EMU: Effective Multi-Hot Encoding Net for Lightweight Scene Text Recognition With a Large Character Set

Bingcong Li, Xin Tang, Xianbiao Qi, Yihao Chen, Chun-Guang Li, *Senior Member, IEEE*, and Rong Xiao, *Member, IEEE*

***Abstract*— Deploying a lightweight deep model for scene text recognition task on mobile devices has great commercial value. However, the conventional softmax-based one-hot classification module becomes a cumbersome obstacle when handling multi-languages or languages with large character set (*e.g.*, Chinese) due to the rapid expansion of model parameters with the number of classes. To this end, we propose an Effective Multi-hot encoding and classification modUle (EMU) for scene text recognition in the scenario of multi-languages or languages with large character set. Specifically, EMU generates a binary multi-hot label for each class with a real-valued sub-network in training stage and produces the prediction by calculating the inner product between the multi-hot code and the multi-hot label. Compared to the softmax-based one-hot classifier, EMU reduces the storage requirement and the time cost in inference stage significantly, retaining similar performance. Furthermore, we design a convolution feature based Lightweight TransFormer to learn the effective features for EMU and consequently develop a lightweight scene text recognition framework, termed Light-Former-EMU. We conduct extensive experiments on seven public English benchmarks and two real-world Chinese challenge benchmarks. Experimental results verify the effectiveness of the proposed EMU and demonstrate the promising performance of the proposed Light-Former-EMU.**

***Index Terms*— Multi-hot encoding, multi-hot classifier, transformer, lightweight transformer, scene text recognition.**

I. Introduction

SCENE text detection [1]–[3] and recognition [4]–[7] in recent years has attracted enormous attention and a number of approaches have been developed due to its great commercial value in various real-world applications, such as content analysis, identity authentication, and digital financial system.

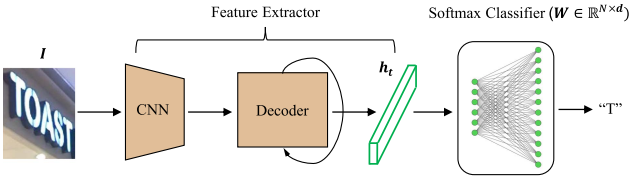


Fig. 1. Illustration for the general structure of scene text recognition pipeline. The softmax-based one-hot classifier contains a parameter matrix $\boldsymbol{W} \in \mathbb{R}^{N\times d}$.

The state-of-the-art scene text recognition approaches [8]–[11] usually consist of the following two modules, as illustrated in Fig. 1: a) a feature extraction module, which includes a convolution neural network backbone and a decoder to extract feature vector; and b) a softmax-based one-hot classifier, which is formed by a fully-connected linear layer combined with a softmax layer. The softmax-based one-hot classifier is to translate the extracted feature vector into the corresponding character.

Recently, there have been a lot of effort to make scene text recognition models more computation and memory efficient, *e.g.*, AutoSTR [12] uses neural architecture search to find an efficient backbone for scene text recognition; PP-OCR [13] utilizes a light CNN backbone, a light head module, a network pruning strategy and a PACT (PArameterized Clipping acTivation) [14] quantization technique to develop a light weight scene text recognition framework. While achieving impressive recognition accuracy, however, the models in these approaches are still too heavy and cumbersome to be deployed on mobile devices due to accommodating a softmax-based one-hot classifier, especially when dealing with multi-languages or languages with a large character set.

In the scenarios, for example, the tasks of Chinese and Japanese recognition, the character set is very large [15], [16]. For example, Chinese Internal Code Extension Specification (GBK) contains over 20,000 Chinese characters. Unfortunately, the storage requirement and FLOPs of the softmax-based one-hot classifier (on the top of a fully-connected linear layer of the features) grow linearly with the number of the classes (*i.e.*, the number of characters) due

Manuscript received 13 September 2021; revised 30 December 2021; accepted 9 January 2022. Date of publication 25 January 2022; date of current version 4 August 2022. The work of Chun-Guang Li was supported by the National Natural Science Foundation of China under Grant 61876022. This article was recommended by Associate Editor W. Lin. *(Corresponding author: Xin Tang.)*

Bingcong Li, Xin Tang, Yihao Chen, and Rong Xiao are with the Visual Computing Group, Ping An Property & Casualty Insurance Company, Shenzhen 518033, China (e-mail: bingcon.li@gmail.com; tangxint@gmail.com; yihaoo.chen@gmail.com; rongxiao@gmail.com).

Xianbiao Qi was with the Visual Computing Group, Ping An Property & Casualty Insurance Company, Shenzhen 518033, China. He is now with the International Digital Economy Academy, Shenzhen 518033, China (e-mail: qixianbiao@gmail.com).

Chun-Guang Li is with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: lichunguang@bupt.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2022.3146240.

Digital Object Identifier 10.1109/TCSVT.2022.3146240

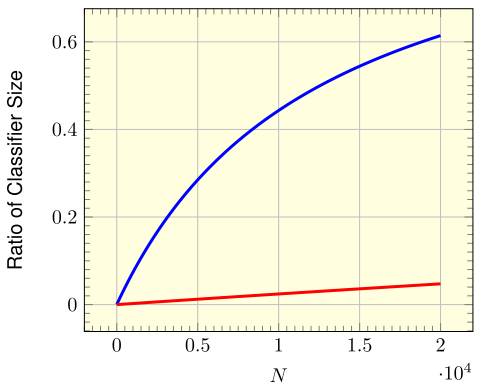


Fig. 2. Ratio of the classifier model size with the increase of the number of classes. The blue line denotes PP-OCR with a softmax-based one-hot classifier; whereas the red line denotes PP-OCR with our Multi-Hot Classifier. MobileNet-v3-small is adopted as the backbone and N is the number of characters. The ratio of the classifier model size is the proportion of the model size of the classifier over the overall model size.

to the fact that the fully-connected layer contains a projection matrix of $N \times d$, where N is the number of classes and d is the dimension of the feature vector. For instance, if $N = 20,000$ and $d = 512$, then the storage requirement is around 40 MB.

To further demonstrate the role of the softmax-based one-hot Classifier when developing a lightweight text recognition model, we take PP-OCR [13] as an example, calculate the proportion of the classifier model size over the overall model size, then show the proportion with the increase of the number of classes in Fig. 2 (*i.e.*, the blue line). As can be perceived, the ratio of the classifier model size dramatically increases when the number of classes increases. When the number of classes is $20,000$, the softmax-based one-hot classifier takes over 60% of the model size.

To overcome the great challenge posed by the large class number (*i.e.*, the large character set), in this paper, we propose an **E**ffective **M**ulti-hot encoding and classification mod**U**le (EMU), which forms a multi-hot classifier to decode the character sequence for lightweight scene text recognition. In EMU, d-dimensional feature vector is mapped into a K-dimensional multi-hot encoding by projection and binarization, and the label of the input feature vector is then determined by looking up the codebook $\boldsymbol{C} \in \mathbb{B}^{N \times K}$ in which each row vector corresponds to a multi-hot label.

To be specific, as illustrated in Fig. 3, the multi-hot classifier contains two groups of parameters: a real-valued projection matrix $\boldsymbol{P} \in \mathbb{R}^{K \times d}$ and a binary codebook $\boldsymbol{C} \in \mathbb{B}^{N \times K}$, where each of its elements is either -1 or 1. When facing a scene text recognition with a large scale character set, the parameter N will be significantly greater than d and K, *i.e.*, $N \gg \{K, d\}$. Therefore, the model size of the multi-hot classifier is significantly smaller than that of the softmax-based one-hot classifier. Taking again PP-OCR [13] as an example, as shown in Fig. 2 (*i.e.*, the red line), when replacing the one-hot classifier by the multi-hot classifier, the ratio of the classifier model size on the overall model is sharply reduced to, *e.g.*, less than 10%. In EMU, we design a two-stream network, which consists of a VisionNet to learn multi-hot codes from images and a ClassNet to learn a multi-hot code for each character.

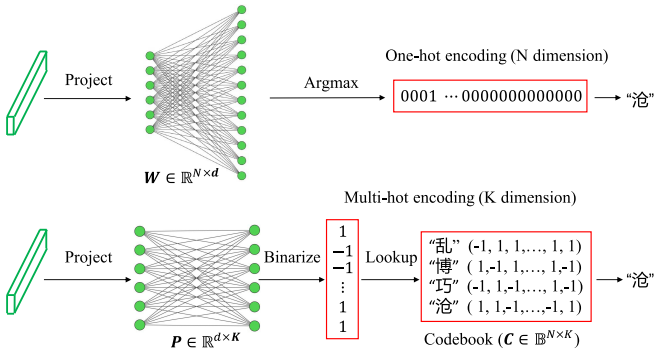


Fig. 3. Softmax-based one-hot classifier vs. multi-hot classifier. The first row shows the one-hot classifier, while the second row shows the multi-hot classifier.

Equipped with EMU, we further develop an end-to-end lightweight framework, termed **Light**weight Trans**Former** based EMU (Light-Former-EMU), which consists of a convolution feature encoder, a lightweight transformer decoder, an output embedding, and a multi-hot classifier, as shown in Fig. 5. Specifically, similar to [11], [17], [18], we adopt three transformer units for decoding, but we remove the feed-forward component in each unit and adopt cross-layer parameter sharing technique across the three units, which reduces the number of parameters in the transformer decoder. Note that the K-dimensional multi-hot codes of all the characters are shared between the output embedding and the multi-hot classifier. Thus, Light-Former-EMU is able to support scene text recognition with a large character set on mobile devices.

A. Contributions

The contributions of the paper are summarized as follows.

- We propose a novel and effective multi-hot encoding and classification module, termed EMU, which forms a multi-hot classifier to replace softmax-based one-hot classifier and learns multi-hot codes from labels and images by a two-stream network. EMU enables efficient scene text recognition with a large character set.
- We develop an end-to-end and lightweight model, named Light-Former-EMU, for scene text recognition with a large characters set (*e.g.*, Chinese and multi-language text recognition). To the best of our knowledge, Light-Former-EMU is the first transformer-based scene text recognition model that can be deployed on mobile devices and is able to handle a large number of characters.

II. Related Work

This section gives an overview on the relevant work.

A. Scene Text Recognition

The state-of-the-art frameworks for scene text recognition are built on an encoder-decoder structure and can be divided into two groups: a) Connectionist Temporal Classification (CTC) based methods [19]–[21], and b) attention mechanism based methods [7], [8], [10], [11], [22]–[26].

For CTC based methods, in [20], a combination of convolutional neural network (CNN) and recurrent neural network (RNN) is employed to model the sequence features of a given text image and at meanwhile a CTC module is adopt to train the model. Similarly, in [21], a Bidirectional Long Short-Term Memory (BiLSTM) is used for modeling sequence. However, CTC-based methods are ineffective to deal with irregular text.

Attention mechanism [22] is widely used in scene text recognition [7], [8], [10]. For example, ASTER [7] attempts to rectify the text using Thin-Plate-Spline (TPS) transformation, and then uses an sequence-to-sequence model with attention mechanism for prediction. To enhance the encoding of contextual dependencies, in [23], several BiLSTM encoders are stacked on top of CNN backbone with intermediate supervision during training. In [24], an additional global semantic information supervised by the word embedding from a pre-trained language model to initialize the decoder is introduced. To handle complex irregular texts, in [8], 1D attention module is replaced with a tailored 2D attention mechanism, and a LSTM encoder-decoder is used for decoding characters. In [25], a novel position aware module is proposed to strengthen the positional encoding capacity. In [10], the attention maps for all time steps are directly generated by utilizing a convolutional neural network to decouple the alignment operation with historical decoding results. Unlike attention mechanism, Mask TextSpotter [27] and Textscanner [26] predict the position and the order of characters based on semantic segmentation.

In addition, there are also some works that utilize self-attention mechanism for scene text recognition [11], [17], [18], [28]. For instance, SRN [11] uses transformer units to effectively capture the global spatial dependencies.

However, compared to CTC based methods, attention mechanism based models require additional memory to store the parameters of the attention-based decoder. Therefore the large model size of scene text recognition models will prohibit theirs widespread usage. It is worth to mention that GTC [9] integrates attention mechanism into CTC to design an efficient and effective model.

B. Lightening Neural Network

To deploy neural networks on mobile devices, several techniques [29]–[33] have been proposed to decrease memory and storage requirement of the model. A number of novel lightweight architectures, *e.g.*, SqueezeNet [34], MobileNet V2 [35], MobileNet V3 [36], and EfficientNet [37], with efficient operations allowing significant reduction of the model size, have been proposed. On the other hand, pruning the non-informative and redundant weights in networks [38], [39] and networks distillation [40] are widely used strategies to reduce the model size due to the high redundancy in neural networks.

In addition, network quantization is another promising and fast solution to compress the deep neural network, which is performed by reducing the precision of the network weights from 32 bit floating point type to lower bit representations, *e.g.*, 8-bit, 4-bit, 2-bit integers [30], [32], [41]. It is shown that 8-bit quantized neural networks can reduce the storage requirement significantly and retain similar performance as that in full precision counterparts [31], [42]. The most extreme quantization is Binary Neural Networks (BNN) [43]–[46], which restrict the weights and the output of the activation function to only two possible values $\{-1, 1\}$. To be specific, BNN attempts to binarize the weights and activations to -1 or $+1$ by the sign$(\cdot)$ function [43]. Since that the derivative of the sign$(\cdot)$ function is zero almost everywhere, the straight-through estimator (STE) is proposed for the gradient back-propagation. However, the training process is still ill-defined because the forward and backward functions are different. More recently, [44], [45] devise a differentiable approximation to the sign$(\cdot)$ function which is able to gradually decrease the smoothness of the approximation during training.

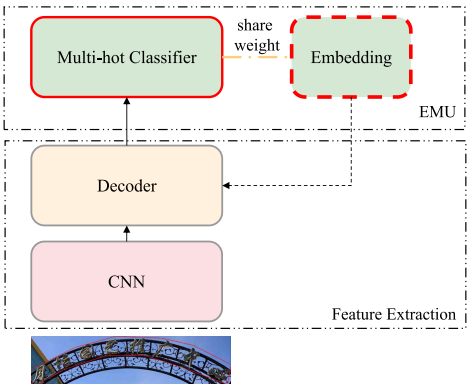


Fig. 4. Our proposed lightweight scene text recognition framework.

Although the quantization networks [31], [47], especially the binary neural networks [30], can significantly reduce the model size by quantizing the weights and activations to lower bits, the last layer, *i.e.*, the fully-connected softmax layer, is kept in higher precision or even is not quantized in most network quantization methods [30], [38], [47]–[49]. In real-world practice, when handling a large character set in scene text recognition, the storage burden of fully-connected softmax-based one-hot classifier is an obstacle to deploy the scene text recognition model on mobile devices. In this paper, we focus on developing an effective multi-hot encoding and classification module for lightweight scene text recognition with a large character set, which allow us to replace the softmax-based one-hot classifier with a novel and efficient multi-hot classifier.

III. Our Proposals

This section presents an end-to-end transformer-based lightweight model for scene text recognition with a large characters set at first, and then especially describes a novel and effective multi-hot encoding and classification module which forms a multi-hot classifier to replace softmax-based one-hot classifier.

A. Overall Framework

Similar to the common framework for scene text recognition, we propose a lightweight transformer-based framework,

TABLE I

SPECIFICATION FOR THE MODIFIED MOBILENETV3-SMALL. SE DENOTES THE SQUEEZE-AND-EXCITE. NL DENOTES THE TYPE OF NONLINEARITY IN USE. HS DENOTES H-SWISH AND RE DENOTES RELU. s DENOTES STRIDE. MAGC DENOTES THE MULTI-ASPECT GCATTENTION IN THE BLOCK [17]

Input	Operator	exp size	$\#out$	SE	NL	s	MAGC [17]
$48 \times 160 \times 3$	conv2d, 3x3	-	16	-	HS	2	-
$24 \times 80 \times 16$	bneck, 3x3	16	16	✓	RE	1	-
$24 \times 80 \times 16$	bneck, 3x3	72	24	-	RE	1	-
$24 \times 80 \times 24$	bneck, 3x3	88	24	-	RE	1	-
$24 \times 80 \times 24$	bneck, 5x5	96	40	✓	HS	2	✓
$12 \times 40 \times 40$	bneck, 5x5	240	40	✓	HS	1	✓
$12 \times 40 \times 40$	bneck, 5x5	240	40	✓	HS	1	✓
$12 \times 40 \times 40$	bneck, 5x5	120	48	✓	HS	1	✓
$12 \times 40 \times 48$	bneck, 5x5	144	48	✓	HS	1	✓
$12 \times 40 \times 48$	bneck, 5x5	144	48	✓	HS	1	✓
$12 \times 40 \times 48$	bneck, 5x5	144	48	✓	HS	1	✓
$12 \times 40 \times 48$	bneck, 5x5	144	48	✓	HS	1	✓
$12 \times 40 \times 48$	bneck, 5x5	144	48	✓	HS	1	✓
$12 \times 40 \times 96$	bneck, 5x5	288	96	✓	HS	(2, 1)	✓
$6 \times 40 \times 96$	bneck, 5x5	576	96	✓	HS	1	-
$6 \times 40 \times 96$	conv2d, 1x1	-	256	-	HS	1	-

TABLE II

SPECIFICATION FOR THE MODIFIED MOBILENETV3-LARGE

Input	Operator	exp size	$\#out$	SE	NL	s	MAGC [17]
$48 \times 160 \times 3$	conv2d, 3x3	-	16	-	HS	2	-
$24 \times 80 \times 16$	bneck, 3x3	16	16	-	RE	1	-
$24 \times 80 \times 16$	bneck, 3x3	64	24	-	RE	1	-
$24 \times 80 \times 24$	bneck, 3x3	72	24	-	RE	1	-
$24 \times 80 \times 24$	bneck, 5x5	72	40	✓	RE	(2, 1)	✓
$12 \times 80 \times 40$	bneck, 5x5	120	40	✓	RE	1	✓
$12 \times 80 \times 40$	bneck, 5x5	120	40	✓	RE	1	✓
$12 \times 80 \times 40$	bneck, 3x3	240	80	-	HS	1	✓
$12 \times 80 \times 80$	bneck, 3x3	200	80	-	HS	1	✓
$12 \times 80 \times 80$	bneck, 3x3	184	80	-	HS	1	✓
$12 \times 80 \times 80$	bneck, 3x3	184	80	-	HS	1	✓
$12 \times 80 \times 80$	bneck, 3x3	184	80	-	HS	1	✓
$12 \times 80 \times 80$	bneck, 3x3	184	80	-	HS	1	✓
$12 \times 80 \times 80$	bneck, 3x3	184	80	-	HS	1	✓
$12 \times 80 \times 80$	bneck, 3x3	184	80	-	HS	1	✓
$12 \times 80 \times 80$	bneck, 3x3	480	112	✓	HS	2	✓
$6 \times 40 \times 112$	bneck, 3x3	672	112	✓	HS	1	✓
$6 \times 40 \times 112$	bneck, 5x5	672	160	✓	HS	1	✓
$6 \times 40 \times 160$	bneck, 5x5	960	160	✓	HS	1	-
$6 \times 40 \times 160$	conv2d, 1x1	-	256	-	HS	1	-

as illustrated in Fig. 4, which consists of four modules: a) a convolution feature extractor, which is a lightweight backbone network; b) a feature sequence decoder, which is a lightweight transformer-based decoder module; c) a multi-hot classifier,[1] which is an efficient multi-hot classifier as the main contribution proposed in this paper; and d) output embedding module.

For clarity, we prepare a detailed illustration for the architecture of our proposed **Light**weight trans**Former**-based EMU framework (Light-Former-EMU) for scene text recognition in Fig. 5.

B. Lightweight Backbone for Convolution Feature Extraction

Most scene text recognition approaches usually adopt a heavy backbone, such as ResNet [50], for feature extraction in order to achieve outstanding recognition accuracy. In this paper, we choose a slightly modified lightweight MobileNet-V3 (including MobileNet-small and MobileNet-large) as the backbones in our proposed framework, which takes a balance between the recognition accuracy and the model size especially for applications on mobile devices.

Different from original MobileNet-V3, as shown in Tables I and II, we make a few adjustments as follows: a) changing the number of some bottlenecks of the network; b) changing the settings of strides of bottlenecks; and c) employing a multi-aspect GCAttention module [17] in bottleneck module.

C. Lightweight Transformer Decoder

We adopt a modified lightweight decoder of Transformer [51] to model the sequence prediction in text recognition. Specifically, we design a light-weight Transformer decoder to decode the output character.

[1]It is worth to note that the multi-hot classifier can be applied to any scene text recognition framework to significantly reduce the model size and computation complexity of the classifier and output embedding.

A standard decoder of Transformer [51] consists of a stack of N identical layers, each of which has three sub-modules: a) a masked self-attention, b) a cross-attention module, and c) a feed-forward network. In our lightweight framework, we remove the feed-forward network and share the cross-layer parameters (as in [52]) to reduce the model size of the decoder.

Let $Y^{(\ell)} \in \mathbb{R}^{T \times d}$ is denoted as the input feature sequence of the ℓ-th layer, where T is the length of the feature sequence and d is the dimension of the input feature.

For a masked self-attention module which consists of H heads, the output is defined as

$$Y^{(\ell)} := \coprod_{k=1}^{H} \mathrm{Att}(Y_Q^{(\ell)} Q_k, Y_K^{(\ell)} K_k, Y_V^{(\ell)} V_k) W^{(\ell)} \tag{1}$$

where $\mathrm{Att}(\mathbb{Q}, \mathbb{K}, \mathbb{V})$ is a scaled dot-product attention [51], which takes the query $\mathbb{Q}$, the key $\mathbb{K}$ and value $\mathbb{V}$ as the inputs, and H is the number of heads, $\coprod_{k=1}^{H}$ denotes a concatenated operator over the outputs of the H heads, $\{Q_k, K_k, V_k\}$ and $W^{(\ell)}$ are the projection matrices, and $\{Y_Q^{(\ell)}, Y_K^{(\ell)}, Y_V^{(\ell)}\}$ are masked $Y^{(\ell)}$ to prevent a given position from incorporating information about future output positions.

For a cross-attention module which consists of H heads, the output is defined as

$$Y^{(\ell)} := \coprod_{k=1}^{H} \mathrm{Att}(Y^{(\ell)} Q'_k, X K'_k, X V'_k) W'^{(\ell)} \tag{2}$$

where X is extracted by the convolution backbone, as shown in Fig. 5, $\{Q'_k, K'_k, V'_k\}$ and $W'^{(\ell)}$ are the projection matrices.

Note that the residual connections and the layer normalization are used in each sub-module but they are not expressed in Eqs. (1) and (2). By sharing cross-layer parameters, the parameters in $\{Q_k, K_k, V_k\}$, $W^{(\ell)}$, $\{Q'_k, K'_k, V'_k\}$, and $W'^{(\ell)}$ are repeatedly used to calculate the output of each decoder layer.

In the conventional encoder-decoder architecture based scene text recognition model, an output embedding is used to encode all characters. In our proposed lightweight transformer-based framework, as shown in Fig. 5, a codebook of the

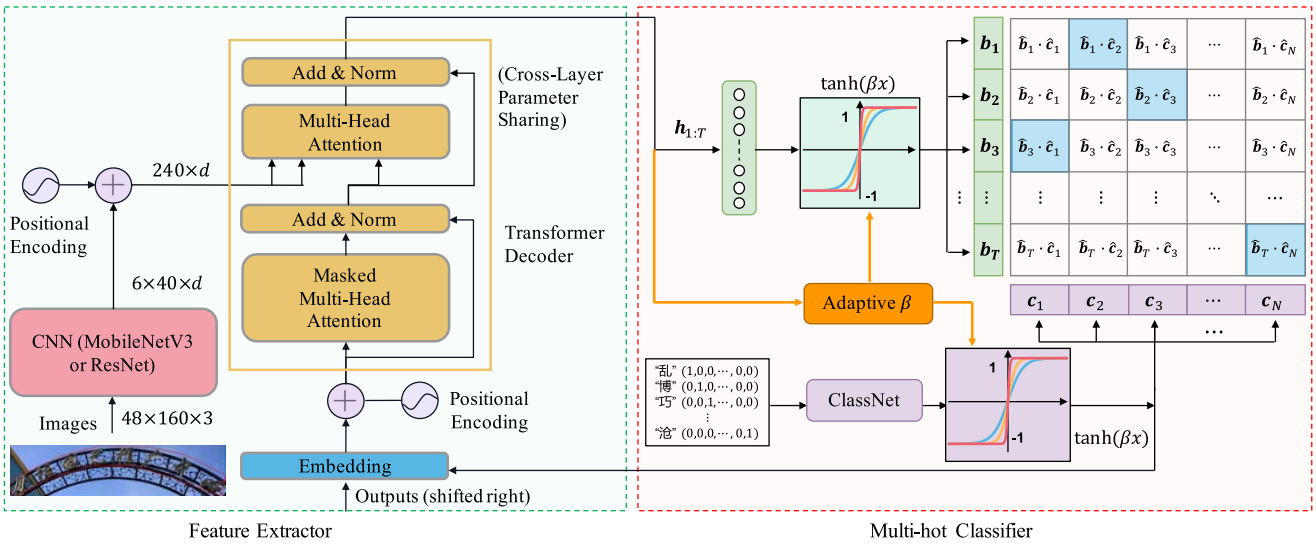


Fig. 5. Architecture of Light-Former-EMU. The left part: feature extraction, which consists of two modules: CNN backbone and transformer decoder. The right part: Multi-hot classifier and output embedding.

multi-hot classifier is used to encode all characters. We share these matrices.

D. Multi-Hot Encoding and Classification

Recall that, as shown in Fig. 3, the softmax-based one-hot classifier maps the feature vector $\boldsymbol{h} \in \mathbb{R}^d$ directly into an N-dimensional class-specific space, where a real-valued weights matrix $\boldsymbol{W} \in \mathbb{R}^{N\times d}$ is used. If N is very large, the weight matrix $\boldsymbol{W}$ consumes much memory. Since that many classes of characters in scene text recognition are similar to each other, the one-hot encoding is extremely inefficient.

In this paper, instead of mapping the feature vector $\boldsymbol{h}$ into an N-dimensional real-valued space, we propose to map it into a K-dimensional binary space. That is, each feature vector is transformed into a K-dimensional binary code $\boldsymbol{b} \in \{-1, 1\}^K$. Unlike the softmax-based one-hot classifier, which is a multi-class extension of logistic regression, we propose to use a naive Bayes classifier for classification. Assume that K elements of $\boldsymbol{b}$ are independent to each other and let $\hat{y}$ be the class label of $\boldsymbol{b}$, then the naive Bayes classifier is defined as:

$$\hat{y}(\boldsymbol{b}) = \arg\max_{y\in\{1,,...,N\}} p(y|\boldsymbol{b}) \tag{3}$$

$$\propto \arg\max_{y\in\{1,,...,N\}} p(\boldsymbol{b}|y)p(y) \tag{4}$$

$$\propto \arg\max_{y\in\{1,,...,N\}} p(y)\prod_{i=1}^{K} p(b^{(i)}|y), \tag{5}$$

where $b^{(i)}$ is the i-th element of $\boldsymbol{b}$.

Since that $\boldsymbol{b}$ is binary vector, the maximum likelihood estimation of $p(b^{(i)}|y)$ is

$$\hat{p}(b^{(i)} = 1|y = j) := \frac{\sum_{k=1}^{M} \mathbb{I}(b_k^{(i)} = 1, y_k = j)}{\sum_{k=1}^{M} \mathbb{I}(y_k = j)}, \tag{6}$$

where M is the number of training pairs $\{(\boldsymbol{b}_k, y_k)\}_{k=1}^{M}$, $\mathbb{I}(\cdot)$ is an indicator function. Similarly, we also have the maximum likelihood estimation for $p(b^{(i)} = -1|y = j)$. Note that the stage of learning binary code $\boldsymbol{b}$ and computing $\hat{p}(b^{(i)}|y = j)$ are independent and thus cannot be jointly optimized.

To jointly optimize the binary code $\boldsymbol{b}$ and the naive Bayes classifier in Eq. (3), we introduce a class-specific score function which is parameterized with $\boldsymbol{c}_j$ for each class j and we have

$$\hat{p}(\boldsymbol{b}|y = j) = \frac{\exp(s(\boldsymbol{c}_j, \boldsymbol{b}))}{\sum_{i\neq j} \exp(s(\boldsymbol{c}_i, \boldsymbol{b}))}, \tag{7}$$

where $s(\cdot,\cdot) : \{-1, 1\}^K \times \{-1, 1\}^K \rightarrow [0, 1]$ is a scoring function defined over two binary vectors. The score function measures the similarity between $\boldsymbol{c}_j$ and $\boldsymbol{b}$. For simplicity, we define the scoring function $s(\cdot,\cdot)$ as

$$s(\boldsymbol{c}_j, \boldsymbol{b}) = \frac{1}{K}\boldsymbol{c}_j^\top \boldsymbol{b}. \tag{8}$$

From Eq. (7), we need to estimate both the binary vector $\boldsymbol{b}$ for each feature vector $\boldsymbol{h}$ and the class-specific binary vectors $\{\boldsymbol{c}_j\}_{j=1}^{N}$ for classification.

1) Learn Multi-Hot Binary Codes: Given an input image $\boldsymbol{I}$, the convolution feature extraction block and lightweight transformer-based decoder module will produce a feature sequence $\{\boldsymbol{h}_t \in \mathbb{R}^d\}_{t=1}^{T}$, as illustrated in Fig. 1. To generate a sequence of binary codes, we introduce a sub-network to map each real-valued feature $\boldsymbol{h}_t$ to a binary code $\boldsymbol{b}_t$, which is defined as follows:

$$\boldsymbol{b}_t := \phi(\boldsymbol{P}^\top \boldsymbol{h}_t), \tag{9}$$

where $\phi(\cdot)$ represents an activation function and $\boldsymbol{P} \in \mathbb{R}^{d\times K}$ is a linear projection matrix. Specifically, in (9), we at first transform $\boldsymbol{h}_t$ into a K-dimensional real-valued vector and then apply an activation function $\phi(\cdot)$. Where $\phi(\cdot)$ is a sign$(\cdot)$ function, which outputs are either -1 or 1.

While using the sign$(\cdot)$ function can yield the binary code, however, it results the standard back-propagation infeasible

because the gradients of sign(·) are zero or non-exists.[2] To solve this issue, in the training stage, we use a smoothed activation function $y = \tanh(\beta x)$ where $\beta > 0$ is a scale parameter, and increase the parameter β gradually to approximate the desired sign(·) function, as illustrated in Fig. 5.

2) Learn Class-Specific Binary Parameters: To estimate a class-specific binary vector $\boldsymbol{c}_j$ for each class in Eq. (7), we propose a novel ClassNet to generate the binary vector $\boldsymbol{c}_j$ for class j as shown in Fig. 5. To be specific, in ClassNet, we use a multi-layer perceptron to learn the parameters from one-hot codes. Possible value for each element of $\boldsymbol{c}_j$ are either -1 or 1. ClassNet maps the one-hot code for each class into its corresponding multi-hot code. However, using discrete-valued vectors is difficult to optimize. Again, as in Eq. (9), we also adopt a smoothed activation function $\tanh(\beta' x)$ to approximate the function sign(·) in order to obtain binary $\boldsymbol{c}_j$ where $\beta' > 0$ is a gradually increasing parameter.

Note that we keep $\boldsymbol{c}_j$ learned in training stage with ClassNet but discard ClassNet at inference stage to save storage requirement.

IV. Implementations and Training Procedures in Practice

This section describes the implementation strategies and training procedures for the proposed end-to-end trainable framework.

A. Adaptive β-Net to Learn Multi-Hot Binary Codes

As stated in the Section III-D, to approximate the sign(·) function, β and β' should be gradually increased during the training process. When β and β' tend to infinity, the elements of $\boldsymbol{b}_t$ and $\boldsymbol{c}_j$ will tend to be either -1 or $+1$. However, the naive way to increase β as in [53] has some drawbacks, which make it failure in scene text recognition. Firstly, since the training process varies greatly for different models and datasets, it is needed to find a sequence of β based on experience, which makes it difficult to be applied. Secondly, we cannot control the binarization process for different input images when the binarization speed of different input images are different.

Therefore, we propose a novel Adaptive β-Net to calculate the scale parameter β to binarize $\boldsymbol{h}_t$, which is formulated to take $\boldsymbol{h}_t$ as input as follows:

$$\beta(\boldsymbol{h}_t) := \text{softplus}(\text{MLP}(\boldsymbol{h}_t)) + 1, \tag{10}$$

where the architecture of MLP is shown in Fig. 6, and the activation function softplus is defined as follows

$$\text{softplus}(x) = \log(1 + \exp(x)). \tag{11}$$

The adaptive β-net automatically computes the scale parameter β of the function of $\boldsymbol{h}_t$ and can be optimized in an end-to-end way with respect to the overall scene text recognition model.[3]

[2]The gradients of all nonzero inputs of sign(·) are zero; whereas the gradient at the zero is non-exists.

[3]Note that the gradients from adaptive β-net is stopped at $\boldsymbol{h}_t$ to avoid disturbing the learning of the overall scene text recognition model, and the adaptive β-net is discarded at inference stage.

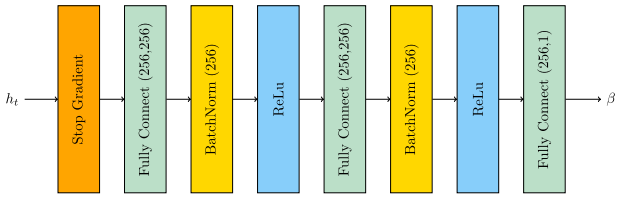


Fig. 6. Illustration for the MLP in Adaptive β-Net.

B. Training Adaptive β-Net to Learn Binarizations

Recall Eq. (8), in order to correctly predict the label of $\boldsymbol{b}_t$, we must let $\boldsymbol{c}_j^\top \boldsymbol{b}_t > \boldsymbol{c}_k^T \boldsymbol{b}_t$ where $k \neq j$. Note that $\boldsymbol{c}_j^\top \boldsymbol{b}_t = \|\boldsymbol{c}_j\|_2 \|\boldsymbol{b}_t\|_2 \cos\theta$, where θ is the angle between $\boldsymbol{c}_j$ and $\boldsymbol{b}_t$. The value of inner product depends upon the norms of $\boldsymbol{c}_j$ and $\boldsymbol{b}_t$, thus the contrastive loss function for a pair of binary codes $(\boldsymbol{b}_t, \boldsymbol{c}_j)$ is defined as

$$\ell_{\boldsymbol{b}_t, \boldsymbol{c}_j} := -\log \frac{\exp(\hat{\boldsymbol{c}}_j^\top \hat{\boldsymbol{b}}_t / \tau)}{\sum_{k=1}^{N} \exp(\hat{\boldsymbol{c}}_k^\top \hat{\boldsymbol{b}}_t / \tau)}, \tag{12}$$

where $\hat{\boldsymbol{c}}_k = \frac{\boldsymbol{c}_k}{\|\boldsymbol{c}_k\|_2}$, $\hat{\boldsymbol{b}}_t = \frac{\boldsymbol{b}_t}{\|\boldsymbol{b}_{y_t}\|_2}$, and $\tau > 0$ is a temperature parameter.

The scale parameter β should be related to the loss $\ell_{\boldsymbol{b}_t, \boldsymbol{c}_j}$. That is, in the early stage, we hope β is smaller and as $\ell_{\boldsymbol{b}_t, \boldsymbol{c}_j}$ decreases, we should increase β. Thus, we define:

$$\hat{\beta}(\boldsymbol{h}_t) = \mu \cdot \ell_{\boldsymbol{b}_t, \boldsymbol{c}_j}^{-1}, \tag{13}$$

in which μ is a hyper-parameter used as a factor to control the scale parameter. Note that $\hat{\beta}(\boldsymbol{h}_t)$ varies with the loss $\ell_{\boldsymbol{b}_t, \boldsymbol{c}_j}$ and it increases when the loss $\ell_{\boldsymbol{b}_t, \boldsymbol{c}_j}$ decreases.

In order to push the scaled $tanh(\cdot)$ function converge to $sign(\cdot)$ function and make β larger than $\hat{\beta}(\boldsymbol{h}_t)$, we form a regularized contrastive loss as follows:

$$\ell = \ell_{\boldsymbol{b}_t, \boldsymbol{c}_j} - \lambda \mathbb{I}(\beta < \hat{\beta}(\boldsymbol{h}_t))\beta, \tag{14}$$

where λ is the hyper-parameter used to balance the two losses, and $\mathbb{I}(\cdot)$ is an indicator function. As can observed from Eq. (14), that we can automatically control the scale β. That is, if β is smaller than $\hat{\beta}(\boldsymbol{h}_t)$, then the second term in Eq. (14) will force β to be larger and $\beta(\boldsymbol{h}_t)$ is thus encouraged to be larger than $\hat{\beta}(\boldsymbol{h}_t)$. As a sequence, the activation function $\phi(\cdot)$ will be sharpened with respect to the individual binarization speed of each input image.

For the scale parameter β' to binarize $\boldsymbol{c}_j$, the average of $\beta(\boldsymbol{h}_t)$ among each mini-batch of data can be used as β' to control the sharpness of activation function, since that $\boldsymbol{c}_j$ is the class-specific representation which is not dependant on the input image. Furthermore, we adopt an exponential moving average to calculate $\beta'^{(i)}$ at the i-th iteration for stabilization:

$$\beta'^{(i)} = \gamma \beta'^{(i-1)} + \frac{1}{mT}(1-\gamma) \sum_{t=1}^{T} \sum_{j=1}^{m} \beta(\boldsymbol{h}_t^{(j)}), \tag{15}$$

where $1 < \gamma < 1$ is the decay parameter, m is the batch size and $\boldsymbol{h}_t^{(j)}$ is the feature vectors.

1) Derivative Vanishing Issue: Given β is calculated, the activation function $\phi(\cdot)$ can be formulated as $\phi_\beta(x) = \tanh(\beta x)$. The derivatives of $\phi_\beta(x)$ with respect to β and x are calculated as

$$\begin{cases} \dfrac{\partial \phi(\beta, x)}{\partial \beta} = (1 - \tanh^2(\beta x))x \\ \dfrac{\partial \phi(\beta, x)}{\partial x} = (1 - \tanh^2(\beta x))\beta \end{cases} \tag{16}$$

When increasing β, both of the derivatives in Eq. (16) will be vanishing. This is an obstacle to train the adaptive β-net. To remedy this issue, we replace the gradients in Eq. (16) with the following formula:

$$\begin{cases} \dfrac{\partial \phi_\beta(x)}{\partial \beta} = 1 - \tanh^2(\beta), \\ \dfrac{\partial \phi_\beta(x)}{\partial x} = 1 - \tanh^2(x). \end{cases} \tag{17}$$

C. Computation Cost Analysis

Suppose r is the size of the type in float point (*e.g.*, 2 bytes for `float16`, and 4 bytes for `float32`). The softmax-based one-hot classifier contains rNd bytes data for the real-valued matrix $\boldsymbol{W}$ which has size $N \times d$. Meanwhile, the multi-hot classifier contains rKd bytes data for the real-valued matrix $\boldsymbol{P}$ which is of $d \times K$, and $NK/8$ bytes data for a binary codebook $\boldsymbol{C}$ which is of $N \times K$. Therefore, the compression ratio between multi-hot classifier and the softmax-based one-hot classifier is $\psi := \frac{K}{N} + \frac{K}{8rd}$. When N is large, the storage requirement of multi-hot classifier is significantly lower than softmax-based one-hot classifier. For example, if $N = 20,000$ and $d = 512$, using 256-dimensional multi-hot codes is sufficient to represent all the N classes of characters, and the storage usage of the multi-hot classifier is 1.72 MB, which is about $\frac{1}{22}$ of the softmax classifier.

V. Experiments

To validate the effectiveness of our proposed EMU and the lightweight transformer-based framework, we select seven public English benchmark datasets and two real-world Chinese challenge benchmarks, and conduct extensive experiments.

A. Datasets

This section will describe seven English benchmark datasets and two Chinese benchmark datasets.

IIIT 5K-Words [55] (**IIIT5K**) is collected from the web and contains 3,000 cropped scene text images for testing.

Street View Text [56] (**SVT**) consists of 257 training images and 647 testing images, which is collected from Google Street Image.

ICDAR 2003 [57] (**IC03**) contains 251 full scene text images. For fair comparision, we exclude images that contain non-alphanumeric characters or have less than three characters. We have 866 cropped images for evaluation.

ICDAR 2013 [58] (**IC13**) contains 1,015 cropped text images for evaluation.

ICDAR 2015 [59] (**IC15**) are captured by Google Glasses. It contains 1,811 cropped images for testing without some extremely distorted images.

SVT-Perspective [60] (**SVTP**) consists of 645 cropped images. Many images have perspective distortions.

CUTE80 [61] (**CUTE**) contains 288 text patches cropped from natural scene images for curved text recognition. Note that the text images in the first four data sets are regular while the last three data sets contain images with irregular text.

We also select two real-world Chinese recognition benchmarks, including **ReCTS** [62] and **LSVT** [63]. For ReCTS, we evaluate methods on the test set of ReCTS. The original LSVT test set is an end-to-end OCR task, and does not provide the location of text. Thus, we use the whole training set of LSVT dataset to evaluate the performance of methods. The Chinese character set contains 6,372 characters. We use the Normalized Edit Distance (NED) as our metric which follows the LSVT 2019 competition [63].

B. Training Strategy and Implementation Details

We replace the multi-hot classifier of Light-Former-EMU in Fig. 5 with a softmax-based one-hot classifier and denote it as Light-Former. The backbone network in Transformer can be implemented with MobileNet-V3 or ResNet31, which are denoted as Light-Former (MobileNet-V3) and Light-Former (ResNet31), respectively. We implement Light-Former and Light-Former-EMU with PyTorch and run all experiments on a NVIDIA Tesla V100 GPU with 16GB memory.

For English scene text benchmarks, all the models are trained only on two synthetic datasets, **SynthText** [54] (ST) and **Synth90K** [64] (90k), without fine-tuning on any real datasets. To train the Chinese recognition model, we use several public Chinese scene text datasets as training data, including MLT-2017 [15], MLT-2019 [16], ReCTs [62], ArT [65], CTW [66], RCTW [67] and Baidu Chinese scene text recognition competition.[4] Besides, we also generate 2.3 million synthetic images for training. All the compared models are trained on the same training data.

In training phase, the batch size is 160. All input images are padded and resized to 48×160. The number of dimension of multi-hot code is set to 512 in all experiments except explicitly specified. The EMU module introduces several hyper-parameters: μ, λ, γ and τ. In experiments, we find that the decay factor γ is not sensitive and thus set it to 0.999, and μ and λ are also insensitive and set to 1, 10^{-4}. In Eq. (12), the loss function contains a temperature parameter τ which is set to 20.

C. English Recognition Evaluation

We compare our proposed Light-Former-EMU to previous state-of-the-art methods on several benchmarks. The results are shown in Table III. Note that we use MobileNet-V3 as the backbone in Light-Former-EMU. To have a comprehensive evaluation, we also report the results of Light-Former-EMU (ResNet31), in which ResNet31 is used as the convolution backbone network. We can read that Light-Former-EMU (ResNet31) achieves comparable results across seven benchmarks. Especially, it obtains 95.1%, 95.1%, 89.2%

[4] https://aistudio.baidu.com/aistudio/datasetdetail/10879

TABLE III

PERFORMANCE COMPARISON ON REGULAR AND IRREGULAR TEXT DATASETS. "SA" ARE THE TRAINING DATA OF SYNTHADD [54]. NOTE THAT THE MODEL SIZE OF PP-OCR IS 5.3 MB AND LIGHT-FORMER-EMU (MOBILENET V3) IS 4.4 MB. THE LIGHT-FORMER-EMU-32 AND LIGHT-FORMER-EMU-64 DENOTE USING 32 AND 64 DIMENSIONAL MULTI-HOT CODE IN EMU FOR EACH CHARACTER, RESPECTIVELY

Methods	Training Data	Regular Text				Irregular Text		
		IIIT5K	SVT	IC03	IC13	IC15	SVTP	CUTE
CRNN [20] (2016)	90k	81.2	82.7	91.9	89.6	–	66.8	54.9
ASTER [7] (2018)	90k+ST	93.4	89.5	94.5	91.8	76.1	78.5	79.5
SAR [8] (2019)	90k+ST+SA	91.5	84.5	-	91.0	69.2	76.4	83.3
Master [17] (2019)	90k+ST+SA	**95.0**	90.6	**96.4**	95.3	79.4	84.5	87.5
SRN [11] (2020)	90k+ST	94.8	**91.5**	-	**95.5**	**82.7**	**85.1**	**87.8**
DAN [10] (2020)	-	94.3	89.2	95.0	93.9	74.5	80.0	84.4
AutoSTR [12] (2020)	90k+ST	94.7	90.9	93.3	94.2	81.8	81.7	-
PP-OCR (2020) (MobileNet V3)	90k+ST	83.7	84.1	92.8	91.7	69.5	71.0	62.1
Light-Former (ResNet31)	90k+ST	95.1	**89.8**	94.5	**95.7**	**79.1**	82.2	**91.0**
Light-Former-EMU (ResNet31)	90k+ST	95.1	89.5	94.4	95.1	78.7	82.2	89.2
Light-Former (MobileNet V3)	90k+ST	91.8	**88.3**	93.6	93.7	74.5	77.1	75.4
Light-Former-EMU-32 (MobileNet V3)	90k+ST	91.5	87.1	94.0	93.5	74.3	75.1	75.7
Light-Former-EMU-64 (MobileNet V3)	90k+ST	92.2	87.8	93.8	94.1	75.0	77.5	**77.4**
Light-Former-EMU (MobileNet V3)	90k+ST	**92.6**	88.1	**94.4**	**94.6**	**75.5**	**78.0**	76.0

on IIIT5K, IC13, CUTE, respectively. Light-Former-EMU (MobileNet-V3) has competitive model size and outstanding performance compared with the PP-OCR model and CRNN. Light-Former-EMU is smaller than PP-OCR, but Light-Former-EMU gains around 5% improvement on irregular text datasets. We can read that multi-hot classifier achieves competitive results across over seven English scene text benchmarks. Since the dimension of the multi-hot code is 512, which is larger than the number of characters in these English benchmarks. The multi-hot Classifier performs slightly better than the softmax-based one-hot classifier.

To validate the effectiveness of the dimensionality of multi-hot code, we also use 32 and 64 dimensional multi-hot code to represent each character. From Table III, we can observe that the performance of Light-Former-EMU, which uses 512 dimensional multi-hot code, is better than Light-Former-EMU-32 and Light-Former-EMU-64. When the dimension of the multi-hot code is higher, Light-Former-EMU achieves better performance.

The multi-hot code of each character is automatically learned by our EMU model from the datasets. We compute the similarity between two multi-hot codes of characters and show the similarity matrix between the multi-hot codes in Fig. 7. From the similarity matrix, we find that some visually similar characters have higher similarity scores between them. For example, the similarity score between Character 'G' and 'C' is higher than Character 'G' with others. Similarly, so for the Characters 'W' and 'V', 'N' and 'M'.

Light-Former-EMU mainly adopts three techniques to reduce the model size, including removing feed-forward network, sharing cross-layer parameter and using multi-hot classifier. To analyse the effect of these techniques, we perform ablation studies on English scene text evaluation datasets. The results are shown in Table IV. As can be seen that, these techniques have little effect on recognition accuracy when ResNet31 is used as the backbone. When using MobileNet-V3-small as the backbone, removing the feed-forward network and the cross-layer parameter sharing lead to an approximately 2% drop in accuracy. Nevertheless,

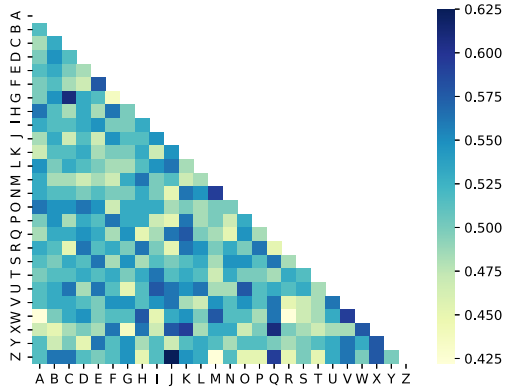


Fig. 7. The similarity matrix between characters. The similarity is computed by $\frac{1}{K}\boldsymbol{b}_i^\top \boldsymbol{b}_j$, where $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$ are the binary codes of the characters i and j.

TABLE IV

EFFECT OF REMOVING FEED-FORWARD NETWORK AND CROSS-LAYER PARAMETER SHARING (NOFFN+PS) AND MULTI-HOT CLASSIFIER ON MODEL SIZE AND ACCURACY. AVERAGE ACCURACY ON SEVEN ENGLISH BENCHMARK DATASETS IS SHOWN IN THE "AVG" COLUMN

Backbone	NoFFN+PS	Multi-hot Classifier	Model Size (MB)	AVG
ResNet 31	×	×	223.9	89.5
	✓	×	184.1	89.6
	×	✓	224.6	89.6
	✓	✓	184.8	89.2
MobileNet V3 small	×	×	19.7	87.1
	✓	×	3.7	84.9
	×	✓	20.4	87.5
	✓	✓	4.4	85.6

as an alternative, the model size is significantly reduced from 19.7MB to 3.7MB. Thus, we can adopt these two techniques into Light-Former-EMU to have a balance between the recognition accuracy and the model size.

D. Chinese Recognition Evaluation

We conduct experiments on two real-world Chinese text datasets. Since there are 6, 372 characters and more complex

TABLE V

PERFORMANCE ON CHINESE BENCHMARKS. SANHL_V1 IS THE CHAMPION OF RECTS CHALLENGE. PP-OCR IS TRAINED ON THE SAME DATASETS USING THE OFFICIAL CODE TO PRODUCE THE RESULTS. MH MEANS EMPLOYING THE MULTI-HOT CLASSIFIER

Methods	Backbone	Model Size (MB)	1-NED	
			ReCTS	LSVT
SANHL_v1	–	–	95.6	–
Light-Former	ResNet 31	208.7	94.3	87.6
Light-Former-EMU		186.2	94.5	87.5
ASTER	ResNet*	105.0	84.0	69.5
ASTER-MH		82.5	84.3	69.1
PP-OCR	MobileNet v3 large x1	15.4	87.2	77.8
Light-Former		19.2	92.0	81.5
Light-Former-EMU		8.0	91.8	81.1
PP-OCR	MobileNet v3 small x1	7.0	81.1	70.4
SAR		33.8	83.8	68.6
Light-Former		16.1	89.8	78.4
SAR-MH		10.8	83.5	68.5
Light-Former-EMU w/o β-Net		4.8	88.7	77.3
Light-Former-EMU		4.8	89.4	78.1

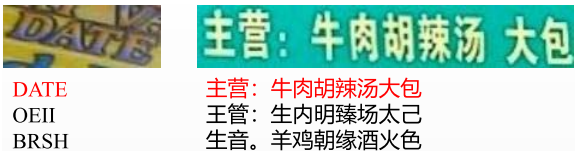


Fig. 8. Samples of the top-3 recognition results of Light-Former-EMU.

visual variations of characters, these two datasets are more challenging.

In Table V, we show the performance of several methods with different backbones and their model sizes on ReCTS and LSVT. SANHL_v1 is the champion method on ReCTS challenge.[5] It uses an ensemble model, which consists of attention-based networks, transformer networks and CTC-based networks. Our Light-Former-EMU (ResNet31) is a single model and achieves 94.5% on ReCTS and 87.5% on LSVT.

When using ResNet as backbone, Light-Former-EMU (ResNet) achieves similar performance with the transformer-based baseline model (Light-Former), which uses the softmax-based one-hot classifier. Light-Former-EMU gives an accuracy increase of +0.2% (94.5% vs 94.3%) on ReCTS and decrease of -0.1 % (87.5% vs 87.6%) on LSVT. To verify the effectiveness of the multi-hot classifier, we implement ASTER, SAR with the multi-hot classifier (ASTER-MH, SAR-MH) respectively. The results clearly demonstrate that the multi-hot classifier largely reduces the model size, without harming the performance. In Fig. 8, we show some top-3 recognition results of Light-Former-EMU.

When using a lightweight backbone, compared to Light-Former, the multi-hot classifier sharply reduces the model size of Light-Former-EMU. For example, we use MobileNet-v3-small $\times 1$ as the backbone, the model size of Light-Former is 16.1MB while Light-Former-EMU is only 4.8MB. The multi-hot classifier saves 70% storage requirement of the baseline model (Light-Former). With the same backbone, we also observe that the size of Light-Former-EMU is also significantly smaller than PP-OCR.

[5]https://rrc.cvc.uab.es/?ch=12&com=evaluation&task=2

Due to the transformer-based decoder, Light-Former-EMU achieves outstanding performance. It is clear that Light-Former-EMU outperforms its counterpart PP-OCR with much smaller model size. When using MobileNet-v3-large, Light-Former-EMU outperforms PP-OCR by 4.6%, and with MobileNet-v3-small, Light-Former-EMU outperforms PP-OCR by 8.3%. We also find that the performance of Light-Former-EMU is slightly lower than the baseline model when MobileNet-v3 is used. Specifically, when using MobileNet-v3-small $\times 1$, the recognition accuracy of Light-Former-EMU slightly decrease on ReCST (-0.4%) and LSVT (-0.3%). Note that from Eq. (9), we observe that the binary code depends on the backbone. The possible reason is that we may obtain better binary codes for characters when using a heavier backbone model.

E. Ablation Study and Analysis

1) Effectiveness of Adaptive β-Net: To validate the effectiveness of the proposed adaptive β-Net, we conduct experiments with Light-Former-EMU without using the adaptive β-Net, marked as "Light-Former-EMU w/o β-Net". In experiments, β is calculated with $(1 + 0.5i)^{0.5}$, in which i is the number of training iterations. In Table V, it shows that removing the adaptive β-Net causes a performance drop of 0.7% on ReCTS and 0.8% on LSVT. To obtain the binary code, we use a smoothed activation function $y = \tanh(\beta x)$ and increase β gradually to approximate the function sign$(\cdot)$. The adaptive β-Net can not only improve the performance of Light-Former-EMU, but also automatically control the binarization progress by updating the value of β. In Fig. 9, we show the histograms of the values of β calculated by the adaptive β-Net as in Eq. (10) at 10K, 100K and 200K iterations. We can observe that, at 10K iterations, most values of β are smaller than 50. At this stage, we need to learn the weights of the model. Hence, using a small β makes the activation function smoother and prevent the gradients to vanish. As the number of training iterations increases, the scale parameter β become larger and the activation function $y = \tanh(\beta x)$ become more non-smooth. For example, at 100K iterations, many values of β are larger than 100. At 200K iterations, most values of β are larger than 200. In this situation, the activation function $y = \tanh(\beta x)$ tends to converge to the function sign$(\cdot)$ and therefore we will obtain the desired binarization.

2) Effectiveness of Multi-Hot Classifier: Multi-hot classifier is used to reduce the model size of the softmax-based one-hot classifier. The softmax-based one-hot classifier contains a weight matrix $\mathbf{W} \in \mathbb{R}^{N \times d}$. When the value of N is very large, like [52], we can use the matrix decomposition technique to save the storage requirement of $\mathbf{W}$. The weight matrix $\mathbf{W}$ is decomposed into the product of two matrices $\mathbf{W}_1 \in \mathbb{R}^{N \times M}$, $\mathbf{W}_2 \in \mathbb{R}^{M \times d}$. By using this decomposition, we reduce the parameters of softmax-based one-hot classifier from $\mathcal{O}(Nd)$ to $\mathcal{O}(NM) + \mathcal{O}(dM)$. The parameter reduction is significant when $d \gg M$. In experiments, we use MobileNet-v3-small$\times 1$ as the backbone and implement the baseline model (Light-Former) with the matrix decomposition technique. In our Chinese recognition scenario, we have $d = 256, N = 6,372$. For the baseline model (Light-Former), the model size of the

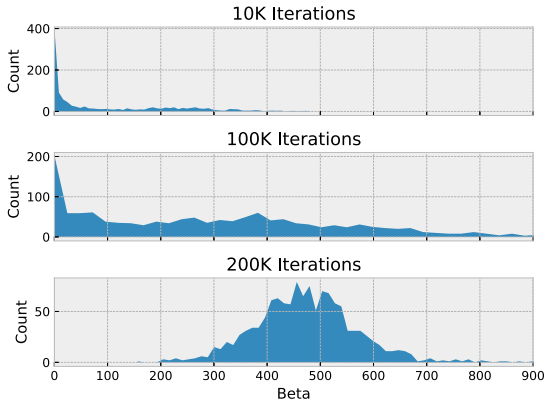


Fig. 9. Histograms of values of β at 10K, 100K and 200K iterations.

TABLE VI

COMPARISON BETWEEN THE MULTI-HOT CLASSIFIER AND MATRIX DECOMPOSITION TECHNIQUE

Method	M	Classifier Size(MB)	ReCTs	LSVT
Matrix decomposition	4	0.10	57.1	46.2
	8	0.20	78.6	66.4
	16	0.40	84.2	73.1
	32	0.81	85.7	75.6
	64	1.62	87.9	77.4
	128	3.24	88.3	77.9
Method	K	Classifier Size(MB)	ReCTs	LSVT
Multi-Hot Classifier	128	0.22	80.2	73.5
	256	0.45	87.8	75.4
	512	0.89	89.4	78.1
	1024	1.78	89.9	78.2

TABLE VII

PERFORMANCE OF PP-OCR WITH THE MULTI-HOT CLASSIFIER. BOTH OF METHODS ARE IMPLEMENTED WITH A LIGHT-WEIGHT MOBILENET BACKBONE, THE MODEL SIZE OF WHICH IS 4.7MB

Method	Multi-Hot Classifier	Classifier Size (MB)	ReCTs	LSVT
PP-OCR		2.3	81.1	70.4
PP-OCR	✓	0.1	80.4	69.5

original softmax-base one-hot classifier is 6.2MB. When $\mathbf{W}$ is decomposed into $\mathbf{W}_1$ and $\mathbf{W}_2$ with different values of M, the corresponding model size of the softmax-based classifier is depicted as in Table VI. Obviously, with the smaller M, the model size is lower. However, the performance of the model significantly decreases when the M is too small. In Table VI, we list the performance of the multi-hot classifier with different values of K. Compared with the matrix decomposition technique, we can read that using a multi-hot classifier can achieve better performance with the smaller model size.

We also implement PP-OCR with our multi-hot classifier. As shown in Table VII, with the help of the multi-hot classifier, the model size of the classifier module decreases from 2.3MB to 0.1MB.

3) Inference Speed: Theoretically, the inference speed of our multi-hot classifier is 26.4 times faster than the original softmax-based one-hot classifier. The matrix multiplication in the multi-hot classifier can be done via a bit-wise operator,

TABLE VIII

COMPUTATION COMPLEXITY (FLOPS) OF CNN+CTC MODEL. "MODEL FLOPS" MEANS THE TOTAL FLOPS IN THE BACKBONE AND THE CLASSIFIER

	One-Hot	Multi-Hot
Complexity of Classfier	$2TDN$	$2T(DK + KN/58)$
Classifier FLOPs (M)	68.5	2.2
Model FLOPs (M)	126.9	60.6

TABLE IX

INFERENCE TIME OF CNN+CTC MODEL. THE INFERENCE TIME IS TESTED ON SNAPDRAGON 625 AND SNAPDRAGON 855, RESPECTIVELY

	CPU	One-Hot	Multi-Hot
Classifier Time (ms)	625	16.2	7.4
Model Time (ms)		36.1	27.3
Classifier Time (ms)	855	8.1	4.5
Model Time (ms)		15.3	11.7

which is about 58 times faster than the floating-point (Refer to XNOR-Net [30]). We test the speed of our proposed approach on Snapdragon 625 and 855 CPUs, respectively. In Table IX, we show that our multi-hot classifier is approximately 2 times faster than the softmax-based one-hot classifier with PP-OCR. Our multi-hot classifier can accelerate PP-OCR by 23%.

VI. CONCLUSION

We proposed an **E**ffective **M**ulti-hot encoding and classification mod**U**le (EMU) to replace the softmax-based one-hot classifier for lightweight scene text recognition. Specifically, in EMU, we designed a novel adaptive β-net to learn the binarization codes from feature sequences (VisionNet) and the class-specific binary parameters which is used to transform the one-hot code of each character into multi-hot code (ClassNet). To effectively learn the binary multi-hot codes, we defined a pairwise contrastive loss with a specific regularization to encourage binarization.

Furthermore, we developed an end-to-end trainable lightweight transformer-based multi-hot classification framework for lightweight scene text recognition on mobile devices. We conducted extensive experiments on benchmark datasets and demonstrated that: a) Compared to the softmax-based one-hot classifier, our proposed multi-hot classifier can significantly reduce the storage requirement without loss of recognition accuracy when dealing with a large scale character set; b) Our proposed lightweight transformer-based multi-hot classification framework can produce promising performance. We hope that our proposed lightweight framework can be widely used in deploying scene text recognition application on mobile devices.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments for improving the quality of the work.

REFERENCES

[1] K. S. Raghunandan, P. Shivakumara, S. Roy, G. H. Kumar, U. Pal, and T. Lu, "Multi-script-oriented text detection and recognition in video/scene/born digital images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 4, pp. 1145–1162, Apr. 2019.

[2] P. Cheng, Y. Cai, and W. Wang, "A direct regression scene text detector with position-sensitive segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 4171–4181, Nov. 2020.

[3] T. Q. Phan, P. Shivakumara, S. Bhowmick, S. Li, C. L. Tan, and U. Pal, "Semiautomatic ground truth generation for text detection and recognition in video images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 8, pp. 1277–1287, Aug. 2014.

[4] C.-Z. Shi, C.-H. Wang, B.-H. Xiao, S. Gao, and J.-H. Hu, "Scene text recognition using structure-guided character detection and linguistic knowledge," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 7, pp. 1235–1250, Jul. 2014.

[5] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4168–4176.

[6] X. Yang, D. He, Z. Zhou, D. Kifer, and C. L. Giles, "Learning to read irregular text with attention mechanisms," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, p. 3.

[7] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "ASTER: An attentional scene text recognizer with flexible rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2035–2048, Sep. 2019.

[8] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8610–8617.

[9] W. Hu, X. Cai, J. Hou, S. Yi, and Z. Lin, "GTC: Guided training of CTC towards efficient and accurate scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11005–11012.

[10] T. Wang *et al.*, "Decoupled attention network for text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2020, p. 12 216–12 224.

[11] D. Yu *et al.*, "Towards accurate scene text recognition with semantic reasoning networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, p. 12113.

[12] H. Zhang, Q. Yao, M. Yang, Y. Xu, and X. Bai, "AutoSTR: Efficient backbone search for scene text recognition," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 751–767.

[13] Y. Du *et al.*, "PP-OCR: A practical ultra lightweight OCR system," 2020, *arXiv:2009.09941*.

[14] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, "PACT: Parameterized clipping activation for quantized neural networks," 2018, *arXiv:1805.06085*.

[15] N. Nayef *et al.*, "ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification–RRC-MLT," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 1454–1459.

[16] N. Nayef *et al.*, "ICDAR 2019 robust reading challenge on multi-lingual scene text detection and recognition—RRC-MLT-2019," in *Proc. Int. Conf. Document Anal. Recognit.*, Nov. 2019, pp. 1582–1587.

[17] N. Lu *et al.*, "MASTER: Multi-aspect non-local network for scene text recognition," *Pattern Recognit.*, vol. 117, Sep. 2021, Art. no. 107980.

[18] L. Yang, P. Wang, H. Li, Z. Li, and Y. Zhang, "A holistic representation guided attention network for scene text recognition," *Neurocomputing*, vol. 414, pp. 67–75, Nov. 2020.

[19] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 369–376.

[20] 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, Nov. 2017.

[21] J. Wang and X. Hu, "Gated recurrent convolution neural network for OCR," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 335–344.

[22] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[23] R. Litman, O. Anschel, S. Tsiper, R. Litman, S. Mazor, and R. Manmatha, "SCATTER: Selective context attentional scene text recognizer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, p. 11962.

[24] Z. Qiao, Y. Zhou, D. Yang, Y. Zhou, and W. Wang, "SEED: Semantics enhanced encoder-decoder framework for scene text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, Art. no. 13528.

[25] X. Yue, Z. Kuang, C. Lin, H. Sun, and W. Zhang, "RobustScanner: Dynamically enhancing positional clues for robust text recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 135–151.

[26] Z. Wan, M. He, H. Chen, X. Bai, and C. Yao, "TextScanner: Reading characters in order for robust scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, 2020, pp. 12120–12127.

[27] 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 *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 67–83.

[28] J. Lee, S. Park, J. Baek, S. J. Oh, S. Kim, and H. Lee, "On recognizing texts of arbitrary shapes with 2D self-attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2020, pp. 546–547.

[29] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–21.

[30] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 525–542.

[31] B. Jacob *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2704–2713.

[32] Y. Choukroun, E. Kravchik, F. Yang, and P. Kisilev, "Low-bit quantization of neural networks for efficient inference," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3009–3018.

[33] O. Zafrir, G. Boudoukh, P. Izsak, and M. Wasserblat, "Q8BERT: Quantized 8Bit BERT," in *Proc. 5th Workshop Energy Efficient Mach. Learn. Cognit. Comput. NeurIPS, Ed., (EMC2-NIPS)*, Dec. 2019, pp. 1–5.

[34] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, *arXiv:1602.07360*.

[35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[36] A. Howard *et al.*, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[37] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[38] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–14.

[39] S. Han, "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. systems.*, 2015, pp. 1135–1143.

[40] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. Adv. Neural Inf. Process. Syst. Workshop*, 2014, pp. 1–9.

[41] A. Bhandare *et al.*, "Efficient 8-bit quantization of transformer neural machine language translation model," in *Proc. Int. Conf. Mach. Learn. Workshop*, 2019.

[42] P. Wang, Q. Chen, X. He, and J. Cheng, "Towards accurate posttraining network quantization via bit-split and stitching," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 9847–9856.

[43] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1," 2016, *arXiv:1602.02830*.

[44] F. Lahoud, R. Achanta, P. Márquez-Neila, and S. Süsstrunk, "Self-binarizing networks," 2019, *arXiv:1902.00730*.

[45] H. Qin *et al.*, "Forward and backward information retention for accurate binary neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2250–2259.

[46] H. Kirchhoffer *et al.*, "Overview of the neural network compression and representation (NNR) standard," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Jul. 9, 2021, doi: 10.1109/TCSVT.2021.3095970.

[47] J. Yang *et al.*, "Quantization networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 7308–7316.

[48] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, and N. Sebe, "Binary neural networks: A survey," *Pattern Recognit.*, vol. 105, Sep. 2020, Art. no. 107281.

[49] A. Bulat, B. Martinez, and G. Tzimiropoulos, "High-capacity expert binary networks," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–19.

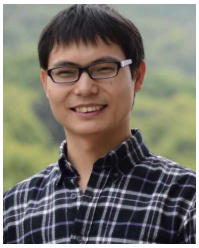
[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[51] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[52] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–17.

[53] Z. Cao, M. Long, J. Wang, and P. S. Yu, "HashNet: Deep learning to hash by continuation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5608–5617.
[54] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2016, pp. 2315–2324.
[55] A. Mishra, K. Alahari, and C. V. Jawahar, "Top-down and bottom-up cues for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2687–2694.
[56] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1457–1464.
[57] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *Proc. 7th Int. Conf. Document Anal. Recognit.*, 2003, pp. 682–687.
[58] D. Karatzas *et al.*, "ICDAR 2013 robust reading competition," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1484–1493.
[59] D. Karatzas *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 1156–1160.
[60] T. Q. Phan, P. Shivakumara, S. Tian, and C. L. Tan, "Recognizing text with perspective distortion in natural scenes," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 569–576.
[61] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Syst. Appl.*, vol. 41, no. 18, pp. 8027–8048, 2014.
[62] R. Zhang *et al.*, "ICDAR 2019 robust reading challenge on reading Chinese text on signboard," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1577–1581.
[63] Y. Sun *et al.*, "ICDAR 2019 competition on large-scale street view text with partial labeling–RRC-LSVT," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1557–1562.
[64] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," in *Proc. Adv. Neural Inf. Process. Syst. Workshop*, 2014, pp. 1–10.
[65] C. K. Chng *et al.*, "ICDAR2019 robust reading challenge on arbitrary-shaped text–RRC-ArT," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, Art. no. 15711576.
[66] T.-L. Yuan, Z. Zhu, K. Xu, C.-J. Li, and S.-M. Hu, "Chinese text in the wild," 2018, *arXiv:1803.00085*.
[67] M. He *et al.*, "ICPR2018 contest on robust reading for multi-type web images," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 7–12.

Bingcong Li received the M.S. degree in control science and engineering from the Guangdong University of Technology. He is currently a Computer Vision Engineer with Ant Financial Services Group, China. His research interests include optical character recognition, few-shot learning, and semi-supervised learning.



Xin Tang received the B.Sc. degree in information and computation science from Huazhong Agricultural University in 2007 and the Ph.D. degree in information and computational science from Sun Yat-sen University in 2014. He was a Lecturer with Huazhong Agricultural University from 2014 to 2017. He was also a Post-Doctoral Researcher with Nanyang Technological University, Singapore, from 2017 to 2019. He is currently an Engineer with Ping An Property & Casualty Insurance Company, Shenzhen, China. His research interests include optical character recognition, object detection, object segmentation, and vision language model.

Xianbiao Qi received the B.E. and Ph.D. degrees from the Beijing University of Posts and Telecommunications (BUPT) in 2008 and 2015, respectively. He visited the Web Search and Mining Group, Microsoft Research Asia (MSRA), from January 2011 to May 2012. From January 2014 to November 2015, he was a Researcher with the Center of Machine Vision Group, University of Oulu, Finland. He was a Post-Doctoral Researcher at Hong Kong Polytechnic University (PolyU) from May 2016 to May 2018. Before, he was a Senior Image Expert at Ping An Property & Casualty Insurance Company. Currently, he is a Senior Research Scientist with International Digital Economy Academy (IDEA), Shenzhen, China. His research interests include vision transformer, large-scale image and multimodal pretrained model, neural network optimization, text detection and recognition, and computer vision.

Yihao Chen received the B.S. degree from the Zhongkai University of Agriculture and Engineering in 2015. Then, he joined a startup company as an Engineer. Before, he was a Research Software Development Engineer (RSDE) at Ping An Property & Casualty Insurance Company. Currently, he is a Senior Engineer with International Digital Economy Academy (IDEA), Shenzhen, China. His research interests include vision transformer, large-scale pretrained model, face recognition, and text detection and recognition in natural images.

Chun-Guang Li (Senior Member, IEEE) received the B.E. degree in telecommunication engineering from Jilin University in 2002 and the Ph.D. degree in signal processing from the Beijing University of Posts and Telecommunications (BUPT) in 2007. From July 2011 to April 2012, he visited the Visual Computing Group, Microsoft Research Asia. From December 2012 to November 2013, he visited the Vision, Dynamics, and Learning Laboratory, Johns Hopkins University. From December 2019 to February 2020, he visited Johns Hopkins Mathematical Institute for Data Science (MINDS). He is currently an Associate Professor with the School of Artificial Intelligence, BUPT. He has published over 50 referred research papers. His research interests include data science, machine learning, and pattern recognition. He is a member of ACM and CCF. He served as the Area Chair for ICPR2020 and CVPR2021.

Rong Xiao (Member, IEEE) received the Ph.D. degree from Nanjing University, Nanjing, China, in 2001. He joined Microsoft Research Asia (MSRA) as an Associate Researcher in July 2001. He enjoyed 18 years of research life in MSRA and Microsoft Research Redmond. From April 2018 and January 2021, he was an AI Director at Ping An Property & Casualty Insurance Company. He has published approximately 30 papers at leading journals and conferences, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, ICCV, CVPR, ECCV, and ACM Multimedia. His research interests include statistical machine learning, face detection and recognition, object detection, and tracking.