PP-OCRv2: Bag of Tricks for Ultra Lightweight OCR System

Yuning Du, Chenxia Li, Ruoyu Guo, Cheng Cui, Weiwei Liu, Jun Zhou, Bin Lu, Yehua Yang, Qiwen Liu, Xiaoguang Hu, Dianhai Yu, Yanjun Ma

Baidu Inc.

{duyuning, yangyehua}@baidu.com

Abstract

Optical Character Recognition (OCR) systems have been widely used in various of application scenarios. Designing an OCR system is still a challenging task. In previous work, we proposed a practical ultra lightweight OCR system (PP-OCR) to balance the accuracy against the efficiency. In order to improve the accuracy of PP-OCR and keep high efficiency, in this paper, we propose a more robust OCR system, i.e. PP-OCRv2. We introduce bag of tricks to train a better text detector and a better text recognizer, which include Collaborative Mutual Learning (CML), CopyPaste, Lightweight CPU Network (PP-LCNet), Unified-Deep Mutual Learning (U-DML) and Enhanced CTCLoss. Experiments on real data show that the precision of PP-OCRv2 is 7% higher than PP-OCR under the same inference cost. It is also comparable to the server models of the PP-OCR which uses ResNet series as backbones. All of the above mentioned models are open-sourced and the code is available in the GitHub repository PaddleOCR [1] which is powered by PaddlePaddle [2].

1 Introduction

OCR (Optical Character Recognition) in the wild, as shown in Figure 1, is well-studied in the last two decades and has various applications scenarios, such as document electronization, identity authentication, digital financial system, and vehicle license plate recognition.

When we build an OCR system in practical, not only the accuracy is considered, but also the computational efficiency. In previous, we proposed a practical ultra lightweight OCR system (PP-OCR) (Du et al. 2020) to balance the accuracy against the efficiency. It consists of three parts, text detection, detected boxes rectification and text recognition. Differentiable Binarization (DB) (Liao et al. 2020a) is used in text detection and CRNN (Shi, Bai, and Yao 2016) is used in text recognition. The system adopts 19 effective strategies to optimize and slim down the size of the models. In order to improve the accuracy of the PP-OCR and keep efficiency, in this paper, we propose a more robust OCR system, i.e. PP-OCRv2. It introduces bag of tricks to train a better text detector and a better text recognizer.

Figure 2 illustrates the framework of PP-OCRv2. Most strategies follow PP-OCR as shown in the green boxes. The strategies in the orange boxes are the additional ones in PP-OCRv2. In text detection, Collaborative Mutual Learning (CML) and CopyPaste are introduced. CML utilizes two student networks and a teacher network to learn a more robust text detector. CopyPaste is a novel data augmentation trick that has been proved effectively boost performance of object detection and instance segmentation tasks (Ghiasi et al. 2021). We show that it also works well for text detection task. In text recognition, Lightweight CPU Network (PP-LCNet)(Cui et al. 2021), Unified-Deep Mutual Learning (U-DML) and CenterLoss are introduced. PP-LCNet is a newly designed lightweight backbone based on Intel CPUs which is modified from MobileNetV1(Howard et al. 2017). U-DML utilizes two student networks to learn a more accurate text recognizer. The role of the CenterLoss is to relax the mistakes of the similar characters. We conduct a series of ablation experiments to verify the effectiveness of the above strategies.

Figure 1: Some recognition results of the proposed PP-OCRv2 system

Besides, the strategies in the gray boxes of Figure 2 are demonstrated to be effective in PP-OCR. But those are not validated in this paper. In the future, we will adopt them to speed up the inference in PP-OCRv2-tiny.

The rest of the paper is organized as follows. In section 2, we present the details of the newly added enhancement strategies. Experimental results are discussed in section 3

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

[2] https://github.com/PaddlePaddle

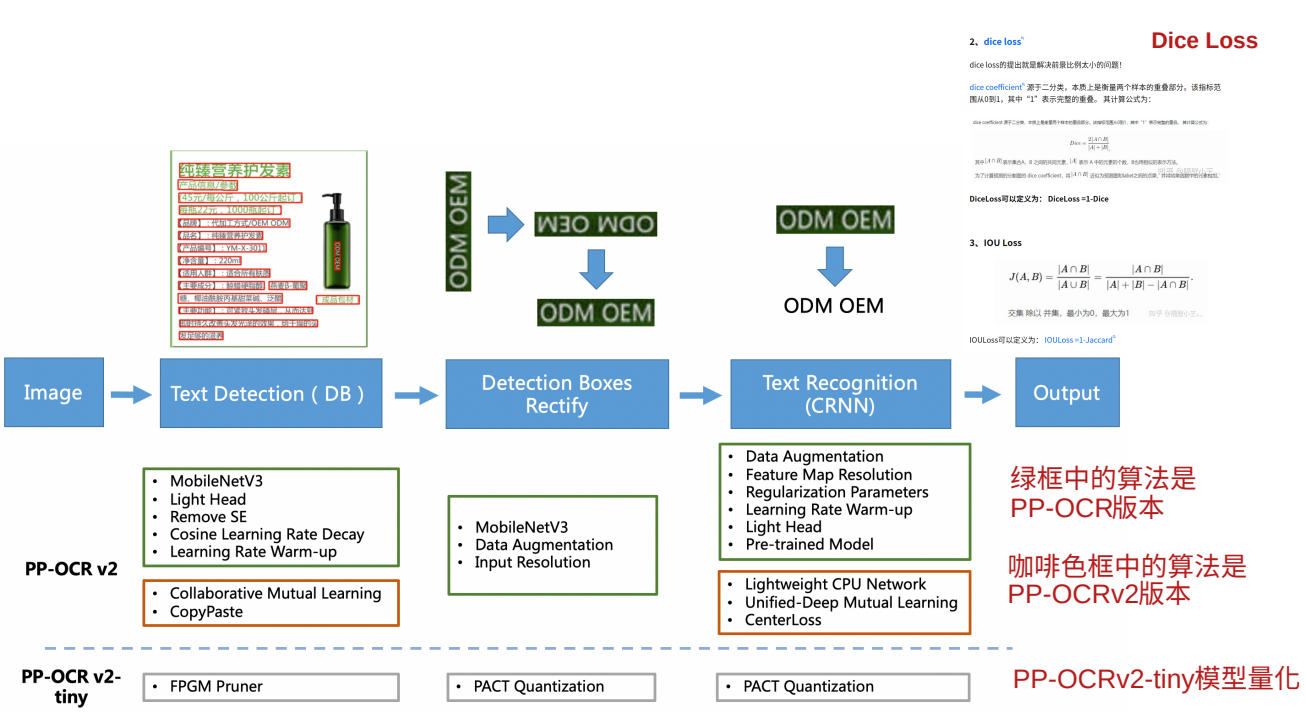


Figure 2: The framework of the proposed PP-OCRv2. The strategies in the green boxes are the same as PP-OCR. The strategies in the orange boxes are the newly added ones in the PP-OCRv2. The strategies in the gray boxes will be adopted by the PP-OCRv2-tiny in the future.

and conclusions are conducted in section 4.

2 Enhancement Strategies

2.1 Text Detection

Collaborative Mutual Learning (CML) We propose the CML method (Zhang et al. 2017) to solve the problem of text detection distillation. There are two problems with distillation: 1. If the accuracy of the teacher model is close to that of the student model, the improvement brought by the general distillation method is limited. 2. If the structure of the teacher model and the structure of the student model are quite different, the improvement brought by the general distillation method is also very limited.

目前蒸馏策略存在的问题

The framework is super network composed of multiple model named student models and teacher models respectively, as illustrated in 3. And the CML method can achieve the performance that the accuracy of the student after distillation exceeds the accuracy of the teacher model in text detection.

In CML, two sub-student models learn from each other using DML method (Zhang et al. 2017). Meanwhile, there is a teacher model to guide the learning of two student models. The teacher model uses ResNet18 as the backbone, and the student model uses MobinenetV3 large model with scale 0.5 as the backbone.

CML在DML的基础上添加了已训练好的老师模型用来指导两个学生模型进行训练

CML aims to optimize sub-student model. The parameters of teacher model are freezed and only sub-student model is trained with designed losses. In general, the supervised information of the sub-student model has three parts, including the ground truth label, the posterior entropy of another student model and output of the teacher model. Correspondingly, there are three loss functions including the ground truth loss L_{gt}, peer loss from student model L_s and distill loss from teacher model L_t.

备注：如果要使用CML算法，则先在数据集合上训练一个大的模型

The ground truth loss, referred as GTLoss, is to make sure that the training is supervised by the true label. We use the DB algorithm (Liao et al. 2020b) to train the sub-student models. Therefore, the ground truth loss L_{gt} is a combined loss, which consists of the loss of the probability map l_p, the loss of the binary map l_b, and the loss of the threshold map l_t following DB.The formula of GTLoss is as follows, where l_p, l_b and l_t are binary cross-entropy loss, Dice loss and L1 loss respectively. α, β are the super-parameters with default values 5 and 10 respectively.

$$Loss_{gt}(T_{out}, gt) = l_p(S_{out}, gt) + \alpha l_b(S_{out}, gt) + \beta l_t(S_{out}, gt) \quad (1)$$

GTLoss使用了DBLoss

The sub-student models learn from each other with reference to DML method (Zhang et al. 2017). But the difference with DML is that the sub-student models is trained simultaneously in every iteration to speed up the training process. KL divergence is used to compute the distance between student models. The peer loss between student models is as follows,

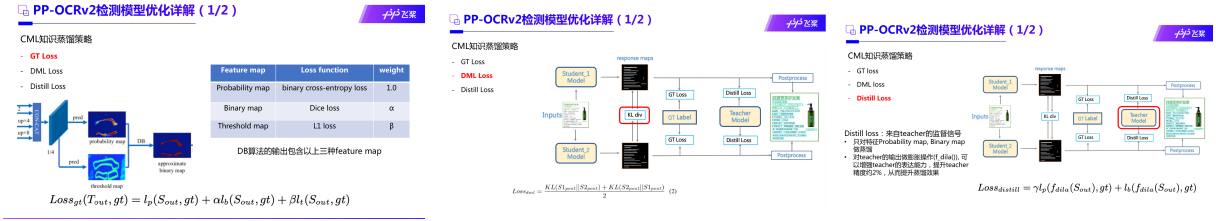
DML损失使用了两个学生之间模型输出的KL散度计算

$$Loss_{dml} = \frac{KL(S1_{pout}||S2_{pout}) + KL(S2_{pout}||S1_{pout})}{2} \quad (2)$$

The distill loss reflects the supervision of the teacher model on the sub-student models. Teacher model can provide a wealth of knowledge to student models which is important for performance improvement. To get better knowledge, we dilate the response probability maps of the teacher model to increase the object area. This operation can slightly improve the accuracy of the teacher model. The distillation loss is as follows, where l_p, l_b are binary cross-entropy loss and Dice Loss respectively. And γ is the super-parameter default as 5. The f_{dila} is the dilation function which kernel is matrix $[[1, 1], [1, 1]]$.

$$Loss_{distill} = \gamma l_p(S_{out}, f_{dila}(T_{out})) + l_b(S_{out}, f_{dila}(T_{out})) \quad (3)$$

CML损失函数由三部分组成



ch_PP-OCRv2_det_cml.yml配置文件中使用了ResNet18作为老师的Backbone，两个学生使用了相同的MobileNetV3作为Backbone，老师和学生的Neck以及Head都分别使用了DBFPN以及DBHead

结合PaddleOCR训练的配置文件来看，老师和学生相同的网络结构应该是指Neck以及Head，但各自的Backbone可以有所差异

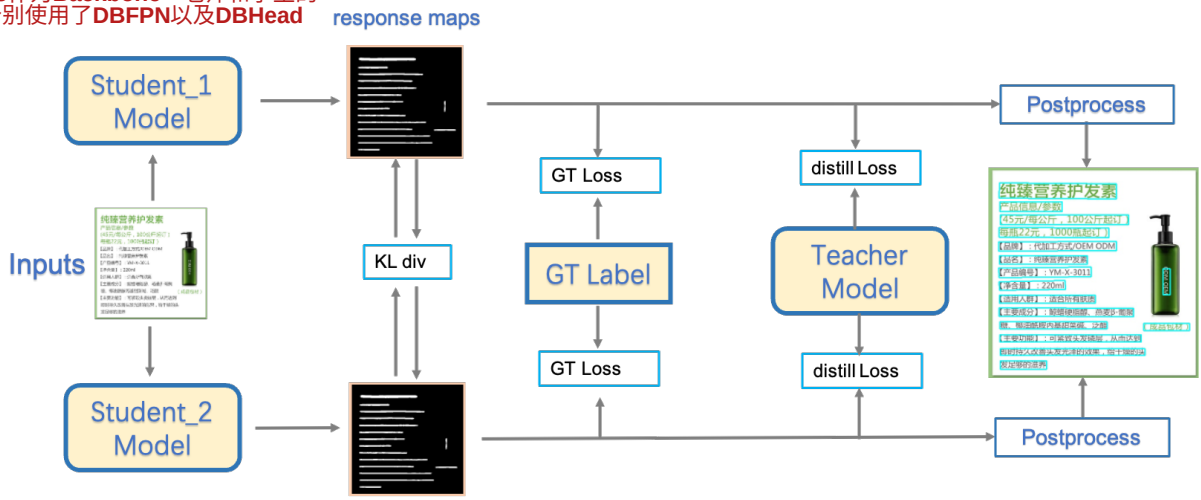


Figure 3: The framework of CML

Finally, the loss function used in the CML method of training the PP-OCR detection model is as follows.

$$Loss_{total} = Loss_{gt} + Loss_{dml} + Loss_{distill} \quad (4)$$

CopyPaste is a novel data augmentation trick that has been proved to be effective in boosting performance of object detection and instance segmentation tasks (Ghiasi et al. 2021). It can synthesize text instances to balance the ratio between positive and negative samples in the training set, which traditional image rotation, random flip and random cropping cannot achieve. Due to all texts in the foreground being independent, CopyPaste pastes texts without overlapping on a randomly selected background image. Figure 4 is an example of CopyPaste.

Figure 4: Example of CopyPaste in text detection. The green boxes are pasted text, the red boxes are the original texts in the image.

2.2 Text Recognition

Lightweight CPU Network (PP-LCNet) In order to get a better accuracy-speed trade-off on Intel CPU, we have designed a lightweight backbone based on Intel CPUs, which provides a faster and more accurate OCR recognition algorithm with mkldnn enabled. The structure of the entire network is shown in Figure 5. Compared with MobileNetV3, as the structure of MobileNetV1 makes it easier to optimize the inference speed when MKLDNN is enabled on Intel CPU, so the network is based on MobileNetV1(Howard et al. 2017).In order to make MobileNetV1 have a stronger ability to extract features, we have made some changes to its network structure. The improvement strategy will be explained in the following four aspects.

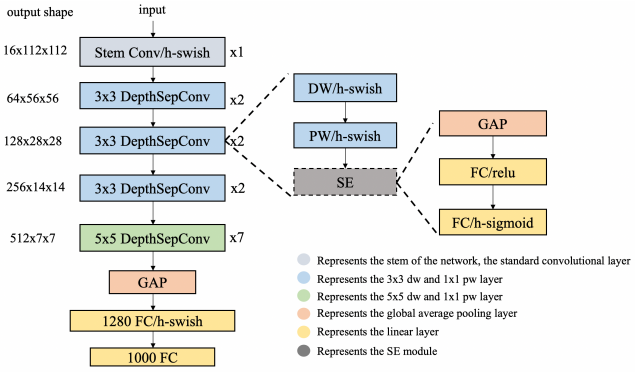


Figure 5: PP-LCNet network structure. The dotted box represents optional modules. The stem part uses standard convolution. DepthSepConv means depthwise separable convolutions, DW means depthwise convolutions, PW means pointwise convolutions, GAP means global average pooling.

DepthSepConv操作流程：先做Depthwise后做Pointwise

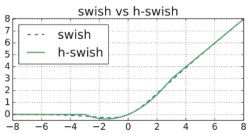
1. Better activation function. In order to increase the fitting ability of MobileNetV1, we replaced the activation

Hard-Swish 激活函数

Hard Swish is a type of activation function based on Swish, but replaces the computationally expensive sigmoid with a piecewise linear analogue:

$$\text{h-swish}(x) = x\frac{\text{ReLU6}(x+3)}{6}$$

Source: Searching for MobileNetV3



function in the network with H-Swish from the original ReLU, which can bring a significant improvement in accuracy with only a slight increase in inference time.

2. SE modules at appropriate positions. SE (Hu, Shen, and Sun 2018) module has been used by a large number of networks since its being proposed. It is a good way to weight the network channels to obtain better features, and is used in many lightweight networks such as MobileNetV3(Howard et al. 2019). However, on Intel CPUs, the SE module increases the inference time, so that we cannot use it for the whole network. In fact, through extensive experiments, we have found that the closer to the tail of the network, the more effective the SE module is. So we just add the SE module to the blocks near the tail of the network. This results in a better accuracy-speed balance. The activation functions of the two layers in the SE module are ReLU and H-Sigmoid respectively.

经过试验发现在网络尾部的添加SE模块会达到精度与速度的平衡

3. Larger convolution kernels. The size of the convolution kernel often affects the final performance of the network. In mixnet(Tan and Le 2019), the authors analyzed the effect of differently sized convolution kernels on the performance of the network, and finally mixed differently sized kernels in the same layer of the network. However, such a mixture slows down the inference speed of the model, so we tried to increase the size of the convolution kernels with as little increase in inference time as possible. In the end, we set the size of the convolution kernel at the tail of the network as 5×5.

在网络尾部使用5x5的卷积核也为了兼顾精度的情况下稍微增加了推理速度

4. Larger dimensional 1x1 conv layer after GAP. In PP-LCNet, the output dimension of the network after GAP is small, and directly connecting the final classification layer will lose the combination of features. In order to give the network a stronger fitting ability, we connected a 1280-dimensional size 1x1 conv to the final GAP layer, which would increase the model size without increasing the inference time.

GAP后网络的输出维度比较小，如果直接连接到最终的分类层将丢失特征的融合信息。为了提升网络拟合能力，使用了1280个1x1卷积核来连接GAP层，这样的好处是在不增加推理时间的情况下只稍微增大了模型的大小。

With these four changes, our model performs well on the ImageNet, and table 4 lists the metrics against other lightweight models on Intel CPUs.

DML没有使用更大的老师网络去做蒸馏

Unified-Deep Mutual Learning (U-DML) Deep mutual learning (Zhang et al. 2017) is a method in which two student networks learn by each other, and a larger teacher network with pre-trained weights is not required for knowledge distillation. In DML, for image classification task, the loss functions contains two parts: (1) loss function between student networks and groundtruth. (2) Kullback–Leibler divergence (KL-Div) loss among the student networks' output soft labels.

DML损失函数由两部分组成 1. 两个学生网络的输出分别与真实值计算损失， 2. 使用KL散度衡量两个学生网络输出的差异程度作为损失函数

Heo proposed OverHaul (Heo et al. 2019), in which feature map distance between student network and teacher network are used for the distillation process. Transform is carried out on student network feature map to keep the feature map alignment.

To avoid too time-consuming teacher model training process, in this paper, based on DML, we proposed U-DML, in which feature maps are also supervised during the distillation process. Figure 6 shows the framework of U-DML.

There are two networks for the distillation process: the student network and the teacher network. They have exactly the same network structures with different initialized weights. The goal is that for the same input image, the two networks can get the same output, not only for the prediction result but also for the feature map.

ch_PP-OCRv2_rec_distillation.yml配置文件中的老师和学生的网络结构是完全一样的，

The total loss function consists of three parts: (1) CTC loss. Since the two networks are trained from scratch, CTC loss can be used for the networks' convergence; (2) DML loss. It's expected that two the networks' final output distributions are same, so DML loss is needed to guarantee the consistency of distribution between the two networks; (3) Feature loss. The two networks' architectures are same, so their feature maps are expected to be same, feature loss can be used to constrain the two networks' intermediate feature map distance.

CTC loss. CRNN is the base architecture for text recognition in this paper, which integrates feature extraction and sequence modeling. It adopts the Connectionist Temporal Classification (CTC) loss (Graves et al. 2006) to avoid the inconsistency between prediction and groundtruth. Since the two sub-networks are trained from scratch, CTC loss is adopted for the two sub-networks. The loss function is as follows.

$$Loss_{ctc} = CTC(S_{hout}, gt) + CTC(T_{hout}, gt) \tag{5}$$

in which S_{hout} denotes head output of the student network and T_{hout} denotes that of the teacher network. gt donates the groudtruth label of the input image.

DML loss. In DML, parameters of each sub-network are updated separately. Here, to simplify the training process, we calculate the KL divergence loss between the two sub-networks and update all the parameter simultaneously. The DML loss is as follows.

$$Loss_{dml} = \frac{KL(S_{pout}||T_{pout}) + KL(T_{pout}||S_{pout})}{2} \tag{6}$$

in which $KL(p||q)$ denotes KL divergence of the p and q. S_{pout} and T_{pout} can be calculated as follows.

$$\begin{aligned} S_{pout} &= Softmax(S_{hout}) \\ T_{pout} &= Softmax(T_{hout}) \end{aligned} \tag{7}$$

Feature loss. During the training process, we hope that the backbone output of the student network is same as that of the teacher network. Therefore, similar to Overhaul, feature loss is used for the distillation process. The loss can be calculated as follows.

$$Loss_{feat} = L2(S_{bout}, T_{bout}) \tag{8}$$

in which S_{bout} means backbone output of the student network and T_{bout} means that of teacher network. Mean square error Loss is utilized here. It is noted that for the feature loss, feature map transformation is not needed because the two feature maps used to calculate the loss are exactly the same.

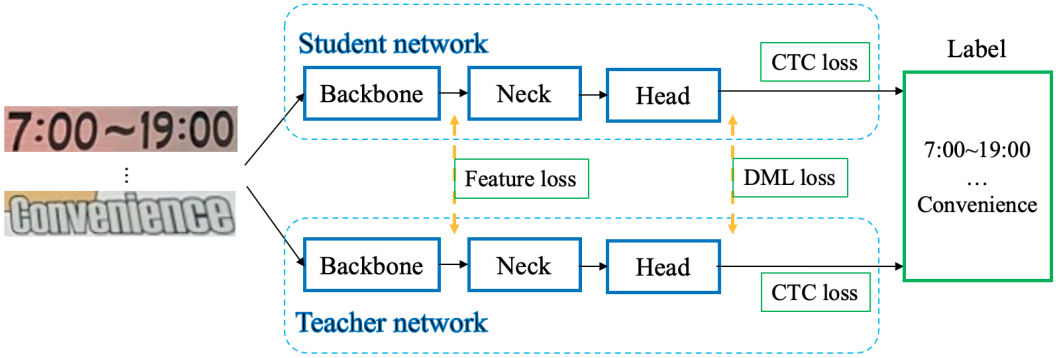


Figure 6: U-DML framework

Finally, the total loss for the U-DML training process is shown as follows.

$$Loss_{total} = Loss_{ctc} + Loss_{dml} + Loss_{feat} \tag{9}$$

分段学习策略

During the training process, we find that piece-wise learning rate strategy is a better choice for distillation. When the feature loss is used, it takes a longer time for the model to reach the best accuracy, so 800 epochs and piece-wise strategy are utilized here for the text recognition distillation process.

标准的CRNN结构的CTC-Head中只使用了一个全连接层，会导致信息解码能力弱，如果使用两个全连接层会让模型的精度提升1.5%，而且不增加模型的推理速度

Moreover, for the standard CRNN architecture, just one FC layer is used in CTC-Head, which is slightly weak for the information decoding process. Therefore, we modify the CTC-Head part, using two FC layers, this leads to about 1.5% accuracy improvement without any extra inference time cost.

Enhanced CTCLoss There exists a lot of similar characters in Chinese recognition task. Their differences in appearance are very small which are often mistakenly recognized. In PP-OCRv2, We designed an enhanced CTCLoss, which combined the original CTCLoss and the idea of CenterLoss (Wen et al. 2016) in metric learning. Some improvements are made to make it suitable for sequence recognition Task. Enhanced CTCLoss is defined as follows:

$$L = L_{ctc} + \lambda * L_{center} \tag{10}$$

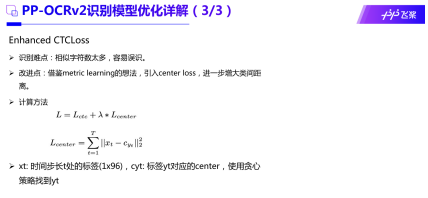
$$L_{center} = \sum_{t=1}^{T} ||x_t - c_{y_t}||_2^2 \tag{11}$$

时间戳

in which, x_t is the feature of timestamp t. c_{y_t} is the center of class y_t. We have no explicit label y_t for x_t because of the misalignment between features and labels in CRNN (Shi, Bai, and Yao 2016) algorithm. We adopt the greedy decoding strategy to get y_t:

$$y_t = argmax(W * x_t) \tag{12}$$

W is the parameters of CTC head. Experiments show that: $\lambda = 0.05$ is a good choice.


PP-OCRv2识别模型优化详解（3/3）

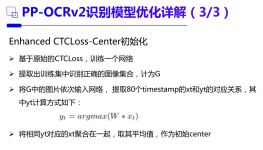
Enhanced CTCLoss

- 识别难点：相似字符数太多，容易误识。
- 改进点：借鉴metric learning的思法，引入center loss，进一步增大类间距离。
- 计算方法

$$L = L_{ctc} + \lambda * L_{center}$$

$$L_{center} = \sum_{t=1}^{T} ||x_t - c_{y_t}||_2^2$$

- xt: 时间步长t处的标签(1x96)，cyt: 标签yt对应的center，使用贪心策略找到yt


PP-OCRv2识别模型优化详解（3/3）

Enhanced CTCLoss-Center初始化

- 基于原始的CTCLoss，训练一个网络
- 提取出训练集中识别正确的图像集合，计为G
- 将G中的图片依次输入网络，提取80个timestamp的xt和yt的对应关系，其中yt计算方式如下：

$$y_t = argmax(W * x_t)$$

- 将相同yt对应的xt聚合在一起，取其平均值，作为初始center

CTCLoss-Center流程：
对文本框3x32x320大小的输入，到最后CTC head阶段，变成1x1x80的大小，80作为时间戳的长度，yt代表每个时间戳上的softmax，

3 Experiments

3.1 Experimental Setup

DataSets We perform experiments on the same datasets as we used in our previous work PP-OCR (Du et al. 2020) as shown in Table 1.

For text detection, there are 97k training images and 500 validation images. The training images consist of 68K real scene images and 29K synthetic images. The real scene images are collected from Baidu image search and public datasets 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). The synthetic images mainly focus on the scenarios for long texts, multi-direction texts and texts in table. The validation images are all from 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 scenarios for different backgrounds, rotation, perspective transformation, noising, vertical text, etc. The corpus of synthetic images comes from the real scene images. All the validation images also come from the real scenes.

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 7 and Figure 8 show some images of the test set.

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

Implementation Details We adopt most of the strategies used in PP-OCR (Du et al. 2020) as you can found in Figure 2. We use Adam optimizer to train all the models, setting the initial learning rate to 0.001. The difference is that we adopt cosine learning rate decay as the learning rate schedule for the training of detection model, but piece-wise decay

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

Table 1: Statistics of dataset for text detection and recognition.

Figure 7: Some images contained scene text

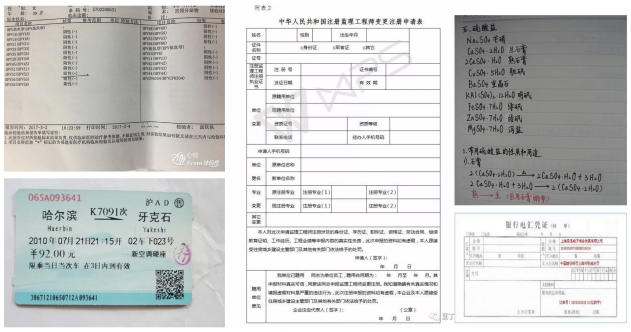


Figure 8: Some images contained document text

for recognition model training. Warm-up training for a few epochs at the beginning is utilized for both detection and recognition models training.

For text detection, the model is trained for 700 epochs in total with warm-up training for 2 epochs. The batch size is set to 8 per card. For text recognition, the model warm up for 5 epochs and is then trained for 700 epochs with the initial learning rate 0.001, and then trained for 100 epochs with learning rate decayed to 0.0001. The batch size is 128 per card.

In the inference period, Hmean is used to evaluate the performance of the text detector and the end-to-end OCR system. Sentence Accuracy is used to evaluate the performance of the text recognizer. GPU inference time is tested on a single T4 GPU. CPU inference time is tested on a Intel(R) Xeon(R) Gold 6148.

3.2 Text Detection

Table 2 shows the ablation study of DML, CML and CopyPaste for text detection. The baseline model is PP-OCR lightweight detection model. The long side of input image is resized to 960 during the test. As the data shows, DML can improve the Hmean metric by nearly 2%, while CML can improve by 3%. At last, the final Hmean can be further improved by 0.6% by the data augmentation method CopyPaste. So PP-OCRv2 detection model yields a 3.6% improvement over PP-OCR at the same speed, as the model structure stays the same. The inference time is the overall time consumed including pre-processing and post-processing.

3.3 Text Recognition

Table 3 shows the ablation study of PP-LCNet, U-DML and Enhanced CTC loss. Comparing PP-LCNet with MV3, the accuracy can be improved by 2.6%. Even though the model size with PP-LCNet is 3M bigger, the inference time is reduced from 7.7ms to 6.2ms due to the reasonable design of the network structure. The U-DML method can improve the accuracy by another 4.6%, which is a significant improvement. Further more, the accuracy can be improved by 0.9% with Enhanced CTC loss. So with all these strategies, the accuracy is improved by 8.1%, with model size 3M bigger but average inference time 1.5ms faster.

Ablation study for PP-LCNet In order to test the generalization ability of PP-LCNet, we used challenging datasets like ImageNet-1k throughout the process of designing the model. Table 4 shows the accuracy-speed comparison among the PP-LCNet and other different lightweight models that we have selected for comparable accuracy on ImageNet. It is obvious that PP-LCNet achieves better performance from both speed and accuracy, even when compared to a very competitive network like MobileNetV3.

3.4 System Performance

In Table 5, we compare the performance between proposed PP-OCRv2 with the previous ultra lightweight and large-scale PP-OCR system. The large-scale PP-OCR system, which uses ResNet18_vd as the text detector backbone and ResNet34_vd as the text recognizer backbone, can achieve higher Hmean but slower inference speed than the ultra lightweight version. It can be found that the Hmean of PP-OCRv2 is 7.3% higher than that of PP-OCR mobile models under the same inference cost, and is comparable to PP-OCR server models. Figure 9 visualizes some end-to-end recognition results of the proposed PP-OCRv2 system and the previous ultra lightweight and large-scale PP-OCR system.

4 Conclusions

In this paper, we proposed a more robust practical ultra lightweight OCR system PP-OCRv2. We introduced bag of tricks to enhance our previous work, PP-OCR, which include Collaborative Mutual Learning (CML), CopyPaste,

Strategy	Precision	Recall	Hmean	Model Size (M)	Inference Time (CPU, ms)
PP-OCR det	0.718	0.805	0.759	3.0	129
PP-OCR det + DML	0.743	0.815	0.777	3.0	129
PP-OCR det + CML	0.746	0.835	0.789	3.0	129
PP-OCR det + CML + CopyPaste	0.754	0.840	0.795	3.0	129

Table 2: Ablation study of CML and CopyPaste for text detection.

Strategy	Acc	Model Size (M)	Inference Time (CPU, ms)
PP-OCR rec (MV3)	0.667	5.0	7.7
PP-OCR rec (PP-LCNet)	0.693	8.0	6.2
PP-OCR rec (PP-LCNet) + U-DML	0.739	8.6	6.2
PP-OCR rec (PP-LCNet) + U-DML + Enhanced CTC loss	0.748	8.6	6.2

Table 3: Ablation study of PP-LCNet, U-DML, and Enhanced CTC loss for text recognition.

Model	Top1-Acc (%)	Inference Time (ms)
MobileNetV1-0.75x	68.81	3.88
MobileNetV2-0.75x	69.83	4.56
MobileNetV3-small-1.0x	68.24	4.20
MobileNetV3-large-0.5x	69.24	4.54
GhostNet-0.5x	66.88	6.63
PP-LCNet-1.0x	**71.32**	**3.16**

Table 4: Metrics of different lightweight models on ImageNet-1k, the CPU used in the test is Intel(R)-Xeon(R)-Gold-6148-CPU, the resolution of the image is 224x224, the batch-size is 1, the number of thread is 4, the option of MKLDNN is on, and the final inference time is the mean inference time of 30000 images.

Model Type	Hmean	Model Size (M)	Inference Time (ms)	
			CPU	T4 GPU
PP-OCR mobile	0.503	8.1	356	116
PP-OCR server	0.570	155.1	1056	200
PP-OCRv2 mobile	0.576	11.6	330	111

Table 5: Compare between PP-OCRv2 system and PP-OCR mobile and server systems.

Lightweight CPU Network (PP-LCNet), Unified-Deep Mutual Learning (U-DML) and CenterLoss. Experiments on the real data show that the accuracy of the PP-OCRv2 is higher than PP-OCR at the same inference cost. The corresponding ablation experiments are also provided. Meanwhile, some practical ultra lightweight OCR models are released with a large-scale dataset.

References

Cui, C.; Gao, T.; Wei, S.; Du, Y.; Guo, R.; Dong, S.; Lu, B.; Zhou, Y.; Lv, X.; Liu, Q.; Hu, X.; Yu, D.; and Ma, Y. 2021. PP-LCNet: A Lightweight CPU Convolutional Neural Network. 1

Du, Y.; Li, C.; Guo, R.; Yin, X.; Liu, W.; Zhou, J.; Bai, Y.; Yu, Z.; Yang, Y.; Dang, Q.; et al. 2020. PP-OCR: A practical ultra lightweight OCR system. *arXiv preprint arXiv:2009.09941* . 1, 3.1, 3.1

Ghiasi, G.; Cui, Y.; Srinivas, A.; Qian, R.; Lin, T.-Y.; Cubuk, E. D.; Le, Q. V.; and Zoph, B. 2021. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2918–2928. 1, 2.1

Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, 369–376. 2.2

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, 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

Heo, B.; Kim, J.; Yun, S.; Park, H.; Kwak, N.; and Choi, J. Y. 2019. A Comprehensive Overhaul of Feature Distillation. *CoRR* abs/1904.01866. URL http://arxiv.org/abs/1904.01866. 2.2

Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; Le, Q. V.; and Adam, H. 2019. Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2.2

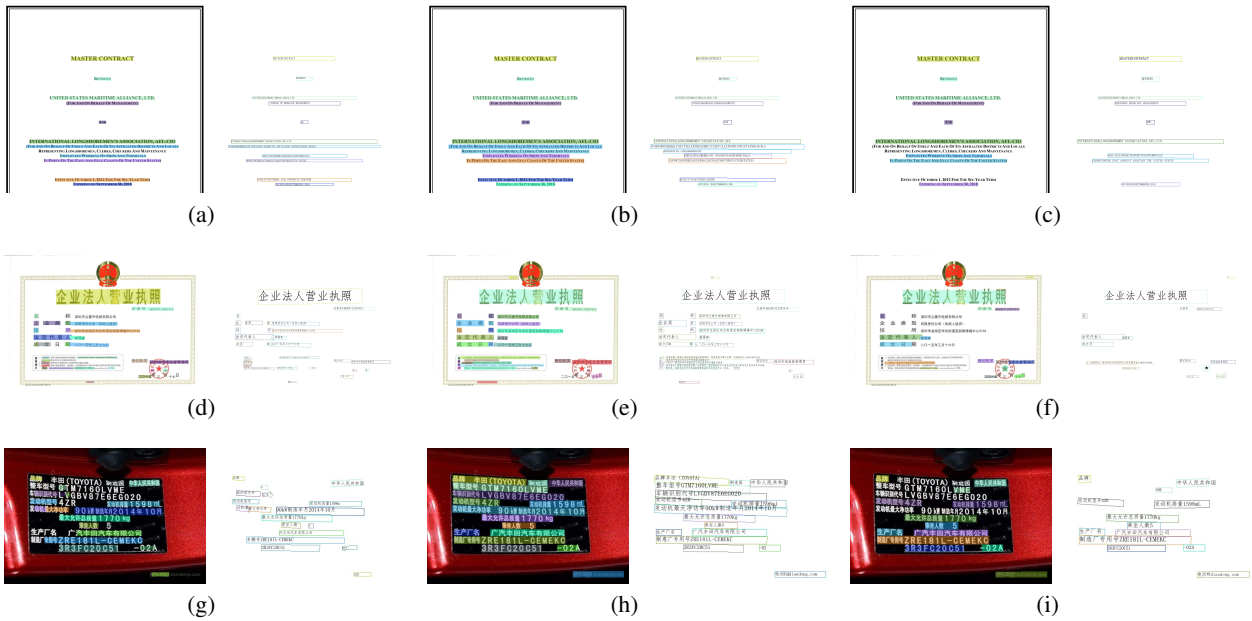


Figure 9: Some images results of the proposed PP-OCRv2 system and the previous ultra lightweight and large-scale PP-OCR system. (a)(d)(g) results of PP-OCR mobile system. (b)(e)(h) results of PP-OCRv2 system. (c)(f)(i) results of PP-OCR server system.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 1, 2.2

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-Excitation Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2.2

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

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

Liao, M.; Wan, Z.; Yao, C.; Chen, K.; and Bai, X. 2020b. Real-time Scene Text Detection with Differentiable Binarization. In *Proc. AAAI*. 2.1

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 recognition—RRC-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, 2.2

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

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

Tan, M.; and Le, Q. V. 2019. MixConv: Mixed Depthwise Convolutional Kernels . 2.2

Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, 499–515. Springer. 2.2

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

Zhang, Y.; Xiang, T.; Hospedales, T. M.; and Lu, H. 2017. Deep Mutual Learning. *CoRR* abs/1706.00384. URL http://arxiv.org/abs/1706.00384. 2.1, 2.1, 2.1, 2.2