Pre-print of article that will appear in 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019.

The published article is available on https://ieeexplore.ieee.org/document/8999065

Pelee-Text: A Tiny Convolutional Neural Network for Multi-Oriented Scene Text Detection

Manuel A. Córdova, Luis G. L. Decker, Jose L. Flores-Campana, Andreza A. dos Santos, Jhonatas S. Conceição
Allan Pinto, Helio Pedrini and Ricardo da S. Torres

Institute of Computing, University of Campinas, Av. Albert Einstein, 1251, Campinas, SP, Brazil

***Abstract*—Nowadays, scene text detection has received a lot of attention due to its complexity given variations in terms of orientations, font size, aspect ratio, and natural backgrounds. In this vein, several deep neural networks have been proposed to deal with this challenging problem. However, such networks produce "heavy" models, hampering their use in applications running in devices with computational constraints. Additionally, few works are focused on the detection of multi-oriented and/or multi-lingual text. Herein, we propose an end-to-end tiny convolutional neural network for multi-oriented multi-lingual scene text called Pelee-Text. Experimental results show that Pelee-Text is at least 3 times smaller than its counterparts with a speed of 2.93 and 18.64 frames per second for its multi-scale and 768-scale versions, respectively. Moreover, in terms of F-measure, our method achieved competitive results on four well-known datasets, i.e., ICDAR'2011 (90.96%), ICDAR'2013 (85.24%), ICDAR'2015 (80.08%), and MSRA-TD500 (80.90%).**

***Keywords*-Text detection, multi-oriented text, multi-lingual, end-to-end, mobile-network, convolutional neural network.**

I. Introduction

Scene text detection and recognition play an important role in a wide range of applications in real-world scenarios, such as traffic sign detection [1], image retrieval [2], and assistive applications [3]. Both detection and recognition tasks are challenging problems, which have been attracting the attention of machine learning and computer vision communities. Different from object detection and document analysis problems, the difficulty in detecting and recognizing texts in a scene relies on variations in textual elements related to font styles and sizes, blurring, orientations, projections, and complex/natural backgrounds.

Recently, methods based on Convolutional Neural Network (CNN) emerged as a promising technique to deal with several hard problems in scene text localization and recognition such as multi-scale and oriented text detections [4]. However, those current solutions address these issues by using deep architectures, such as VGG and ResNet [5, 6], which are computationally expensive in terms of memory and storage footprints. That makes their use unfeasible, in practice, in applications with computational constraints, such as mobile devices.

Aligned with this trend, He et al. [7] deal with text detection using direct regression and their proposal was evaluated over three deep network architectures (VGG-16 [5], ResNet-50 [6], and S-VGG [8]), originally designed for object detection. Moreover, we noticed that several approaches in the literature go deeper without concerns with efficiency aspects by fusing two or more deep architectures. In this vein, IncepText [9] combines two ResNet-101, two ResNet-50, and one VGG network. Fast Oriented Text Spotting method (FOTS) [10] merge tasks for text detection and recognition in the training stage. This method uses a ResNet-50 for text detection and a Recurrent Neural Network encoder composed of a VGG-16 and one bi-directional Long Short-Term Memory [11], along with a Connectionist Temporal Classification decoder [12] for text recognition. In total, FOTS ranges from 34 to 63 million of parameters.

Similarly, TextBoxes++ [4] uses VGG-16 aiming to overcome the limitations regarding arbitrary-oriented text detection by adapting the Single Shot MultiBox Detector network (SSD) [13]. Additionally, the authors used a Convolutional Recurrent Neural Network (CRNN) [14] for text recognition. Also inspired by VGG-16 architecture, Lyu et al. [15] presented an approach that takes advantage of Feature Pyramid Networks (FPN) [16] and Deconvolutional Single Shot Detector [17] for localizing corner points and segmenting text regions through position-sensitive maps prediction.

Currently, most of the methods focus on text segmentation as starting point instead of bounding boxes regression, which is especially helpful for deal with arbitrary shaped text. In this context, based on geometric attributes, Long et al. [18] proposed TextSnake that uses a Fully Convolutional Network (FCN) based on a U-Net approach with a VGG-16 network as backbone. Based on the same feature extractor, PixelLink [19] and TextField [20] define text instances taking into account pixel-wise and binary masks, respectively. Mask TextSpotter [21], in turn, defines a specific branch for handling text instances segmentation using a FPN with ResNet, along with a Fast R-CNN [22] for bounding boxes regression. Using a ResNet, PSENet [23] is a recent work based on segmentation for arbitrarily-oriented text detection that effectively separates closed text instances.

As we can observe, state-of-the-art methods are based in CNN-based solutions; nevertheless, they produce "heavy" models (all previously mentioned models are about 138MB and 350MB of model size), which may difficult their use in devices with computational restrictions. With this regards, this work introduces the Pelee-Text network, an end-to-end solution for text localization handling multi-oriented and multi-lingual texts. Our solution takes advantage of best features of TextBoxes++ [4] and PeleeNet [24], which are architectures specifically designed for image classification in restrictive computing scenarios. We evaluate our proposal over four well-known datasets: ICDAR'2011 (born-digital images), ICDAR'2013 (near horizontal text), ICDAR'2015 (arbitrary-oriented text), and MSRA-TD500 (multi-oriented and multi-lingual).

Our experiments demonstrated the ability of Pelee-Text by achieving competitive results. Experimentally, we show that Pelee-Text is at least 3 times smaller than its counterparts with a speed of 2.93 and 18.64 frames per second for its multi-scale and 768-scale versions, respectively. To the best of our knowledge, this is the first approach to scene text detection using mobile-oriented CNN architectures.

This paper is organized as follows. Section II presents our method. Section III details the adopted experimental protocol. Next, in Section IV, we present and discuss achieved results. Finally, Section V presents our conclusions and possible future research venues.

We thank Samsung R&D Institute Brazil and CNPq for the financial support. Authors are also grateful to São Paulo Research Foundation – FAPESP (grants #2014/12236-1, #2015/24494-8, #2016/50250-1, #2017/12646-3, and #2017/20945-0). This study was partially financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and also by Samsung Eletrônica da Amazônia Ltda., through the project "Algoritmos para Detecção e Reconhecimento de Texto Multilíngue (MLTSR)", within the scope of the Informatics Law No. 8248/91.

II. Proposed Method: Pelee-Text

This section presents the Pelee-Text network, our proposal of a fast and lightweight architecture designed for detecting and recognizing multi-oriented and multi-lingual text. Our goal is to introduce an efficient and effective architecture for scene text detection, which is expected to be more appropriate for constrained processing scenarios.

A. Overview

The proposed solution adopts Pelee [24], an efficient network recently proposed for object detection. In terms of efficiency, Pelee network was demonstrated to present a lower memory footprint and higher rates of FPS in comparison to MobileNetV2 network [25].

Pelee was adapted towards localizing multi-oriented textual elements in natural scenes, inspired by recent research results of scene text detection communities. As a result, we came up with a competitive "mobile" CNN, named Pelee-Text, which achieves competitive results in both tasks, text detection and recognition over four popular public datasets.

In natural scenes, oriented text and distortions associated with image projections are the main concerns for localizing textual elements. In this scenario, the typical solution based on determining rectangular bounding boxes are not convenient. To overcome this limitation, Pelee-Text uses text-boxes layers [4], which represent regions of interest as quadrilaterals defined by four vertices $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$, in clockwise order, where (x_1, y_1) refers to the first point located on the top left. Similar to other networks for object detection, the Pelee adopted the use of a Single Short MultiBox Detector (SSD) [13] for detecting objects, which performs a regression during the training stage for estimating the center (c_x, c_y) of the default bounding boxes (d) and their respective width (w) and height (h). In turn, our method performs a regression in the training stage towards estimating offsets for each vertex of the quadrilateral. It also assigns a confidence score c for each bounding box, considering a two-class classification problem defined in terms of text and non-text classes.

Additionally, to predict bounding boxes more efficiently, we use a simplified SSD version that uses two 19×19 feature maps with different prior-boxes scales in replacement to the 38×38 feature map of original SSD. Next, the last six layers were built considering kernels with rectangular receptive fields, by adopting kernels size 3×5, to deal with long continuous text regions and arbitrary orientations.

During the test stage, we use a complementary strategy for defining final bounding boxes applying a Non-Maximum Suppression (NMS) procedure over results from four scales (384×384, 768×768, 1024×1024, and 1536×1536), where each scale covers some special text region that could be missed for the rest of scales.

Finally, for evaluating our method as an end-to-end approach (text localization and recognition), we use a CRNN [14] for text recognition. For the end-to-end task, we expand the predicted bounding boxes before sending them to the CRNN recognizer, because predicted bounding boxes as well as the ground truth are very adjusted to the word and this hampers the recognition task.

B. Architecture

In this work, we adopted the Pelee network as a feature extractor, which comprises 5 stages. The first stage is composed of a stem block designed to empower the network for a better image characterization increasing the amount of channels with a minimum computational costs. In order to find useful features from large objects, Two-Way Dense Layers with stacked 3×3 convolutions are used. In opposite to DenseNet, whose one-way dense blocks work with number of channels $4\times$ larger than input channels, Pelee-Text inherits from Pelee an efficient scheme to control the channel expansion on each dense block using a two-way approach where each way works with half of channels. Moreover, a transitional layer is used at the end of each stage that does not compress the feature space, i.e., the same number of input channels are used as output, which is important to keep the feature discriminability [24]. An overview of Pelee-Text architecture is presented in Figure 1.

Six convolutional layers were specifically designed to detect textual elements at different scales using different feature maps. These layers are: the final layer of third stage (19×19) – used twice considering different scales for prior-boxes; the last layer of the fourth stage (10×10); and SSD extra-layers, which is referred to as $conv_2$ (5×5), $conv4$ (3×3), and $conv_6$ (1×1) layers, respectively.

More specifically, these layers use 3×5 filters in order to have receptive fields more appropriate to detect oriented textual elements. Furthermore, Pelee-Text uses different aspect ratios (2, 3, 5, $1/2$, $1/3$ and $1/5$), considering that texts are usually longer. Additionally, given that some areas are not fully covered for prior-boxes and, therefore, some text regions could be missed, we dense prior-boxes for covering those isolated regions based on vertical offsets. For computing the loss during training, we use the same function from [13], which takes into account the losses of localization (L_{loc}) and confidence (L_{conf}):

$$L(p, c, l, g) = \frac{1}{N}(L_{conf}(p, c) + \alpha L_{loc}(p, l, g)), \quad (1)$$

where p are the predicted bounding boxes, c is the confidence of being from the predicted class, l is the predicted location, g refers to the ground truth, N is the number of matched default boxes, and α is the weight for the L_{loc}. Moreover, L_{loc} is computed with the smooth L1 loss and L_{conf} with a 2-class soft-max loss (text or background).

Before prediction, each feature map from the six source layers passes through a residual block, and at the end, overlapped bounding boxes are discarded using two levels of NMS. First, an NMS is applied over bounding boxes with an overlap greater or equal than 0.5 taking into account the results from each of the four scales (384, 768, 1024, and 1536). Then, for text localization, a final NMS is performed over the final results merged from the four scales, the values for this second NMS stage are provided in Section III-B. On the other hand, for filtering the final results in the end-to-end task, we used Equation 2 proposed in [4], which takes advantage of the detection (d_s) and recognition (r_s) scores:

$$S = \frac{2 \times e^{(d_s + r_s)}}{e^{d_s} + e^{r_s}}. \quad (2)$$

III. Experimental Setup

In this section, we present the metrics adopted for measuring the effectiveness of the proposed method and datasets, along with their respective protocols.

A. Datasets

We evaluate our proposed method upon four benchmark datasets widely used in the literature with different particularities, as described in the following sections. Those datasets are detailed in Table I. The SynthText dataset is used during our first training stage and then, we executed a fine-tuning over each dataset.

B. Protocol

We train our network considering multiple stages towards aiding a multi-scale detection. Furthermore, we split the batches between two GPUs and the batch sizes differ between stages because of the image size (see Table II). In the first stage, we used SynthText [26] dataset

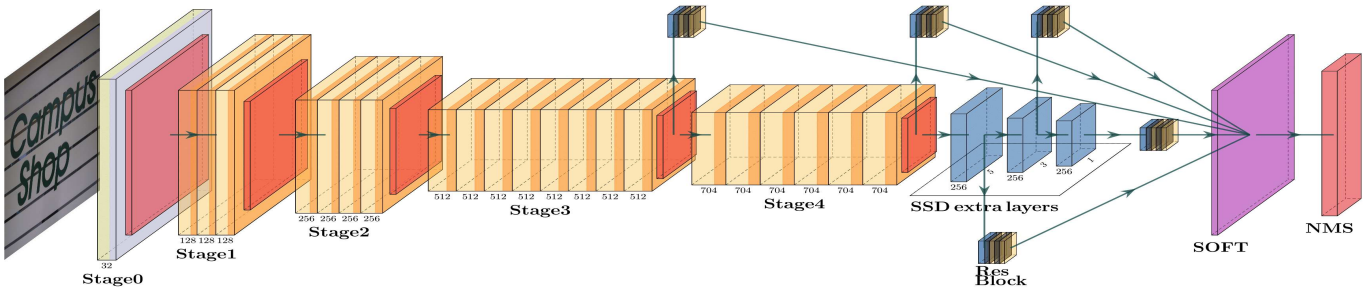

Fig. 1: Overview of the Pelee-Text architecture. Stage0 is composed of a stem block, while the next stages are composed of two-way dense blocks. At the end of each stage, a transition layer is added. On stages 0, 1, 2, 3 and 4 are used 3, 4, 8 and 6 two-way dense blocks, respectively. At the end, a simplified version of the SSD is used. Additionally, all layers used for prediction pass through residual blocks.

TABLE I: Datasets.

Dataset	#Train Images	#Test Images	Text Orientation	Languages
SynthText [26]	858,750	–	Horizontal	English
ICDAR'2011 [27]	410	141	Horizontal	English
ICDAR'2013 [28]	229	233	Horizontal	English
ICDAR'2015 [29]	1000	500	Arbitrary-oriented	English
MSRA-TD500 [30]	300	200	Multi-oriented	English-Chinese

with a batch size of 48, then two final stages for fine-tuning on each dataset with batch sizes of 48 and 10, respectively. Moreover, some training parameters are adjusted through different stages, i.e., learning rate, steps for learning rate decay, and L_{loc} weight. Additionally, given that some textures are very similar to text in natural scenes, we adjusted the ratio between the negatives and positives. Also, we used the Stochastic Gradient Descent (SGD) to optimize our network and the "Xavier" technique for initializing the weights. The weight decay and momentum were fixed in 5×10^{-4} and 0.9, respectively. It is worth mentioning that we used the same detection score (0.6) and overlap threshold (0.2) from [4] for ICDAR'2013 and ICDAR'2015 datasets. On the other hand, for ICDAR'2011 and MSRA datasets, those parameters were defined through a grid search procedure over training datasets. The overlap threshold for the two datasets was 0.1, and the detection score was 0.7 for ICDAR'2011 and 0.9 for MSRA.

We measure the effectiveness of our method in terms of Recall (R), Precision (P), and F-measure (F1). In turn, the efficiency takes into account the FPS, and memory and storage footprints. We consider a predicted bounding box as a true positive if the Intersection over Union (IoU) is equal or greater than 50%. To evaluate and compare Pelee-Text against state-of-the-art methods, we use the evaluation tools and lexicons (generic, weakly and strong) freely available for each dataset in the ICDAR Competition official site.[1] For ICDAR'13 dataset, we used the ICDAR13 metric, whereas for the MSRA dataset, we converted its ground truth representation (x ,y, *width*, *height*, θ) to quadrilateral format (x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4).

All experiments were performed considering an Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz with 12 cores, 64GB of RAM, Ubuntu 64-bits OS and two GeForce GTX 1080ti.

IV. Results and Discussion

This section presents the experimental results obtained to validate our method. For comparison purposes, given that Textboxes++ is directly related to Pelee-Text, we considered Textboxes++_MS which achieved the best results in [4] ("_MS" refers to its multi-scale version).

[1]http://rrc.cvc.uab.es/ (As of September 2019.)

A. Detecting Text in Born-Digital Images

We used the ICDAR'2011 dataset to analyze the effectiveness of our method in detecting text in born-digital image (Table III). This dataset contains low-resolution images with several JPG artifacts since these images were collected from the Internet. We observed that Pelee-Text was able to detect textual elements with better results than several methods published in the literature. For instance, the Pelee-Text_MS reached a relative error reduction of 60.18% and 39.73% in comparison with Jaderberg's work [32] and TextBoxes [33], respectively. Additionally, our single scale version of 768, running at 18.64 FPS, outperforms all the methods and losses by only 0.2 percentage points against our multi-scale version.

B. Detecting Horizontal and Near-Horizontal Texts

Table IV shows several results for the ICDAR'2013 and ICDAR'2015, which contain horizontal and near-horizontal texts with challenging scenarios. Given that our main focus is on the final model size generated by each method, it is worth mentioning that, for comparison purposes, in Table IV we considered only those methods which have information about their model size or number of parameters. That information was taken from their papers or authors' official Github. Moreover, missing values in the table means that the authors did not present those results on their papers.

As we can observe, our proposed network achieved competitive results. For the ICDAR'2013, Pelee-Text presented a good balance between F-measure and model size. In terms of model size, only FCN [34] (57MB) is directly comparable to Pelee-Text (40MB); however, our multi-scale version outperforms FCN by 2.24 and 26.08 percentage points in terms of F-measure on ICDAR'2013 and ICDAR'2015 datasets, respectively. On the other hand, considering state-of-the-art-methods, Textboxes++_MS [4], PixelLink_MS [19], MaskTextSpotter [21] and FOTS_MS [10] outperform our method, but their models size are 3.33, 6.15, 8.7, 3.38 times larger than Pelee-Text, respectively, which make difficult their use in devices with hardware constraints. Additionally, our single scale versions of 768 and 1024 obtained good results on the two datasets and they run at 18.64 and 11.67 FPS, respectively.

C. Detecting Multi-Oriented Multi-Lingual Text

Table V shows a comparison of effectiveness considering the MSRA-TD500 dataset. We can notice that our proposed method was able to detect both English and Chinese texts, as illustrated in Figure 2. This dataset contains high-resolution images and it is composed of English and Chinese texts, which makes this dataset more challenging than ICDAR'2013 and ICDAR'2015 datasets. The experimental results showed that Pelee-Text presented better performance results in terms of F-measure in comparison with methods

TABLE II: Parameter values and number of iterations used in the training protocol.

Learning Stage	Learning Rate	Image Size	Negative Ratio	L_{loc} Weight	Number of Iterations				Steps for Learning Rate Decay			
					ICDAR'2011	ICDAR'2013	ICDAR'2015	MSRA	ICDAR'2011	ICDAR'2013	ICDAR'2015	MSRA
Stage 1	5×10^{-3}	384	3	0.8	80K	80K	80K	80K	60K	60K	60K	60K
Stage 2	5×10^{-3}	384	3	0.2	4K	6k	12K	8k	2k, 3k	2k, 4k	4k, 8k	2k, 4k, 5k
Stage 3	5×10^{-4}	768	6	1.0	18k	10k	18k	16k	8k, 12k, 16k	4k, 8K	10k	6k, 12k

(a) Successful cases.

(b) Failure cases.

Fig. 2: Detection results of our proposed method: (a) correct detections; (b) some failure cases.

TABLE III: Text detection results on ICDAR'11 dataset.

Methods	P (%)	R (%)	F1 (%)
Text-CNN [31]	91.00	74.00	82.00
SynthText [26]	94.30	76.90	84.70
Jaderberg [32]	89.20	68.40	77.30
TextBoxes [33]	88.00	82.00	85.00
Pelee-Text_768	89.88	91.65	90.76
Pelee-Text_MS	94.99	87.26	90.96

such as PixelLink [19] and TextSnake [18], which have heavier models than ours, reaching a relative error reduction of 13.96% and 11.98%, respectively. Additionally, only TextField and the method proposed by Lyu et al. [15] outperformed our method by 0.4 and 0.6 percentage points on F-measure, respectively; nevertheless, their models in terms of Megabytes are at least 3× larger than Pelee-Text.

D. How Efficient is the Pelee-Text?

The comparison of efficiency in terms of both speed and memory footprints between Pelee-Text and our directly related work (TextBoxes++_MS) revealed that our method achieved a better balance between efficiency and efficacy. Regarding the model size, in terms of number of parameters, the Pelee-Text proved to be a lighter model, being 3.33× smaller than TextBoxes++. Moreover, concerning disk consumption, our proposed method also presented a thinner model than TextBoxes++_MS, whose disk consumption reached 133MB. In turn, Pelee-Text network requires only 40MB to storage its model (Table IV). Additionally, in terms of frames per second, Pelee-text is 1.29 times faster than its counterpart. In summary, the Pelee-Text network was superior in terms of frames per second and model size reduction, which reflects a lower disk and RAM memory consumption. Considering both efficiency and effectiveness results presented in this work, we believe our proposed network matches with our aims of achieving a CNN-based method more appropriate for a restrictive computing scenario.

E. Text Recognition Task

Similar to TextBoxes++, we also take advantage of Convolutional Recurrent Neural Network (CRNN) [14] to recognize text in candidate regions provided by Pelee-Text network toward building an end-to-end solution. Table VI presents a comparison of performance results for Pelee-Text and end-to-end recognition approaches available in the literature on the ICDAR'2013 and ICDAR'2015 datasets. Compared to state-of-the-art methods in terms of F-measure, our

TABLE IV: Text detection results on ICDAR'2013 and ICDAR'2015 datasets.

Methods	ICDAR'2013 P(%)	R(%)	F1(%)	FPS	ICDAR'2015 P(%)	R(%)	F1(%)	FPS	Model MB	#Parameters
FCN [34]	88.00	78.00	83.00	< 1.00	71.00	43.00	54.00	< 1.00	57	–
LC [15]	93.30	79.40	85.80	10.40	94.10	70.70	80.70	3.60	162	–
PixelLink_2s [19]	86.40	83.60	84.50	–	85.50	82.00	83.70	3.00	246	–
TextField [20]	−−	−−	−−	–	84.30	80.50	82.40	6.00	138	–
MaskTextSpotter [21]	95.00	88.60	91.70	4.60	91.60	81.00	86.00	4.80	348	–
PSENet [23]	−−	−−	−−	–	86.92	84.50	85.69	1.60	219	–
FOTS [10]	−−	−−	88.23	23.90	91.00	85.17	87.99	7.80	–	34.98M
Textboxes++ [4]	74.00	86.00	80.00	11.60	76.70	87.20	81.70	11.60	133	–
Pelee-Text_768	80.13	79.87	80.00	18.64	−−	−−	−−	−−	40	10.35M
Pelee-Text_1024	−−	−−	−−	−−	85.19	72.27	78.20	11.67	40	10.35M
LC_MS [15]	92.00	84.40	88.00	1.00	89.50	79.70	84.30	1.00	162	–
PixelLink_2s_MS [19]	88.60	87.50	88.10	–	−−	−−	−−	−−	246	–
TextField_MS [20]	−−	−−	−−	–	83.90	84.30	84.10	1.80	138	–
FOTS_MS [10]	−−	−−	92.50	–	91.85	87.92	89.84	−−	–	34.98M
Textboxes++_MS [4]	91.00	84.00	88.00	2.30	87.80	78.50	82.90	2.30	133	–
Pelee-Text_MS	88.41	82.28	85.24	2.93	87.73	73.66	80.08	2.93	40	10.35M

TABLE V: Text detection results on MSRA-TD500 dataset.

Methods	P (%)	R (%)	F1 (%)
EAST [35]	67.43	87.28	76.08
FCN [34]	83.00	67.00	74.00
LC [15]	87.60	76.20	81.50
SegLink [36]	86.00	70.00	77.00
TextSnake [18]	83.20	73.90	78.30
PixelLink_2s [19]	83.00	73.20	77.80
TextField [20]	87.40	75.90	81.30
Pelee-Text_768	74.78	73.88	74.33
Pelee-Text_MS	89.40	73.88	80.90

TABLE VI: End-to-end evaluation.

Methods	ICDAR'2013 Strong	Weakly	Generic	ICDAR'2015 Strong	Weakly	Generic
Neumann [37]	77.00	63.10	54.20	35.00	19.90	15.60
MaskTextSpotter [21]	92.20	91.10	86.50	79.30	73.00	62.40
FOTS_MS [10]	91.99	90.11	84.77	83.55	79.11	65.33
Textboxes++_MS [4]	93.00	92.00	85.00	73.34	65.87	51.90
Pelee-Text_MS	87.63	87.21	82.11	73.81	71.14	58.66

approach presented competitive on the ICDAR'2013 dataset. On the other hand, for the ICDAR'2015 dataset, we reached superior results than our directly related work (Textboxes++), with an improvement of 0.47, 5.27 and 6.76 percentage points considering the generic, weakly and strong lexicons, respectively. Additionally, on ICDAR'2011, in terms of F-measure, Pelee-Text obtained 85.33, 84.63, and 81.95 considering the strong, weakly and generic lexicons, respectively.

F. Discussion

Experimental results provided an overview of the performance of our proposed method, in terms of its effectiveness and efficiency, considering recent works published in the literature. The experiments showed that Pelee-Text is a very efficient and yet effective method for detecting text in several scenarios including born-digital images, scene text, multi-oriented textual elements, and bilingual texts. Figure 2 illustrates examples of success and failure, from which we can observe that the Pelee-Text network was able to detect textual elements in different orientations even considering complex backgrounds, and detections considering English and Chinese languages. On the other hand, Pelee-Text had some difficulty to localize texts in defocused scenes, textured textual elements in complex background, and (near)-vertical texts. We believe that we could improve our results, by focusing on these difficulties through smart data augmentation strategies, adopting new protocols for training, and considering other datasets for pre-training as some state-of-the arts methods do. We leave this investigation for future work.

V. Conclusions

This paper introduced a novel method for localizing and recognizing text in digital and natural scenes. Different from other works, our proposal focuses on devising and development of a tiny convolutional neural network for dealing with text localization inspiring its uses on mobile-oriented applications. In this work, we adapted the Pelee network, which was recently proposed for object detection, for our target problem, keeping in mind some particularities of textual elements. Motivated by the TextBoxes++ network, we proposed a mobile-based CNN architecture considering design decisions that favour detection of long and oriented textual elements such as use of polygonal bounding boxes, convolutional layers with rectangular receptive field, and default bounding boxes with different aspect ratios. The experimental results showed the effectiveness of the proposed method and also the improvements, in terms of efficiency, brought in this research, in comparison to current state-of-the-art methods for localizing text in natural scenes. This work showed the feasible of using a tiny CNN architecture for designing efficient text localization methods, which goes in opposite to recent trends in the text localization community, which has been adopting the fusion of deep architectures for devising novel text detectors.

Future research efforts will focus on better-characterizing texts with arbitrary shapes and (near)-vertical textual elements, considering polygon-based bounding boxes represented with more than four vertices or even pixel-based regression in text detection based on networks designed for segmentation problems.

References

[1] J. Greenhalgh and M. Mirmehdi, "Recognizing text-based traffic signs," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1360–1369, 2015.

[2] H. Yang and C. Meinel, "Content based lecture video retrieval using speech and video text information," *IEEE Trans. Learn. Technol.*, vol. 7, no. 2, pp. 142–154, 2014.

[3] C. Yi, Y. Tian, and A. Arditi, "Portable camera-based assistive text and product label reading from hand-held objects for blind

persons," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 3, pp. 808–817, 2014.

[4] B. S. Minghui Liao and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, 2018.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Representations.*, 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Int. Conf. Comput. Vision and Pattern Recognition*, 2016, pp. 770–778.

[7] W. He, X. Zhang, F. Yin, and C. Liu, "Multi-oriented and multi-lingual scene text detection with direct regression," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5406–5419, 2018.

[8] W. He, X. Zhang, F. Yin, and C. Liu, "Deep direct regression for multi-oriented scene text detection," in *IEEE Int. Conf. Comput. Vision and Pattern Recognition*, 2017, pp. 745–753.

[9] Q. Yang, M. Cheng, W. Zhou, Y. Chen, M. Qiu, and W. Lin, "Inceptext: A new inception-text module with deformable PSROI pooling for multi-oriented scene text detection," in *Int. Joint Conf. Art. Intell.*, 2018, pp. 1071–1077.

[10] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast Oriented Text Spotting with a Unified Network," *IEEE Int. Conf. Comput. Vision and Pattern Recognition*, pp. 5676–5685, 2018.

[11] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, pp. 2673–2681, 1997.

[12] A. Graves, S. Fernndez, and F. Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Int. Conf. Mach. Learn.*, 2006, pp. 369–376.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015.

[14] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, 2017.

[15] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *IEEE Int. Conf. Comput. Vision and Pattern Recognition*, 2018, pp. 7553–7563.

[16] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *IEEE Int. Conf. Comput. Vision and Pattern Recognition*, 2017, pp. 936–944.

[17] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," *CoRR*, vol. abs/1701.06659, 2017.

[18] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "Textsnake: A flexible representation for detecting text of arbitrary shapes," in *15th European Conf. Comput. Vision*, 2018, pp. 19–35.

[19] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," in *AAAI Conf. Art. Intell.*, 2018, pp. 6773–6780.

[20] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "Textfield: Learning a deep direction field for irregular scene text detection," *IEEE Trans. Image Process.*, 2019.

[21] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in *15th European Conf. Comput. Vision*, 2018, pp. 71–88.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[23] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," in *IEEE Int. Conf. Comput. Vision and Pattern Recognition*, 2019.

[24] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," in *Advances in Neural Information Processing Systems*, 2018, pp. 1967–1976.

[25] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *IEEE Int. Conf. Comput. Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[26] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *IEEE Int. Conf. Comput. Vision and Pattern Recognition*, 2016, pp. 2315–2324.

[27] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 Robust Reading Competition Challenge 2: Reading Text in Scene Images," in *Int. Conf. Document Analysis and Recognit.*, 2011, pp. 1491–1496.

[28] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazàn, and L. P. de las Heras, "ICDAR 2013 Robust Reading Competition," in *Int. Conf. Document Analysis and Recognit.*, 2013, pp. 1484–1493.

[29] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, "ICDAR 2015 Competition on Robust Reading," in *Int. Conf. Document Analysis and Recognit.*, 2015, pp. 1156–1160.

[30] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *IEEE Int. Conf. Comput. Vision and Pattern Recognition*, 2012, pp. 1083–1090.

[31] T. He, W. Huang, Y. Qiao, and J. Yao, "Text-attentional convolutional neural network for scene text detection," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2529–2541, 2016.

[32] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *Int. Journal Comput. Vision*, vol. 116, no. 1, pp. 1–20, 2016.

[33] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," in *AAAI Conf. Art. Intell.*, 2017, pp. 4161–4167.

[34] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *IEEE Int. Conf. Comput. Vision and Pattern Recognition*, 2016, pp. 4159–4167.

[35] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: An efficient and accurate scene text detector," in *IEEE Int. Conf. Comput. Vision and Pattern Recognition*, 2017, pp. 2642–2651.

[36] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *IEEE Int. Conf. Comput. Vision and Pattern Recognition*, 2017, pp. 3482–3490.

[37] L. Neumann and J. Matas, "Real-time lexicon-free scene text localization and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1872–1885, 2016.