}[(y_j - Q(s_j, a_j; \theta_j))^2] \tag{7}$$

where θ_j is the parameters of Q-network in the current iteration j, and y_j is the supervising signal:

$$y_j = \begin{cases} r_j & a_j \text{ is trigger} \\ r_j + \gamma \max_{a'} Q(s_{j+1}, a'; \theta_j) & \text{otherwise} \end{cases} \tag{8}$$

After calculating the loss, we update the parameters of Q-network using the following gradient:

$$\nabla_{\theta_j} L(\theta_j) = \mathbb{E}_{(s_j,a_j)\sim D}[(y_j - Q(s_j, a_j; \theta_j))\nabla_{\theta_j} Q(s_j, a_j; \theta_j)] \tag{9}$$

V. EXPERIMENTS

To evaluate our method, we conduct the quantitative experiments on the public benchmark, ICDAR 2013 [44]. We use the training set of ICDAR2013 to train the Deep Q-Network, and use the English2k sub-dataset of FORU dataset [14] together with the training set of ICDAR2013 to refine the feature extractor.

A. datasets

ICDAR 2013 consists of 229 training images and 233 testing images, and the text of each image has been marked at word-level for the text location task. The numbers of ground-truth boxes in training set and testing set are 849 and 1089, respectively.

FORU is collected from Flickr website. The dataset contains two parts that are Chinese2k and English2k sub-dataset. In the English2k sub-dataset, there are 1200 training images and 515 testing images at word-level.

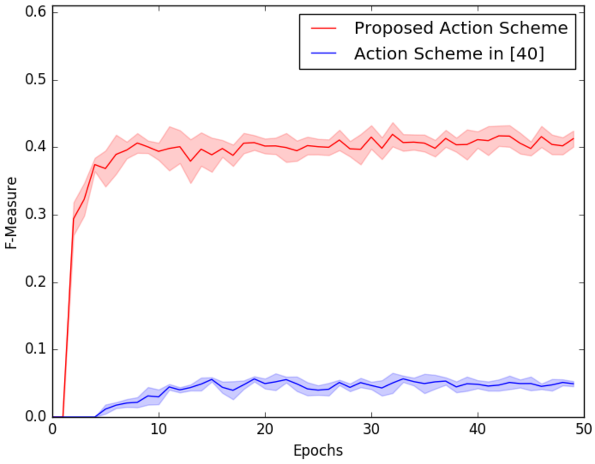

Fig. 4. The F-measure curves of the model using the proposed action scheme and the baseline model

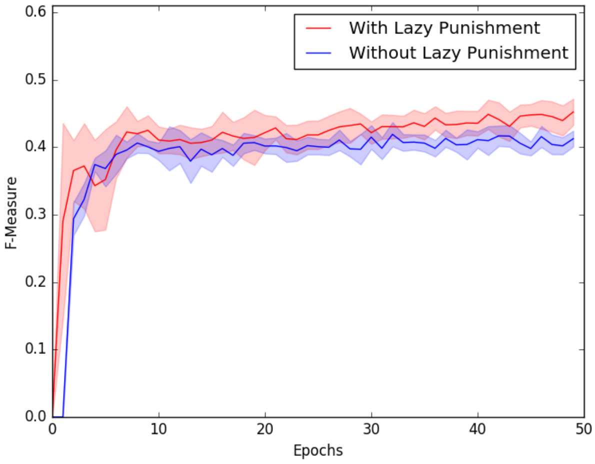

Fig. 5. The F-measure curves of the model using the dense reward scheme with and without the lazy punishment.

B. implementation details

The training of the Deep Q-network with the ϵ-greedy scheme lasts for 50 epochs, during which the ϵ falls smoothly from 1.0 to 0.1 in the first 10 epochs and becomes unchanged in the remaining epochs. In each iteration, the parameters are updated with Adam optimizer [45] with the learning rate of 10^{-7}, while the size of random batch from the replay memory that has a capacity of 1000, is set to 100. During testing, the maximum steps of each search episode is set to 25 so that it is possible for the agent to focus on anywhere in the images. Since there are multiple texts in each image, we set the maximum searching steps of each image to be a few times larger than 25, which is 200 in this task. The search episode is stopped when the agent chooses to trigger or 25 steps pass without triggering, and the final image of the episode is covered with the image mean of the dataset. When all the scene text regions have been searched and the prescribed steps of the whole image is not reached, we don't want to continue the searching process episode after episode. To this end, we stop the searching procedure of the image when the agent keeps searching in the same path. All experiments are replicated for 5 times and we report the average and standard deviation of the output. The predicted regions with IoU greater than 0.5 are regarded as the positive predictions. Since the precision and the recall are both considered in the scene text detection task, we use F-score as the measurement of the detection performance.

C. evaluation of the proposed action scheme

We first show how the detection performance is affected using the proposed action scheme. During the experiment, we implement the deep reinforcement learning method [40] to locate scene text using the original action set for training a baseline model. And we compare the detection performance after the proposed action scheme is integrated in the same framework with the benchmark, as illustrated in Figure 4. It can be observed that the F-score of the proposed method increased sharply after only 1 epoch of training is executed. In contrast, F-score of the method in [40] gently rise after 5 epochs implementation. From the figure, the model using the proposed flexible action scheme significantly outperform the baseline model. After both models reached the steady state, the proposed method achieved significantly higher F-score than the baseline. To summarize, our action scheme speeds up the convergence and improve the scene text detection performance.

D. evaluation of the dense reward scheme with lazy punishment

We then conduct an experiment to evaluate the effect of the model using lazy punishment. In this experiment, the same framework in [40] combining the proposed action scheme is used to train two agents. These two agents are guided by the dense reward scheme with and without the proposed lazy punishment, respectively, and the performances are shown in Figure 5. We can observe that the agent trained with the proposed lazy punishment begins to improve its performance earlier than the other one. Besides, when both of the agents become statble, the one with help of the lazy punishment have a higher F-score than the other one. In conclusion, the proposed lazy punishment helps the agent improve itself more quickly and leads to a better performance.

E. comparisons with other results

Table I presents the performances of [39], [40] and ours on ICDAR 2013. As Table I shows, our method achieves the highest precision P, recall R and F-score F among the three. Besides, it's also shown that the number of used region proposals R_w is less than 10 in Miriam Bellver's method [40] and our method on average. It is nearly as few as the number of the ground truth bounding box, which is counted to be 4.7 for the ICDAR 2013 test set. It may be caused by the commonalities that each action gradually reduces the search space and avoids full search in the entire image as in [39]. By comparison, R_w is 10.1 for the method in [39], as can be explained that search can be trapped in a loop within an episode and then the corresponding searching episode will not yield any text location result. To this end, larger number of searching episodes is required for the basically the same detection performance. Further, in contrast to most of region proposal based methods, all the three deep reinforcement learning based methods demonstrate obvious advantages in the number of region proposal. For example, [14], [46] generates 2000 and 1310 region proposals in each image on average, and [47] generates totally 215,325 region proposals in ICDAR2013, namely 940.3 region proposals per image. Finally, scene text detection performance in method in [40] is very poor, as demonstrated by the low precision and recall. Thus, the advantage of small number of region proposal doesn't make sense.

TABLE I
STATE-OF-THE-ART RESULTS ON THE ICDAR 2013.P, R, F AND RW ARE PRECISION, RECALL, F-MEASURE AND REGIONS PROPOSALS USED IN EACH IMAGE, RESPECTIVELY

Methods	P	R	F	R_w
Caicedo et al. [39]	0.65	0.30	0.41	10.1
Miriam Bellver et al. [40]	0.12	0.03	0.05	3.8
Our method	0.80	0.32	0.46	4.0

VI. Conclusion

We have presented a scene text detection method based on deep reinforcement learning. The method is fundamentally different from the mainstream methods in the field of scene text detection. During the detection process, the agent starts from the whole image and locates the text with sequential decisions. Considering the characteristics of scene text, we propose an action set with flexible attention mechanism and an improved reward scheme with a lazy punishment. Experiments show that the formulation following the MDP framework uses less region proposals than most region proposal based methods , and demonstrate that the proposed action set and reward scheme help the deep reinforcement learning algorithm to achieve better performance in terms of F-score.We have to admitted that there is still some distance left before the proposed deep reinforcement based method becomes the state-of-the-art. One of the possible improvements is to adopt some coarse region candidates as the starting points of the search episodes. It will be left for next step of our research work.

Acknowledgment

This work was supported by NSFC [No. 61673182, 61472144], the National Key Research Development Plan of China [No. 2016YFB1001405], GD-NSF (No. 2017A030312006), GDSTP (No. 2015B010131004, 2017A010101027), GZSTP(No. 201707010160, 201607010227)

References

[1] L.Ulanoff, "Hands on with google translate: A mix of awesome and ok," in Mashable, 2015.

[2] Looktel, http://www.looktel.com/.

[3] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," *arXiv preprint arXiv:1703.01425*, 2017.

[4] S. M. Hanif, L. Prevost, and P. Negri, "A cascade detector for text detection in natural scene images." in *ICPR*, 2008, pp. 1–4.

[5] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. Lim Tan, "Text flow: A unified text detection system in natural scene images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4651–4659.

[6] S. Tian, S. Lu, and C. Li, "Wetext: Scene text detection under weak supervision," *arXiv preprint arXiv:1710.04826*, 2017.

[7] S. M. Hanif and L. Prevost, "Text detection and localization in complex scene images using constrained adaboost algorithm," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE, 2009, pp. 1–5.

[8] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *European conference on computer vision*. Springer, 2014, pp. 512–528.

[9] K. I. Kim, K. Jung, and J. H. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1631–1639, 2003.

[10] W. Huang, Z. Lin, J. Yang, and J. Wang, "Text localization in natural images using stroke feature transform and text covariance descriptors," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1241–1248.

[11] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Asian Conference on Computer Vision*. Springer, 2010, pp. 770–783.

[12] K. Wang and S. Belongie, "Word spotting in the wild," in *European Conference on Computer Vision*. Springer, 2010, pp. 591–604.

[13] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang, "Scene text recognition using part-based tree-structured character detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2961–2968.

[14] Z. Zhong, L. Jin, S. Zhang, and Z. Feng, "Deeptext: A unified framework for text proposal generation and text detection in natural images," *arXiv preprint arXiv:1605.07314*, 2016.

[15] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," in *AAAI*, 2017.

[16] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[17] G. J. Zelinsky, "A theory of eye movements during target acquisition." *Psychological review*, vol. 115, no. 4, p. 787, 2008.

[18] C. Mancas-Thillou and B. Gosselin, "Spatial and color spaces combination for natural scene text extraction," in *Image Processing, 2006 IEEE International Conference on*. IEEE, 2006, pp. 985–988.

[19] ——, "Color text extraction with selective metric-based clustering," *Computer Vision and Image Understanding*, vol. 107, no. 1, pp. 97–107, 2007.

[20] C. Yi and Y. Tian, "Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4256–4268, 2012.

[21] V. Wu, R. Manmatha, and E. M. Riseman, "Textfinder: An automatic system to detect and recognize text in images," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 21, no. 11, pp. 1224–1229, 1999.

[22] M. Cai, J. Song, and M. R. Lyu, "A new approach for video text detection," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 1. IEEE, 2002, pp. I–I.

[23] P. Shivakumara, T. Q. Phan, and C. L. Tan, "New fourier-statistical features in rgb space for video text detection," *IEEE transactions on circuits and systems for video technology*, vol. 20, no. 11, pp. 1520–1532, 2010.

[24] Y. Zhong, H. Zhang, and A. K. Jain, "Automatic caption localization in compressed video," *IEEE transactions on pattern analysis and machine intelligence*, vol. 22, no. 4, pp. 385–392, 2000.

[25] H. Li, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital video," *IEEE transactions on image processing*, vol. 9, no. 1, pp. 147–156, 2000.

[26] P. Clark and M. Mirmehdi, "Recognising text in real scenes," *International Journal on Document Analysis and Recognition*, vol. 4, no. 4, pp. 243–257, 2002.

[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[29] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[30] J. Heinrich and D. Silver, "Smooth uct search in computer poker." in *IJCAI*, 2015, pp. 554–560.

[31] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu, "Reinforcement learning with unsupervised auxiliary tasks," *arXiv preprint arXiv:1611.05397*, 2016.

[32] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," in *Advances in Neural Information Processing Systems*, 2016, pp. 3675–3683.

[33] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[34] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu, "Feudal networks for hierarchical reinforcement learning," *arXiv preprint arXiv:1703.01161*, 2017.

[35] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017.

[36] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 1889–1897.

[37] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with large-scale data collection," in *International Symposium on Experimental Robotics*. Springer, 2016, pp. 173–184.

[38] Z. Yang, K. Merrick, H. Abbass, and L. Jin, "Multi-task deep reinforcement learning for continuous action control," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 3301–3307.

[39] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2488–2496.

[40] M. Bellver, X. Giró-i Nieto, F. Marqués, and J. Torres, "Hierarchical object detection with deep reinforcement learning," *arXiv preprint arXiv:1611.03718*, 2016.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[42] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[43] S. Zhang, Y. Liu, L. Jin, and C. Luo, "Feature enhancement network: A refined scene text detector," *arXiv preprint arXiv:1711.04249*, 2017.

[44] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras, "Icdar 2013 robust reading competition," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 1484–1493.

[45] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[46] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2558–2567.

[47] M. Busta, L. Neumann, and J. Matas, "Fastext: Efficient unconstrained scene text detector," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1206–1214.