PP-OCR: A Practical Ultra Lightweight OCR System

Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu,
Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, Haoshuang Wang
Baidu Inc.
{duyuning, yangyehua}@baidu.com

Abstract

The Optical Character Recognition (OCR) systems have been widely used in various of application scenarios, such as office automation (OA) systems, factory automations, online educations, map productions etc. However, OCR is still a challenging task due to the various of text appearances and the demand of computational efficiency. In this paper, we propose a practical ultra lightweight OCR system, i.e., PP-OCR. The overall model size of the PP-OCR is only 3.5M for recognizing 6622 Chinese characters and 2.8M for recognizing 63 alphanumeric symbols, respectively. We introduce a bag of strategies to either enhance the model ability or reduce the model size. The corresponding ablation experiments with the real data are also provided. Meanwhile, several pre-trained models for the Chinese and English recognition are released, including a text detector (97K images are used), a direction classifier (600K images are used) as well as a text recognizer (17.9M images are used). Besides, the proposed PP-OCR are also verified in several other language recognition tasks, including French, Korean, Japanese and German. All of the above mentioned models are open-sourced and the codes are available in the GitHub repository, i.e., https://github.com/PaddlePaddle/PaddleOCR.

1 Introduction

OCR (Optical Character Recognition), a technology which targets at recognizing text in images automatically as shown in Figure 1, has a long research history and a wide range of application scenarios, such as document electronization, identity authentication, digital financial system, and vehicle license plate recognition. Moreover, in factory, products can be more conveniently managed by extracting the text information of products automatically. Students offline homework or test paper can be electronized with an OCR system to make the communication between teachers and students more efficient. OCR can also be used for labeling the point of interests (POI) of a street view image, benefiting the map production efficiency. Rich application scenarios endow OCR technology with great commercial value, meanwhile, a lot of challenges.

Various of Text Appearances Text in image can be generally divided into two categories: scene text and document text. Scene text refers to the text in natural scene as shown in Figure 3, which usually changes dramatically for the factors such as perspective, scaling, bending, clutter, fonts, multilingual, blur, illumination, etc. Document text, as shown in Figure 4, is more often encountered in practical application. Different problems caused by the high density and long text need to be solved. Otherwise, document image text recognition often comes with the need to structure the results, which introduced a new hard task.

Figure 1: Some image results of the proposed PP-OCR system.

Computational Efficiency In practical, the images that need to be processed are usually massive, which makes high computational efficiency an important criterion for designing an OCR system. CPU is preferred to be used than GPU

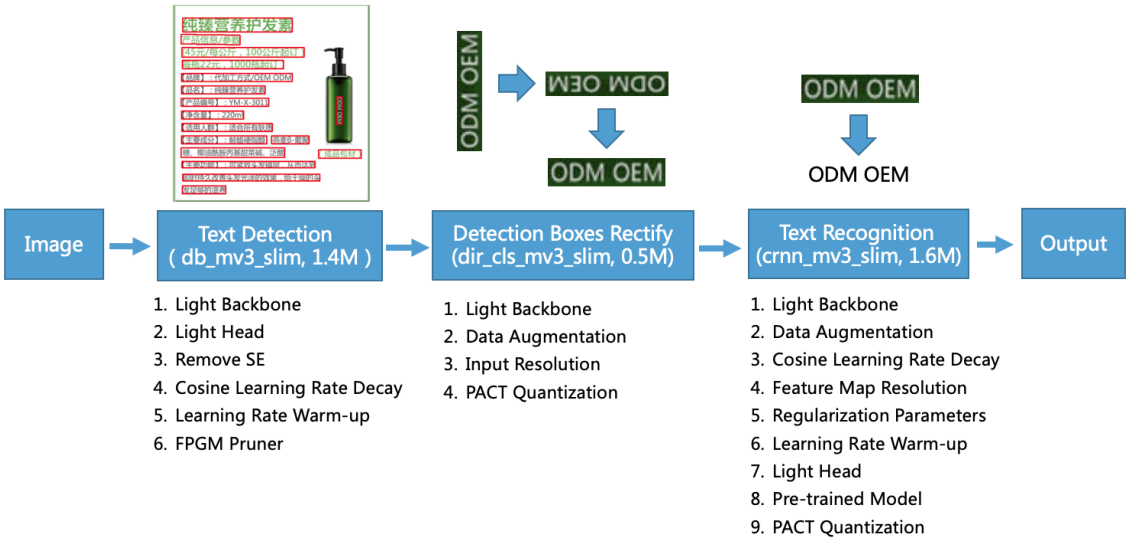


Figure 2: The framework of the proposed PP-OCR. The model size in the figure is about Chinese and English characters recognition. For alphanumeric symbols recognition, the model size of text recognition is from 1.6M to 0.9M. The rest of the models are the same size.

Figure 3: Some images contained scene text.

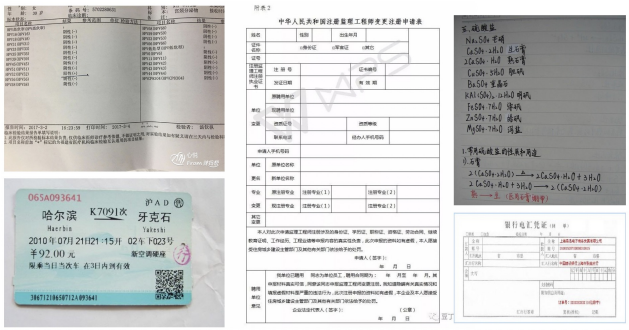


Figure 4: Some images contained document text.

considering the cost. In particular, the OCR system need to be run on embedded devices in many scenarios, such as cell phones, which makes it necessary to consider the model size. Trade off model size and performance is difficult but of great value. In this paper, we propose a practical ultra lightweight OCR system, named as PP-OCR, which consists of three parts, text detection, detected boxes rectification and text recognition as shown in Figure 2.

Text Detection The purpose of text detection is to locate the text area in the image. In PP-OCR, we use Differentiable Binarization (DB) (Liao et al. 2020) as text detector which is based on a simple segmentation network. The simple post-processing of DB makes it very efficient. In order to further improve its effectiveness and efficiency, the following six strategies are used: light backbone, light head, remove SE module, cosine learning rate decay, learning rate warm-up, and FPGM pruner. Finally, the model size of the text detector is reduced to 1.4M.

Detection Boxes Rectify Before recognizing the detected text, the text box needs to be transformed into a horizontal rectangle box for subsequent text recognition, which is easy to be achieved by geometric transformation as the detection frame is composed of four points. However, the rectified boxes may be reversed. Thus, a classifier is needed to determine the text direction. If a box is determined reversed, further flipping is required. Training a text direction classifier is a simple image classification task. We adopt the following four strategies to enhance the model ability and reduce the model size: light backbone, data augmentation, input resolution and PACT quantization. Finally, the model size of the text direction classifier is 500KB.

Text Recognition In PP-OCR, we use CRNN (Shi, Bai, and Yao 2016) as text recognizer, which is widely used and practical for text recognition. CRNN integrates feature extraction and sequence modeling. It adopts the Connectionist Temporal Classification(CTC) loss to avoid the inconsistency between prediction and label. To enhance the model ability and reduce the model size of a text recognizer, the following nine strategies are used: light backbone, data augmentation, cosine learning rate decay, feature map resolu-

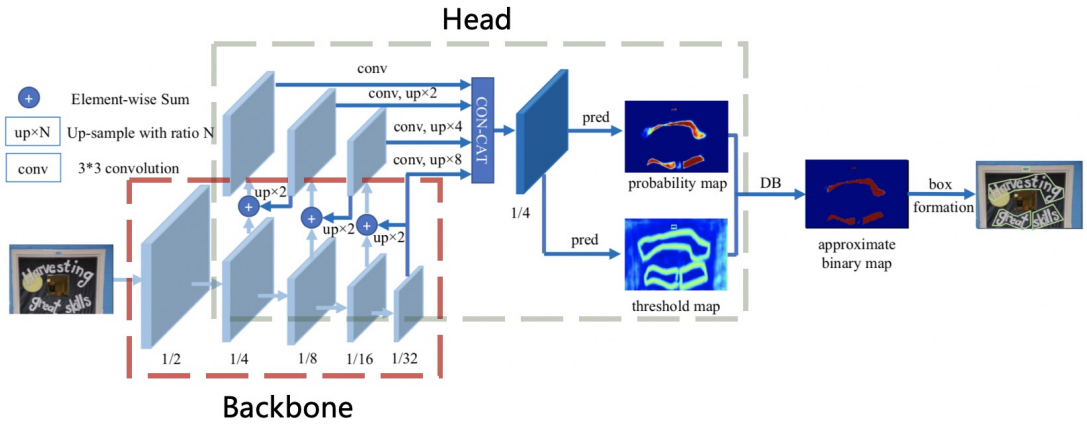


Figure 5: Architecture of the text detector DB. This figure comes from the paper of DB (Liao et al. 2020). The red and gray rectangles show the backbone and head of the text detector separately.

tion, regularization parameters, learning rate warm-up, light head, pre-trained model and PACT quantization. Finally, the model size of the text recognizer is only 1.6M for Chinese and English recognition and 900KB for alphanumeric symbols recognition.

In order to implement a practical OCR system, we construct a large-scale dataset for Chinese and English recognition as an example. Specifically, text detection dataset has 97K images. Direction classification dataset has 600k images. Text recognition dataset has 17.9M images. A small amount of the data are selected to conduct ablation experiments quickly and choose the appropriate strategies. We make a lot of ablation experiments to show the effects of different strategies in Figure 2. Besides, we also verify the proposed PP-OCR system for other languages recognition which including alphanumeric symbols, French, Korean, Japanese and German.

The rest of the paper is organized as follows. In section 2, we present the bag of model enhancement or slimming strategies. Experimental results are discussed in section 3 and conclusion is conducted in section 4.

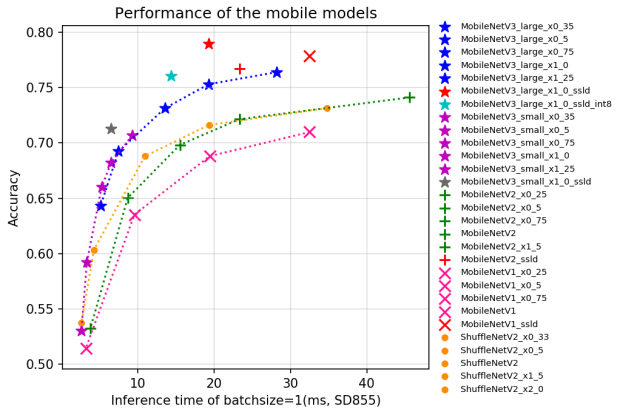


Figure 6: The performance of some light backbones on the ImageNet 1000 classification, including MobileNetV1, MobileNetV2, MobileNetV3 and ShuffleNetV2 series. The inference time is tested on Snapdragon 855 (SD855) with the batch size set as 1.

2 Enhancement or Slimming Strategies

2.1 Text Detection

In this section, the details of seven strategies for enhancing the model ability or reducing the model size of a text detector will be introduced. Figure 5 shows the architecture of the text detector DB.

Light Backbone The size of backbone has dominant effect on the model size of a text detector. Therefore, light backbones should be selected for building the ultra lightweight models. With the development of image classification, MobileNetV1, MobileNetV2, MobileNetV3 and ShuffleNetV2 series are often used as the light backbones. Each series has different scale. Thanks to the inference time on CPU and accuracy of more than 20 kinds of backbones are provided by PaddleClas[1], as shown in Figure 6, MobileNetV3 can achieve higher accuracy when the predict time are same. As for the choice of scale, we adopt MobileNetV3_large_x0.5 to balance accuracy and efficiency empirically. Incidentally, PaddleClas provides a total of up to 24 series of image classification network structures and training configurations, 122 models' pretrained weights and their evaluation metrics, such as ResNet, ResNet_vd, SEResNeXt, Res2Net, Res2Net_vd, DPN, DenseNet, EfficientNet, Xception, HRNet, etc.

Light Head The head of the text detector is similar as the FPN (Lin et al. 2017) architecture in object detection and fuse the feature maps of the different scales to improve the effect for the small text regions detection. For convenience of merging the different resolution feature maps, 1×1 convolution is often used to reduce the feature maps to the same number of channel (we use inner_channels for

[1]https://github.com/PaddlePaddle/PaddleClas/

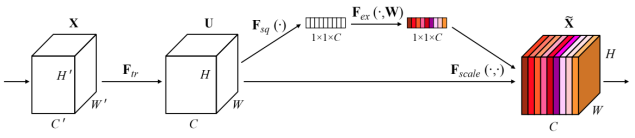


Figure 7: Architecture of the SE block. This figure comes from the paper (Hu, Shen, and Sun 2018).

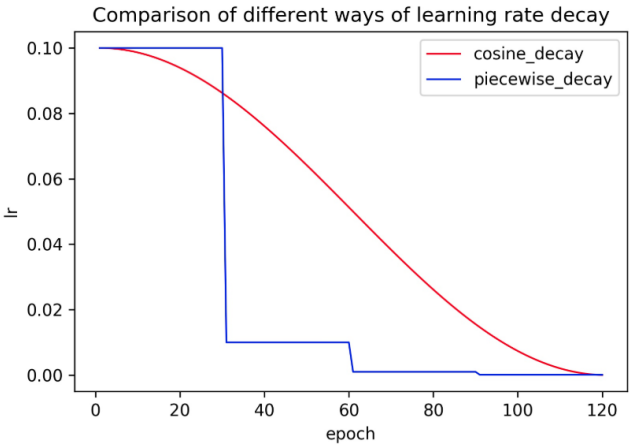


Figure 8: Comparison of different ways of learning rate decay.

short). The probability map and the threshold map are generated from the fused feature map with convolutions which are also associated with the above inner_channels. Thus inner_channels has a great influence on the model size. When inner_channels is reduced from 256 to 96, the model size is reduced from 7M to 4.1M, but the accuracy declines slightly.

Remove SE SE is the short for squeeze-and-excitation (Hu, Shen, and Sun 2018). As shown in Figure 7, SE blocks model inter-dependencies between channels explicitly and re-calibrate channel-wise feature responses adaptively. Because SE blocks can improve the accuracy of the vision tasks obviously, the search space of MobileNetV3 contains them and numerous of SE blocks are in MobileNetV3 architecture. However, when the input resolution is large, such as 640×640, it is hard to estimate the channel-wise feature responses with the SE block. The accuracy improvement is limited, but the time cost is very high. When the SE blocks are removed from the backbone, the model size is reduced from 4.1M to 2.5M, but the accuracy has no effect.

Cosine Learning Rate Decay The learning rate is the hyperparameter to control the learning speed. The lower the learning rate, the slower the change of the loss value. Though using a low learning rate can ensure that you will not miss any local minimum, but it also means that the convergence speed is slow. In the early stage of training, the weights are in random initialization state, so we can set a relatively large learning rate for faster convergence. In the late stage of training, the weights are close to the optimal values, so a relatively smaller learning rate should be used. Cosine learning rate decay has become the preferred learning rate reduction strategy for improving model accuracy. During the entire training process, cosine learning rate decay keeps a relatively large learning rate, so its convergence is slower, but the final convergence accuracy is better. Figure 8 compares the different ways of learning rate decay.

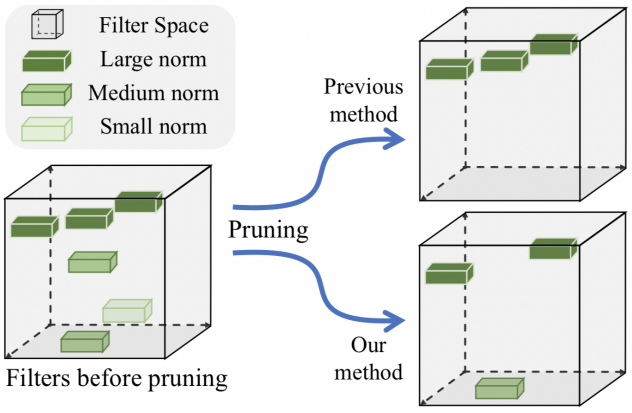


Figure 9: Illustration of FPGM Pruner. This figure comes from the paper (He et al. 2019b).

Learning Rate Warm-up The paper (He et al. 2019a) shows that using learning rate warm-up operation can help to improve the accuracy in the image classification. At the beginning of the training process, using a too large learning rate may result in numerical instability, a small learning rate is recommended to be used. When the training process is stable, the initial learning rate is to be used. For text detection, the experiments show that this strategy also is effective.

FPGM Pruner Pruning is another method to improve the inference efficiency of neural network model. In order to avoid the model performance degradation caused by the model pruning, we use FPGM (He et al. 2019b) to find the unimportant sub-network in original models. FPGM uses geometric median as the criterion and the each filter in a convolution layer is considered as a point in Euclidean space. Then calculate the geometric median of these points and remove the filters with the similar values, as shown in Figure 9. The compress ratio of each layer is also important for pruning a model. Pruning every layer uniformly usually leads to significant performance degradation. In PP-OCR, the pruning sensitivity of each layer is calculated according to the method in (Li et al. 2016) and then used to evaluate the redundancy of each layer.

2.2 Direction Classification

In this section, the details of four strategies for enhancing the model ability or reducing the model size of a direction classifier will be introduced.

Light Backbone We also adopt MobileNetV3 as the backbone of the direction classifier which is the same as the text detector. Because this task is relatively simple, we use MobileNetV3_small_x0.35 to balance accuracy and efficiency empirically. When using larger backbones, the accuracy doesn't improve more.

Data Augmentation This paper (Yu et al. 2020) shows

some image processing operations to train a text recognizer, such as rotation, perspective distortion, motion blur and Gaussian noise. Those processes are referred to as BDA (Base Data Augmentation) for short. They are randomly added to the training images. The experiment shows that BDA also is useful for the direction classifier training. Besides BDA, some new data augmentation operations are proposed recently for improving the effect of image classification, for example, AutoAugment (Cubuk et al. 2019), RandAugment (Cubuk et al. 2020), CutOut (DeVries and Taylor 2017), RandErasing (Zhong et al. 2020), HideAndSeek (Singh and Lee 2017), GridMask (Chen 2020), Mixup (Zhang et al. 2017) and Cutmix (Yun et al. 2019). But the experiments show that most of them don't work for the direction classifier training except for RandAugment and RandErasing. RandAugment works best. Eventually, we add BDA and RandAugment to the training images of the direction classification.

Input Resolution In general, when the input resolution of a normalized image is increased, accuracy will also be improved. Since the backbone of the direction classifier is very light, increasing the resolution properly will not lead to the computation time raise obviously. In the most of the previous text recognition methods, the height and width of a normalized image is set as 32 and 100, respectively. However, in PP-OCR, the height and width is set as 48 and 192, respectively, to improve the accuracy of the direction classifier.

PACT Quantization Quantization allows the neural network model to have lower latency, smaller volume and lower computational power consumption. At present, quantization is mainly divided into two categories: offline quantization and online quantization. Offline quantization refers to a fixed-point quantization method that uses methods such as KL divergence and moving average to determine quantization parameters and does not require retraining. Online quantization is to determine quantization parameters during the training process, which can provide less quantization loss than offline quantization mode.

PACT (PArameterized Clipping acTivation) (Choi et al. 2018) is a new online quantification method that removes some outliers from the activations in advance. After removing the outliers, the model can learn more appropriate quantitative scales. The formula for PACT to preprocess the activations is as follows:

$$y = PACT(x) = 0.5(|x| - |x-\alpha| + \alpha) = \begin{cases} 0 & x \in (-\infty, 0) \\ x & x \in [0, \alpha) \\ \alpha & x \in [\alpha, +\infty) \end{cases} \tag{1}$$

The preprocessing of the activation value of the ordinary PACT method is based on the ReLU function. All activation values greater than a certain threshold are truncated. However, the activation functions in MobileNetV3 are not only ReLU, but also hard swish. Using ordinary PACT quantization leads to a higher quantization loss. Therefore, we modify the formula of the activations preprocessing as follows to reduce the quantization loss.

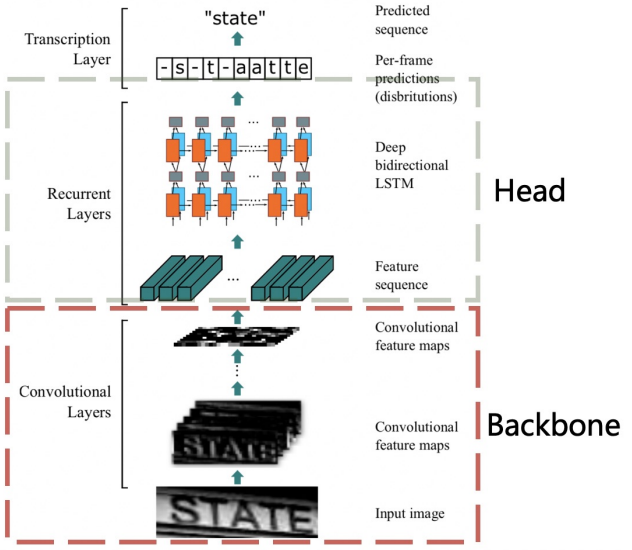


Figure 10: Architecture of the text recognizer CRNN. This figure comes from the paper (Shi, Bai, and Yao 2016). The red and gray rectangles show the backbone and head of the text recognizer separately.

$$y = PACT(x) = \begin{cases} -\alpha & x \in (-\infty, -\alpha) \\ x & x \in [-\alpha, \alpha) \\ \alpha & x \in [\alpha, +\infty) \end{cases} \tag{2}$$

We used the improved PACT quantification method to quantify the direction classifier model. In addition, L2 regularization with a coefficient of 0.001 is added to the PACT parameters to improve the model robustness.

The implementation of the above FPGM Pruner and PACT quantization is based on PaddleSlim[1]. PaddleSlim is a toolkit for model compression. It contains a collection of compression strategies, such as pruning, fixed point quantization, knowledge distillation, hyperparameter searching neural architecture search.

2.3 Text Recognition

In this section, the details of nine strategies for enhancing the model ability or reducing the model size of a text recognizer will be introduced. Figure 10 shows the architecture of the text recognizer CRNN.

Light Backbone We also adopt MobileNetV3 as the backbone of the text recognizer which is the same as the text detection. MobileNetV3_small_x0.5 is selected to balance accuracy and efficiency empirically. If you're not that sensitive to the model size, MobileNetV3_small_x1.0 is also a good choice. The model size is just increased by 2M, the accuracy is improved obviously.

Data Augmentation Besides BDA (Base Data Augmentation) which is often used in text recognition as mentioned earlier, TIA (Luo et al. 2020) also is an effective data augmentation method for text recognition. As shown in Figure

[1]https://github.com/PaddlePaddle/PaddleSlim/

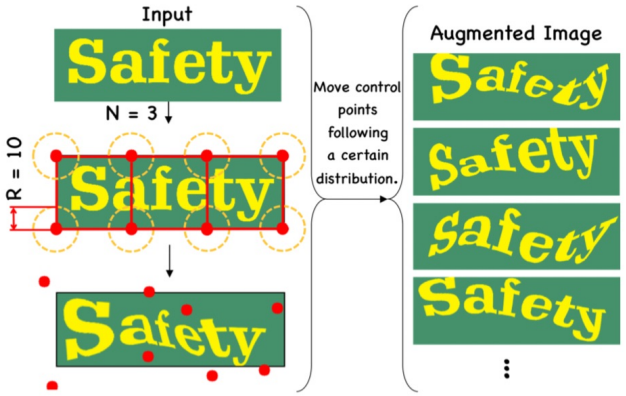
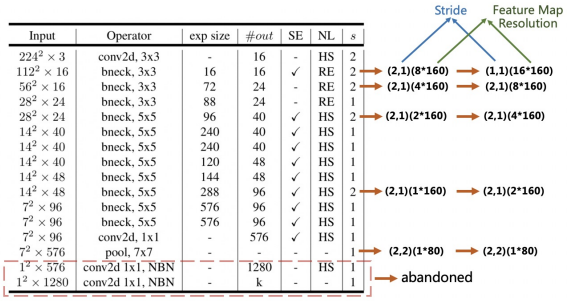


Figure 11: Illustration of data augmentation, TIA. This figure comes from the paper (Luo et al. 2020).



Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Figure 12: Illustration of the modify of the feature map resolution. The table comes from the paper (Howard et al. 2019)

11, at first, a set of fiducial points are initialized on the image. Then move the points randomly to generate a new image with the geometric transformation. In PP-OCR, we add BDA and TIA to the training images of the text recognition.

Cosine Learning Rate Decay As mentioned in text detection, cosine learning rate decay has become the preferred learning rate reduction method. The experiments show that cosine learning rate decay strategy is also effective to enhance the model ability for text recognition.

Feature Map Resolution In order to adapt to multilingual recognition, particularly in Chinese recognition, in PP-OCR the height and width of the CRNN input are set as 32 and 320. Then, the strides of the original MobileNetV3 is not appropriate for text recognition. As shown in Figure 12, for the sake of keeping more the horizontal information, we modify the stride of the down sampling feature map except the first one from (2,2) to (2,1). In order to keep more vertical information, we further modify the stride of the second down sampling feature map from (2,1) to (1,1). Thus, the stride of the second down sampling feature map s_2 affects the resolution of the whole feature map and the accuracy of the text recognizer dramaticly. In PP-OCR, s_2 is set as (1,1) to achieve the better performance empirically.

Regularization Parameters Overfitting is a common term in machine learning. A simple understanding is that the model performs well on the training data, but it performs poorly on the test data. To avoid overfitting, many regular ways have been proposed. Among them, weight_decay is one of the widely used ways to avoid overfitting. After the final loss function, L2 regularization (L2_decay) is added to the loss function. With the help of L2 regularization, the weight of the network tend to choose a smaller value, and finally the parameters in the entire network tends to 0, and the generalization performance of the model is improved accordingly. For text recognition, L2_decay has a great influence on the accuracy.

Learning Rate Warm-up Similar as the text detection, learning rate warm-up is also helping the text recognition. For text recognition, the experiments show that using this strategy is also effective.

Light Head A full connection layer is used to encode the sequence features to the predicted characters in the ordinary. The dimension of the sequence features have an impact on the model size of a text recognizer, especially for Chinese recognition whose characters are more than 6 thousands. Meanwhile, it is not that the higher of the dimension, the stronger of the ability of the sequence features representation. In PP-OCR, the dimension of the sequence features is set to 48 empirically.

Pre-trained Model If the training data is fewer, fine tune the existing networks, which are trained on a large data set such as ImageNet, to achieve fast convergence and better accuracy. The transfer learning in image classification and object detection show the above strategy is effective. In real scenes, the data used for text recognition is often limited. If the models are trained with tens of millions samples, even if they are synthesized ones, the accuracy can be significantly improved with the above models. We demonstrate the effectiveness of this strategy through experiments.

PACT Quantization We adopt the similar quantization scheme of the direction classification to reduce the model size of a text recognizer except for skipping the LSTM layers. Those layers will not be quantified at present since the complexity of LSTM quantization.

3 Experiments

3.1 Experimental Setup

DataSets As shown in Table 1, in order to implement a practical OCR system, we construct a large-scale dataset for Chinese and English recognition as an example.

For text detection, there are 97k training images and 500 validation images. Among the training images, 68K images are real scene images, which come from some public datasets and Baidu image search. The public datasets used include LSVT (Sun et al. 2019), RCTW-17 (Shi et al. 2017), MTWI 2018 (He and Yang 2018), CASIA-10K (He et al. 2018), SROIE (Huang et al. 2019), MLT 2019 (Nayef et al. 2019), BDI (Karatzas et al. 2011), MSRA-TD500 (Yao et al. 2012) and CCPD 2019 (Xu et al. 2018). Most the training images from Baidu image search are document text images. The remaining 29K synthetic images mainly focus on the scenarios for long text, multi direction text and table text. All the validation images come from the real scenes.

Task	Number of training data			Number of validation data
	Total	Real	Synthesis	Real
Text Detection	97K	68K	29K	500
Direction Classification	600K	100K	500K	310K
Text Recognition	17.9M	1.9M	16M	18.7K

Table 1: Statistics of dataset for Chinese and English Recognition.

Task	Character Number	Number of training data			Number of validation data		
		Total	Real	Synthesis	Total	Real	Synthesis
Chinese and English Recognition	6622	17.9M	1.9M	16M	18.7K	18.7K	0
Alphanumeric Symbols Recognition	63	15M	0	15M	12K	12K	0
French Recognition	118	1.08M	0	1.08M	80K	0	80K
Japanese Recognition	4399	0.99M	0	0.99M	80K	0	80K
Korean Recognition	3636	0.94M	0	0.94M	80K	0	80K
German Recognition	131	1.96M	0	1.96M	170K	0	170K

Table 2: Statistics of dataset for multilingual recognition.

For direction classification, there are 600k training images and 310K validation images. Among the training images, 100K images are real scene images, which come from the public datasets (LSVT, RCTW-17, MTWI 2018). They are horizontal text which rectify and crop the ground truth of the images. The remaining 500K synthetic images mainly focus on the reversed text. We use the vertical fonts to synthesize some text images and then rotate them horizontally. All the validation images come from the real scenes.

For text recognition, there are 17.9M training images and 18.7K validation images. Among the training images, 1.9M images are real scene images, which come from some public datasets and Baidu image search. The public datasets used include LSVT, RCTW-17, MTWI 2018 and CCPD 2019. The remaining 16M synthetic images mainly focus on the scenarios for different backgrounds, translation, rotation, perspective transformation, line disturb, noise, vertical text and so on. The corpus of synthetic images come from the real scene images. All the validation images also come from the real scenes.

In order to conduct ablation experiments quickly and choose the appropriate strategies, we select 4k images from the real scene training images for text detection, and 300k ones from the real scene training images for text recognition.

In addition, we collected 300 images for different real application scenarios to evaluate the overall OCR system, including contract samples, license plates, nameplates, train tickets, test sheets, forms, certificates, street view images, business cards, digital meter, etc. Figure 3 and Figure 4 show some images of the test set.

Furthermore, to verify the proposed PP-OCR for other languages, we also collect some corpus for alphanumeric symbols recognition, French recognition, Korean recognition, Japanese recognition and German recognition. Then synthesize the text line images for text recognition. Some images for alphanumeric symbols recognition come from the public datasets, ST (Gupta, Vedaldi, and Zisserman 2016) and SRN (Yu et al. 2020). Table 2 shows the statistics. Since MLT 2019 for text detection includes multilingual images, the text detector for Chinese and English recognition also can support multi language text detection. Due to the limited data, we haven't found the proper data to train the direction classifier for multilingual.

The data synthesis tool used in text detection and text recognition is modified from text_render (Sanster 2018).

Implementation Details We use Adam optimizer to train all the models and adopt cosine learning rate decay as the learning rate schedule. The initial learning rate, batch size and the number of epochs for different tasks can be found in Table 4. When we obtain the trained models, FPGM pruner and PACT quantization can be used to reduce the model size further with the above models as the pre-trained ones. The training processes of FPGM pruner and PACT quantization are similar as previous.

In the inference period, HMean is used to evaluate the performance of a text detector. Accuracy is used to evaluate the performance of a direction classifier or a text recognizer. F-score is used to evaluate the performance of an OCR system. In order to calculate F-score, a correct text recognition result should be the accurate location and the same text. GPU inference time is tested on a single T4 GPU. CPU inference time is tested on a Intel(R) Xeon(R) Gold 6148. We use the Snapdragon 855 (SD 855) to evaluate the inference time of the quantification models.

3.2 Text Detection

Table 5 compares the performance of the different backbones for text detection. HMean, the model size and the in-

inner_channel of the head	Remove SE	Cosine Learning Rate Decay	Learning Rate Warm-up	Precision	Recall	HMean	Model Size (M)	Inference Time (CPU, ms)
256				0.6821	0.5560	0.6127	7	406
96				0.6677	0.5524	0.6046	4.1	213
96	√			0.6952	0.5413	0.6087	2.6	173
96	√	√		0.7034	0.5404	0.6112	2.6	173
96	√	√	√	0.7349	0.5420	0.6239	2.6	173

Table 3: Ablation study of inner_channel of the head, SE, cosine learning rate decay, learning rate warm-up for text detection.

Task	Initial Learning Rate	Batch Size	Number of Epochs	
			Ablation Data	Total Data
Text Detection	0.001	16	400	60
Direction Classification	0.001	512	100	100
Text Recognition	0.001	1024	500	100

Table 4: Implementation details of the model training.

Backbone	HMean	Model Size (M)	Inference Time (CPU, ms)
MobileNetV3_large_x1	0.6463	16	447
MobileNetV3_large_x0.5	0.6127	7	406
MobileNetV3_large_x0.35	0.5935	5.4	367
MobileNetV3_small_x1	0.5919	7.5	380

Table 5: Compare the performance of the different backbones for text detection.

ference time of the different scales of MobileNetV3 change greatly. In PP-OCR, we choose MobileNetV3_large_x0.5 to balance accuracy and efficiency.

Tabel 3 shows the ablation study of inner_channel of the head, SE, cosine learning rate decay, learning rate warm-up for text detection. Firstly, by reducing the internal channels of the detector head from 256 to 96, the model size was reduced by 41%, and the inference time was accelerated by nearly 50% with HMean only dropped slightly. Therefore, reducing the inner channel is an effective way to lighten the detector. Then, when remove the SE block of the detector backbone, the model size is reduced 36.6% and the inference time has accelerated 18.8% further. Meanwhile, HMean will not be affected. Therefore, for text detection, the accuracy improvement of SE blocks is limited, but the time cost is very high. Finally, using both cosine learning rate decay instead of the fix learning rate and learning rate warm-up, HMean will be improved obviously. At the same time, the model size and the inference time will not be affected. Cosine learning rate decay and learning rate warm-up are effective strategies for text detection.

FPGM Pruner	HMean	Model Size (M)	Inference Time (SD 855, ms)
	0.6239	2.6	164
√	0.6169	1.4	133

Table 6: Ablation study of FPGM pruner for text detection.

Backbone	Accuracy	Model Size (M)	Inference Time (CPU, ms)
MobileNetV3_small_x0.5	0.9494	1.34	3.22
MobileNetV3_small_x0.35	0.9403	0.85	3.21
ShuffleNetV2_x0.5	0.9017	1.72	3.41

Table 7: Compares the performance of the different backbones for direction classification.

Table 6 shows the ablation study of FPGM pruner for text detection. Using FPGM pruner, the model size is reduced 46.2% and the inference time has accelerated 18.9% on SD 855 device with HMean slightly dropped. Therefore, FPGM pruner can prune the text detection model effectively.

3.3 Direction Classification

Table 7 compares the performance of different backbones for direction classification. The accuracy of MobileNetV3 with difference scales $(0.35, 0.5)$ are close. The model size and the inference time of MobileNetV3_small_x0.35 are much better. Besides, ShuffleNetV2 is used to train a direction classifier in some previous work. From the table, whether it's accuracy or the model size or the inference time, ShuffleNetV2 is not a good choice.

Tabel 9 shows the ablation study of data augmentation for direction classification. The baseline accuracy of text director classify without data augmentation is only 88.79%. When we adopt BDA (base data augmentation), the accuracy can boost 2.55%. We also verified that RandomErasing

Input Resolution	PACT Quantization	Accuracy	Model Size (M)	Inference Time (SD 855, ms)
$3 \times 32 \times 100$		0.9212	0.85	3.19
$3 \times 48 \times 192$		0.9403	0.85	3.21
$3 \times 48 \times 192$	$\surd$	0.9456	0.46	2.38

Table 8: Ablation study of input resolution and PACT quantization for direction classification.

Data Augmentation	Accuracy
NO	0.8879
BDA	0.9134
BDA+CutMix	0.9083
BDA+Mixup	0.9104
BDA+Cutout	0.9081
BDA+HideAndSeek	0.8598
BDA+GridMask	0.9140
BDA+RandomErasing	0.9193
BDA+AutoAugment	0.9133
BDA+RandAugment	0.9212

Table 9: Ablation study of data augmentation for direction classification.

Backbone	Accuracy	Model Size (M)	Inference Time (CPU, ms)
MobileNetV3_small_x0.35	0.6288	22	17
MobileNetV3_small_x0.5	0.6556	23	17.27
MobileNetV3_small_x1	0.6933	28	19.15

Table 10: Compares the performance of the different backbones for text recognition. The number of channel in the head is 256.

and RandAugment are useful for text direction classification. Therefore, in PP-OCR, we use BDA (base data augmentation) and RandAugment to train a direction classifier.

Table 8 shows the ablation study of input resolution and PACT quantization for direction classification. When the input resolution is adjusted from $3 \times 32 \times 100$ to $3 \times 48 \times 192$, The classification accuracy has improved but the prediction speed is basically unchanged. Furthermore, we also verified quantization strategy is effective in accelerating the prediction speed of the text direction classifier. The model size is reduced 45.9% and the inference time has accelerated 25.86%. Accuracy is slight promotion.

3.4 Text Recognition

Table 10 compares the performance of the different backbones for text recognition. The accuracy, the model size and the inference time of the different scales of MobileNetV3 change greatly. In PP-OCR, we choose MobileNetV3_small_x0.5 to balance accuracy and efficiency.

the number of channel	Accuracy	Model Size (M)	Inference Time (CPU, ms)
256	0.6556	23	17.27
96	0.6673	8	13.36
64	0.6642	5.6	12.64
48	0.6581	4.6	12.26

Table 11: Ablation study of the number of channel in the head for text recognition. The data augmentation is only used BDA.

Table 11 compares the number of channel in the CRNN head for text recognition. Reduce the number of channel from 256 to 48, the model size is reduced from 23M to 4.6M and the inference time has accelerated nearly 30%. However, the accuracy will not be affected. We can see the number of channel in the head has a great influence on the model size of a lightweight text recognizer.

Tabel 12 shows the ablation study of data augmentation, cosine learning rate decay, the stride of the second down sampling feature map, regularization parameters L2_decay and learning rate warm-up for text recognition.

To verify the advantages of each strategy, the setting of the basic experimental is the strategy S1. When using BDA, the accuracy will be improved 3.12%. Data augmentation is very necessary for text recognition. When we adopt the cosine learning rate decay further, the accuracy will be improved 1.47%. The cosine learning rate is an effective strategy for text recognition. Next, when we increase the feature map resolution and reduce the stride of the second down sampling feature map from (2,1) to (1,1), the accuracy will be improved 5.27%. Then, when we adjust the regularization parameters L2_decay from 0 to $1e-5$ further, the accuracy will be improved 3.4%. The feature map resolution and L2_decay have a great influence on the performance. Final, using learning rate warm-up, the accuracy will be improved 0.62%. Using TIA data augmentation, the accuracy will be improved 0.91%. Learning rate warm-up and TIA also are effective strategies for text recognition.

Tabel 13 shows the ablation study of PACT quantization for text recognition. When we use PACT quantization, the model size is reduced 67.39% and the inference time has accelerated 8.3%. Since there was no quantification on LSTM, The acceleration is not obvious. However, accuracy achieves a significant improvement. Therefore, PACT quantization also is an effective strategy for reducing the model size of a text recognizer.

In the end, we will illustrate the effect of pre-trained

Strategy	Data Augmentation	Cosine Learning Rate Decay	Stride	L2_decay	Learning Rate Warm-up	Accuracy	Inference Time (CPU, ms)
S1	NO		(2,1)	0		0.5193	11.84
S2	BDA		(2,1)	0		0.5505	11.84
S3	BDA	$\surd$	(2,1)	0		0.5652	11.84
S4	BDA	$\surd$	(1,1)	0		0.6179	12.96
S5	BDA	$\surd$	(1,1)	$1e-5$		0.6519	12.96
S6	BDA	$\surd$	(1,1)	$1e-5$	$\surd$	0.6581	12.96
S7	BDA+TIA	$\surd$	(1,1)	$1e-5$	$\surd$	0.6670	12.96

Table 12: Ablation study of data augmentation, cosine learning rate decay, the stride of the second down sampling feature map, regularization parameters L2_decay and learning rate warm-up for text recognition. Backbone is MobileNetV3_small_x0.5. The number of channel in the head is 48.

PACT Quantization	Accuracy	Model Size (M)	Inference Time (SD 855, ms)
	0.6581	4.6	12
$\surd$	0.674	1.5	11

Table 13: Ablation study of PACT quantization for text recognition.

Slim	F-score	Model Size (M)	Inference Time (SD 855, ms)
	0.5193	8.1	306
$\surd$	0.5210	3.5	268

Table 14: Ablation study of the prunner or quantization for the OCR system.

model. We utilize 17.9M training images to learn a text recognizer. Then, use this model as the pre-trained model to fine-tuning the samples for the ablation experiments. When using above pre-trained model, the accuracy will go from 65.81% to 69% and the effect is very obvious.

3.5 System Performance

Table 14 shows the ablation study of the prunner or quantization for the OCR system. When we use the slim approaches, the model size is reduced 55.7% and the inference time has accelerated 12.42%. F-score has no impact. The inference time includes pre-process and post-process of each parts of the system. Therefore, FPGM pruner and PACT quantization also are effective strategies for reducing the model size.

To compare the gap between the proposed ultra lightweight OCR system and large-scale OCR system, we also train a large-scale OCR system and use Res18_vd as the text detector backbone and Res34_vd as the text recognizer backbone. Table 15 shows the comparison. F-score of the large-scale OCR system is higher than the ultra lightweight OCR system, but the model size and the inference time of the ultra lightweight system are better obviously.

Figure 13 and Figure 14 show some image results of the proposed PP-OCR system for Chinese and English recognition. Figure 15 show some image results of the proposed PP-OCR system for multilingual recognition.

Model Type	F-score	Model Size (M)	Inference Time (ms)	
			CPU	T4 GPU
Ultra lightweight	0.5193	8.1	421	137
Large scale	0.5414	155.1	1199	204

Table 15: Compare between the ultra lightweight OCR system and the large scale one.

4 Conclusions

In this paper, we propose a practical ultra lightweight OCR system, PP-OCR, which the overall model size is only 3.5M for recognizing 6622 Chinese characters and 2.8M for recognizing 63 alphanumeric symbols. We introduce a bag of strategies to either enhance the model ability or light the model. The corresponding ablation experiments are also provided. Meanwhile, some practical ultra lightweight OCR models are released with a large-scale dataset.

References

Chen, P. 2020. GridMask data augmentation. *arXiv preprint arXiv:2001.04086* . 2.2

Choi, J.; Wang, Z.; Venkataramani, S.; Chuang, P. I.-J.; Srinivasan, V.; and Gopalakrishnan, K. 2018. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085* . 2.2

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 113–123. 2.2

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 702–703. 2.2

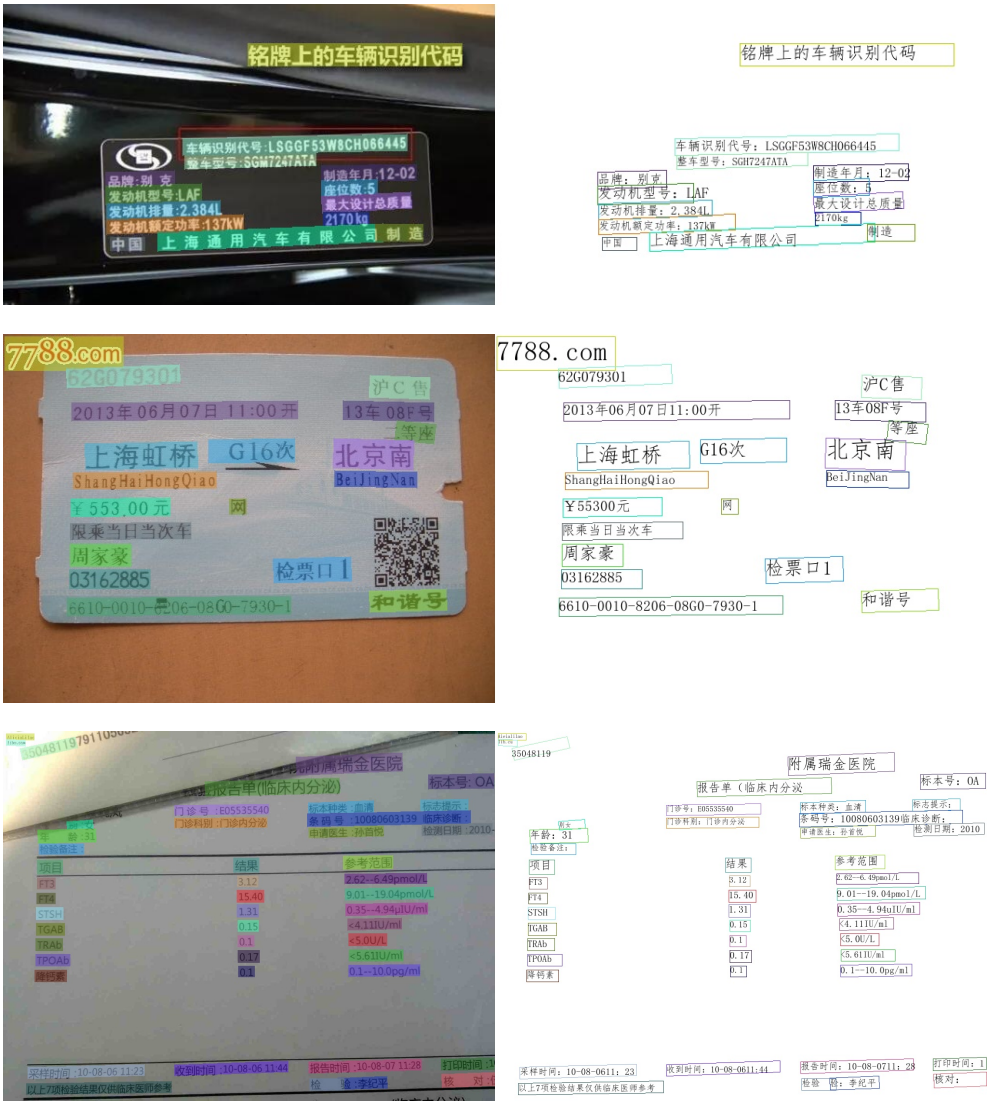


Figure 13: Some image results of the proposed PP-OCR system for Chinese and English recognition.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552* . 2.2

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2315–2324. 3.1

He, M.; and Yang, Z. 2018. ICPR 2018 contest on robust reading for multi-type web images (MTWI). https://tianchi.aliyun.com/competition/entrance/231651/information. 3.1

He, T.; Zhang, Z.; Zhang, H.; Zhang, Z.; Xie, J.; and Li, M. 2019a. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 558–567. 2.1

He, W.; Zhang, X.-Y.; Yin, F.; and Liu, C.-L. 2018. Multi-oriented and multi-lingual scene text detection with direct regression. *IEEE Transactions on Image Processing* 27(11): 5406–5419. 3.1

He, Y.; Liu, P.; Wang, Z.; Hu, Z.; and Yang, Y. 2019b. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4340–4349. 9, 2.1

Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. 2019. Searching for mobilenetv3. In *Proceedings of the IEEE International Conference on Computer Vision*, 1314–1324. 12

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141. 7, 2.1

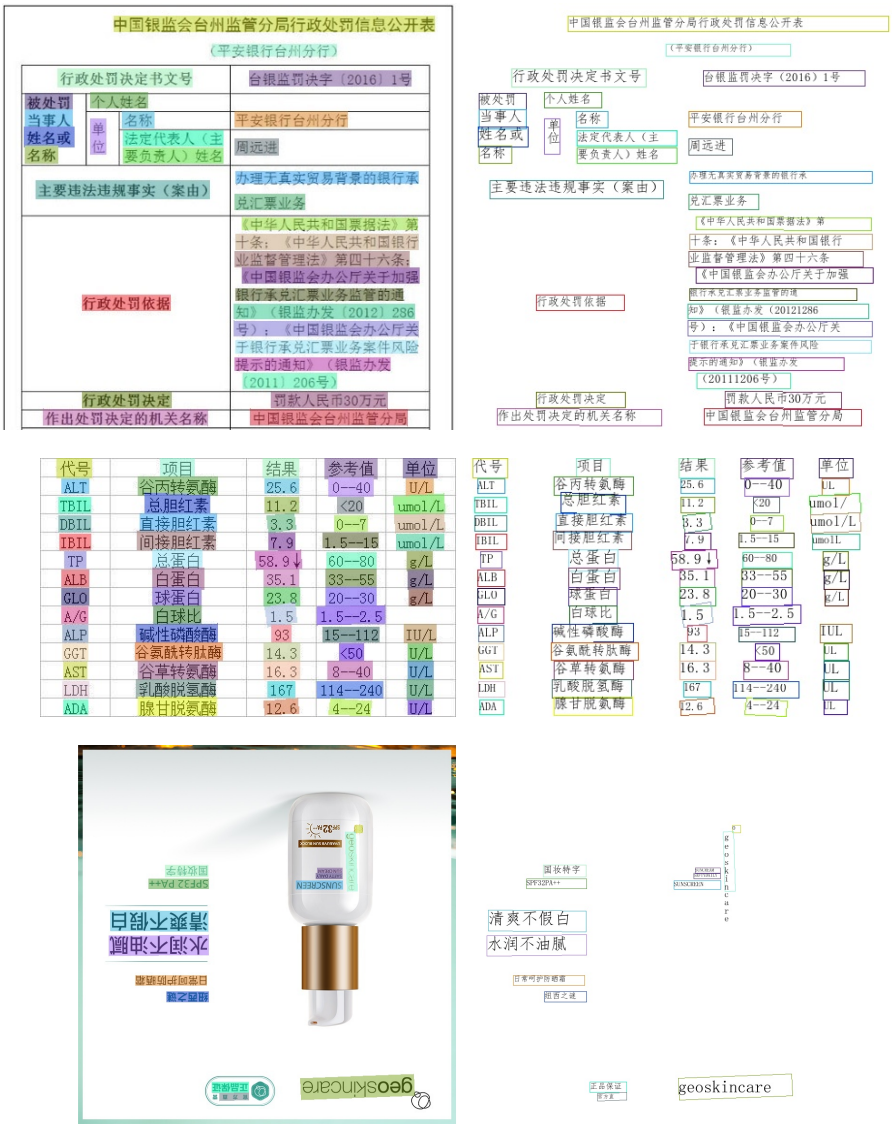


Figure 14: Some image results of the proposed PP-OCR system for Chinese and English recognition.

Huang, Z.; Chen, K.; He, J.; Bai, X.; Karatzas, D.; Lu, S.; and Jawahar, C. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1516–1520. IEEE. 3.1

Karatzas, D.; Mestre, S. R.; Mas, J.; Nourbakhsh, F.; and Roy, P. P. 2011. ICDAR 2011 robust reading competition-challenge 1: reading text in born-digital images (web and email). In *2011 International Conference on Document Analysis and Recognition*, 1485–1490. IEEE. 3.1

Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; and Graf, H. P. 2016. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710* . 2.1

Liao, M.; Wan, Z.; Yao, C.; Chen, K.; and Bai, X. 2020. Real-Time Scene Text Detection with Differentiable Binarization. In *AAAI*, 11474–11481. 1, 5

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125. 2.1

Luo, C.; Zhu, Y.; Jin, L.; and Wang, Y. 2020. Learn to Augment: Joint Data Augmentation and Network Optimization for Text Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13746–13755. 2.3, 11

Nayef, N.; Patel, Y.; Busta, M.; Chowdhury, P. N.; Karatzas, D.; Khlif, W.; Matas, J.; Pal, U.; Burie, J.-C.; Liu, C.-l.; et al. 2019. ICDAR2019 robust reading challenge on multi-lingual scene text detection and recognitionRRC-

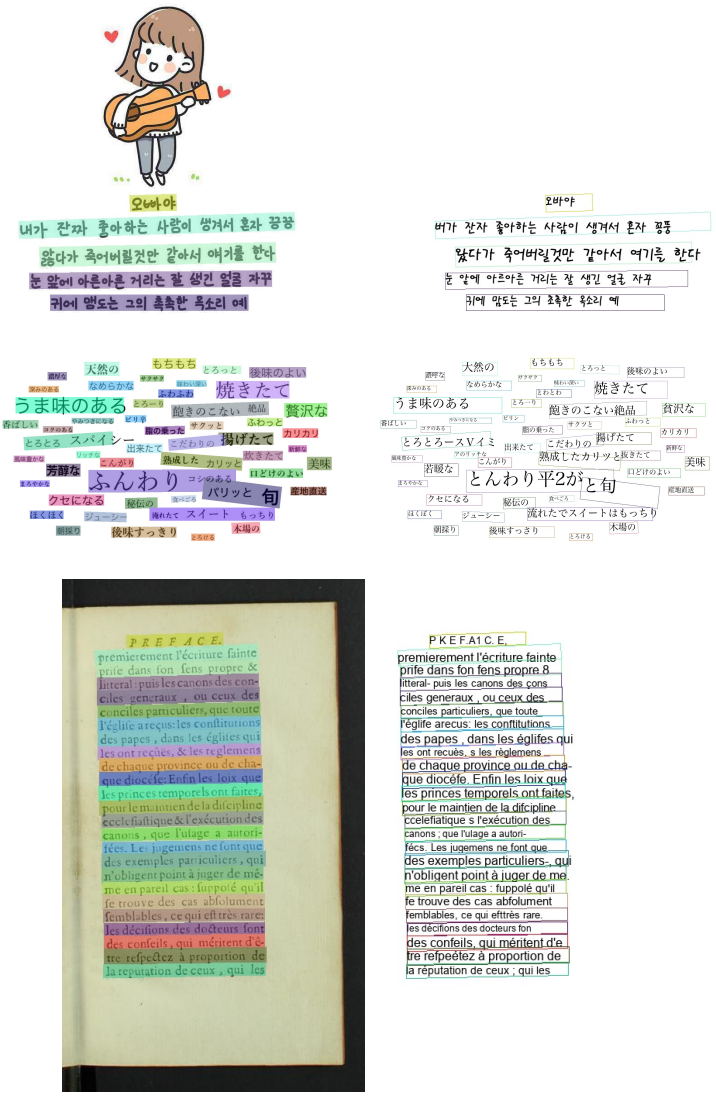


Figure 15: Some image results of the proposed PP-OCR system for multilingual recognition.

MLT-2019. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1582–1587. IEEE. 3.1

Sanster. 2018. Generate text images for training deep learning ocr model. https://github.com/Sanster/text_renderer. 3.1

Shi, B.; Bai, X.; and Yao, C. 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence* 39(11): 2298–2304. 1, 10

Shi, B.; Yao, C.; Liao, M.; Yang, M.; Xu, P.; Cui, L.; Belongie, S.; Lu, S.; and Bai, X. 2017. ICDAR2017 competition on reading chinese text in the wild (RCTW-17). In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, 1429–1434. IEEE. 3.1

Singh, K. K.; and Lee, Y. J. 2017. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *2017 IEEE international conference on computer vision (ICCV)*, 3544–3553. IEEE. 2.2

Sun, Y.; Liu, J.; Liu, W.; Han, J.; Ding, E.; and Liu, J. 2019. Chinese street view text: Large-scale chinese text reading with partially supervised learning. In *Proceedings of the IEEE International Conference on Computer Vision*, 9086–9095. 3.1

Xu, Z.; Yang, W.; Meng, A.; Lu, N.; Huang, H.; Ying, C.; and Huang, L. 2018. Towards end-to-end license plate detection and recognition: A large dataset and baseline. In *Proceedings of the European conference on computer vision (ECCV)*, 255–271. 3.1

Yao, C.; Bai, X.; Liu, W.; Ma, Y.; and Tu, Z. 2012. Detecting texts of arbitrary orientations in natural images. In *2012*

IEEE conference on computer vision and pattern recognition, 1083–1090. IEEE. 3.1

Yu, D.; Li, X.; Zhang, C.; Liu, T.; Han, J.; Liu, J.; and Ding, E. 2020. Towards accurate scene text recognition with semantic reasoning networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12113–12122. 2.2, 3.1

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, 6023–6032. 2.2

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* . 2.2

Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y. 2020. Random Erasing Data Augmentation. In *AAAI*, 13001–13008. 2.2