Handwritten Text Recognition with Convolutional Prototype Network and Most Aligned Frame Based CTC Training

Likun Gao[1,2], Heng Zhang[1], and Cheng-Lin Liu[1,2,3]

[1] National Laboratory of Pattern Recognition (NLPR),
Institution of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2] School of Artificial Intelligence,
University of Chinese Academy of Sciences, Beijing 100049, China
[3] CAS Center for Excellence in Brain Science and Intelligence Technology,
Beijing 100190, China
gaolikun2018@ia.ac.cn, heng.zhang@ia.ac.cn, liucl@nlpr.ia.ac.cn

Abstract. End-to-end Frameworks with Connectionist Temporal Classification (CTC) have achieved great success in text recognition. Despite high accuracies with deep learning, CTC-based text recognition methods also suffer from poor alignment (character boundary positioning) in many applications. To address this issue, we propose an end-to-end text recognition method based on robust prototype learning. In the new CTC framework, we formulate the *blank* as the rejection of character classes and use the one-vs-all prototype classifier as the output layer of the convolutional neural network. For network learning, based on forced alignment between frames and character labels, the most aligned frame is up-weighted in CTC training strategy to reduce estimation errors in decoding. Experiments of handwritten text recognition on four benchmark datasets of different languages show that the proposed method consistently improves the accuracy and alignment of CTC-based text recognition baseline.

Keywords: Text Recognition · Connectionist Temporal Classification · Convolutional Prototype Network · Frame Alignment · Most Aligned Frame.

1 Introduction

Text (character string) recognition, as an important sequence labeling problem, has been widely studied by researchers in industry and academia. Text recognition has potential applications in many scenarios, such as street number reading, bank checks, mail sorting, and historical documents. Due to the complexity of image layout, the diversity of handwriting styles, and the variety of image backgrounds, text recognition is remaining a challenging task. Taking advantage of deep learning approaches, text recognition has been largely advanced in recent years. Especially, the Connectionist Temporal Classification(CTC) [8] and

attention-based end-to-end frameworks, representing the state-of-the-art, have achieved superior results in many text recognition works.

Early CTC-based methods [27] used hand-crafted image features such as histogram of oriented gradient (HOG), and recurrent neural network (RNN) for context modeling. Replacing HOG with Convolutional Neural Network (CNN), Shi et al. [24] proposed an end-to-end model in scene text recognition named Convolutional Recurrent Neural Network (CRNN). Yin et al. [40] proposed a new framework using CNN for sliding-window-based classification to enable parallel training and decoding. The sliding-window-based method not only achieves better performance but also largely reduces model parameters and computation cost. As for attention-based methods, since firstly applied to scene text recognition by Shi et al. [25], this framework has been followed by many researchers. Combining with the attention mechanism, RNN integrates global information at each step and directly outputs the decoding results. In addition to the flexibility of decoding from 1D to 2D alignment, the attention framework can also memorize semantic information to improve the recognition accuracy in scene text recognition.

Despite the great success of end-to-end frameworks in text recognition, there are remaining problems. One problem is the inaccurate character position alignment, although the final recognition result (transcript) is correct. This is due to the mismatch between the confidence peak and the true position of the character in CTC-based methods [19]. For the attention-based methods, the current decoding step depends on outputs of previous steps, so once attention maps deviate from the character position, the accumulation of errors will appear [3]. Besides, the model confidence will also directly affect the recognition accuracy. With more accurate model confidence, higher model recognition performance can be reached.

To improve the character alignment and alleviate the overconfidence problem of the state-of-the-art frameworks, we propose an end-to-end text recognition method using convolutional prototype network (CPN) [37], and most aligned frame based CTC training. CPN is used to replace conventional CNN for sliding-window-based character classification based on the nearest prototype in convolutional feature space. In our prototype learning framework, the *blank* symbol can be regarded as the rejection of character classes with more reliable confidence than linear classification. In CPN training, the prototype loss (PL) loss is similar to the maximum likelihood regularization proposed in [17], which can improve the intra-class compactness in feature representation. To better differentiate between character classes and background (*blank*), we use one-vs-all prototype learning [38]. Also, to better exploit character samples in CPN training, we propose a Most Aligned Frame Selection (MAFS) based training strategy. By estimating the most aligned frames of characters in a text image, the sequence labeling problem is transformed into a character classification problem, thereby both the network training and the recognition are improved. We conducted experiments on four handwritten text datasets ORAND-CAR, CVL HDS (digit strings), IAM (English), and ICDAR-2013 (Chinese). The experi-

mental results demonstrate the superiority of the proposed method compared with the baseline, and the benefits of both CPN and MAFS are justified.

The rest of this paper is organized as follows. Section 2 reviews some related works. Section 3 describes our proposed methods with CPN and MAFS based training. Section 4 presents our experimental results, and Section 5 draws concluding remarks.

2 Related Work

2.1 Text Recognation

Before the prosperity of end-to-end text recognition methods, over-segmentation-based methods with character classification and Hidden Markov Model (HMM) modeling were mostly used for Chinese and Latin handwriting [30] [6]. Along with advances in deep learning, end-to-end methods have gradually become dominant in text recognition. Shi et al. [24] proposed an end-to-end RNN-CTC framework, which improves the image feature representation with CNN. This Convolutional RNN (CRNN) has gained success in various scenarios. For example, Zhan et al. [42] applied the RNN-CTC model to handwritten digit string recognition and obtained improved recognition accuracy. Ly et al. [20] used the CRNN framework in historical document recognition. Yin et al. [40] proposed a pure CNN-based model with sliding window classification and CTC decoding. With much fewer parameters and faster calculation speed than CRNN, the sliding-window-based model achieves better recognition results. As for attention-based methods, Shi et al. [25] first proposed an end-to-end framework with RNN and attention for scene text recognition. Since then, attention-based methods [4] [26] [32] have become popular in text recognition tasks. Recently, Bartz et al. [1] replaced the RNN-attention part with the transformer and achieved further improvements.

2.2 Prototype Learning

Prototype learning is a classical and representative method in pattern recognition. The predecessor of prototype learning is k-nearest-neighbor (KNN) classification. To reduce the storage space and computation resources of KNN, prototype reduction and learning methods (including learning vector quantization (LVQ) [11]) have been proposed. Among the methods, some designed suitable updating conditions and rules to learn the prototypes [15] [13], while others treated prototypes as learnable parameters and learned the prototypes through optimizing the related loss functions [22] [23] [9]. A detailed review and evaluation of prototype learning methods can be found in [16]. As for text recognition, prototype learning [17] [30] is also widely used before the advent of deep learning. Previous prototype-based methods mainly employ hand-crafted features before the arrival of CNN. Yang et al. [37] combined the prototype classifier with deep convolutional neural networks and proposed a convolutional prototype network

(CPN) for high accuracy and robust pattern classification. They show that CPN with one-vs-all training can give better performance of open set recognition (classifying known-class samples while rejecting unknown-class samples) [38].

3 Method

We use the sliding-window model and CTC training proposed in [40] as the text recognition framework (see Fig.1). Based on the CTC analysis, we firstly transform the CTC loss into a cross-entropy between pseudo-label distributions and probabilities output by the neural network. Then the cross-entropy loss is improved based on the CPN model and MAFS method for robust text recognition.

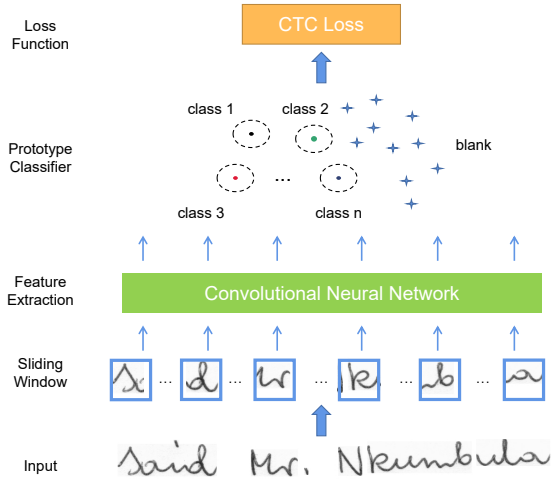


Fig. 1: An illustration of our text recognition based on convolutional prototype learning

3.1 Outline and analysis of CTC

In the CTC recognition framework, the input is a T length sequence $Y = \{y^1, y^2, ...y^T\}$, where y^i is a L'-dimension vector from the neural network output. Except for the L characters to be recognized, there is also a *blank* class, so $L' = L + blank$.

CTC is not only a loss but also a decoding algorithm B i.e. removing the repeated labels then all blanks from the given path. By concatenating one prediction of each frame at all time-steps, a path is formed. The probability of a path is defined as

$$p(\pi|Y) = \prod_{t=1}^{T} y_{\pi_t}^t, \forall \pi \in L'^T, \tag{1}$$

where $y_{\pi_t}^t$ is the probability passing through path π at frame t. Given a sequence label l, a feasible path is defined as the path that can map onto l via B. During training, the probabilities of all feasible paths are added up as the posterior probability $P(l|Y)$ and the negative logarithm of $P(l|Y)$ is taken as the objective function:

$$p(l|Y) = \sum_{\pi \in B^{-1}(l)} p(\pi|Y). \tag{2}$$

$$Loss_{CTC} = -\log p(l|Y), \tag{3}$$

Then the loss partial differential concerning y_k^t is computed as:

$$\frac{\partial Loss_{CTC}}{\partial y_k^t} = -\frac{1}{p(l|Y) y_k^t} \sum_{\{\pi | \pi \in B^{-1}(l), \pi_t = k\}} p(\pi|Y). \tag{4}$$

We rewrite Eq.(4) as

$$\frac{\partial Loss_{CTC}}{\partial y_k^t} = -\frac{\sum_{\{\pi | \pi \in B^{-1}(l), \pi_t = k\}} p(\pi|Y)}{p(l|Y)} \frac{\partial \log y_k^t}{\partial y_k^t} = -z_k^t \frac{\partial \log y_k^t}{\partial y_k^t}, \tag{5}$$

where we regard $z_k^t = \frac{\sum_{\{\pi | \pi \in B^{-1}(l), \pi_t = k\}} p(\pi|Y)}{p(l|Y)}, k = 0, ..., L' - 1$, as the pseudo-label distribution in the CTC decoding graph at frame t. When z_k^t is regarded as a constant, we can find that Eq.(5) is a derivative form of a cross-entropy between z_k^t and y_k^t. So CTC loss is equivalent to the cross-entropy between pseudo-label distributions and classifier outputs:

$$Loss_{CE} = Loss_{CTC} = -\sum_t \sum_k z_k^t \log y_k^t. \tag{6}$$

Based on the above formulation, we can divide CTC training in each iteration into two steps (see Fig.2): pseudo-label estimation and cross-entropy training. The first step is to estimate the pseudo-label distribution for each frame using the model output Y and the ground truth l. Secondly, update the model parameters with the cross-entropy criteria. In this step, the pseudo-label distribution plays a similar role to the one-hot label used in the classification task. Therefore, CTC loss is an alternate-updating process, where the classifier output and pseudo-label distribution interact with each other and become more and more accurate.

As we know in [7], to facilitate the calculation of loss, CTC introduces a one-way graph G based on the extended ground truth l'. For example in Fig.2, ground truth $l = \{A, P, P\}$ and $l' = \{blank, A, blank, P, blank, P, blank\}$, and $l'(n)$ means the character of the n-th element in l'. Following this setting and the definition of pseudo-label distribution z^t, we can define raw pseudo-label

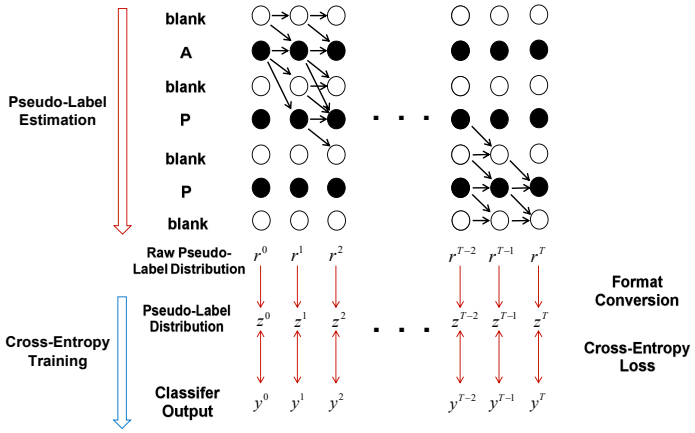


Fig. 2: CTC in two steps

distribution as $r_n^t = \frac{\sum_{\{\pi|\pi\in B^{-1}(l),\pi_t=l'(n)\}} p(\pi|Y)}{p(l|Y)}$, which is the probability distribution in the CTC decoding graph at frame t concerning to the n-th node. Using raw pseudo-label distribution r_n^t, we can compute pseudo-label distribution as $z_k^t = \sum_{l'(n)=k} r_n^t$. For the class k not appearing in the ground truth (except for *blank*), $z_k^t = 0$.

Eq.(6) indicates that reliable confidence can improve the pseudo-label estimation, thereby making the model performance accurate. So we use CPN instead of the CNN model for a better confidence estimation. On the other hand, the most aligned frames corresponding to the ground truth are more important for training. So we can use the MAFS method to improve CTC training.

3.2 One-vs-All Prototype Learning

In the CTC recognition framework, the classifier-output confidence y_k^t is directly used for training or decoding as in Eq.(6). Although the linear classifier has achieved excellent recognition results in [40], the confidence is still not robust enough. Therefore, we use the convolutional prototype network to improve pseudo-label estimation and cross-entropy training in CTC. Prototype learning is to train one or more prototypes for each class and use the idea of template matching to classify samples. In our work, for simplicity, we use only one prototype for each character class. We assume that each character class presents a standard Gaussian distribution in the feature space, so Euclidean distance is used to describe the similarity between each sample and the prototype. But as for *blank*, a class describing all non-character samples in the CTC framework, Gaussian assumption seems unreasonable. So we choose not to learn the prototype of *blank* but estimate its probability as the rejection of characters. Then we take inspiration from Liu's [14] work and merge multiple two-class classifiers to build One-vs-All prototype learning.

We use the prototypes of L classes to construct L two-class classifiers,

$$m_k^t = Sigmoid(-\tau(d_k^t - T_k)), \tag{7}$$

$$d_k^t = ||f(x^t) - p_k||_2, \tag{8}$$

where d_k is the Euclidean distance between the feature map $f(x^t)$ of frame t and the prototype p_k of class k. T_k is a learnable threshold and τ is a predefined temperature coefficient, which is set to 5 in our experiment. We can regard m_k^t as the confidence of a two-class classifier whether frame t belongs to class k. For the L classifiers, we follow the principle in [14] to calculate the confidence y_k^t in Eq.(6),

$$y_k^t = A^{-1} m_k \prod_{k^o \neq k} (1 - m_{k^o}^t), \qquad k \in \{k | k \neq blank\} \tag{9}$$

$$y_{blank}^t = A^{-1} \prod_{k^o} (1 - m_{k^o}^t) \tag{10}$$

$$A^t = \sum_{k \neq blank} m_k^t \prod_{k^o \neq k} (1 - m_{k^o}^t) + \prod_{k^o} (1 - m_{k^o}^t), \tag{11}$$

where A^t is a normalization factor.

In addition to optimize the cross-entropy in Eq.(6), the prototype loss is also added according to Yang et al. [37],

$$Loss = Loss_{CE} + \alpha Loss_{pl}, \tag{12}$$

where α is set to 0.01 in our experiment.

The difficulty of applying prototype learning to text recognition is that the ground truth of each frame is unknown, so it is impossible to gather samples of a certain class around the corresponding prototype. With the pseudo-label distribution z^t for weighting, this problem can be solved,

$$Loss_{pl} = \sum_t \sum_{k \neq blank} z_k^t ||f(x^t) - p_k||_2. \tag{13}$$

Based on the prototype classifier and its loss function, we can make text recognition more robust.

3.3 Most Aligned Frame Selection Based Training

As discussed at the end of Sec.3.1, by selecting the most aligned frames with the ground truth for training, the model can converge better.

Based on the raw pseudo-label distribution r^t, the probability of frame t aligned with the c-th character in the ground truth can be computed as $F_c(t) =$

$\frac{r^t_{2c+1}}{\sum_t r^t_{2c+1}}$, where we regard t as a random variable confirming to Gaussian distribution. Then the most aligned frame can be achieved with the expectation of t,

$$t_c = Round(\sum_t t \cdot \frac{r^t_{2c+1}}{\sum_t r^t_{2c+1}}), \tag{14}$$

where t_c means the most aligned frame of the c-th character in the ground truth and $Round(\cdot)$ is a rounding function. Here, smaller is the variance, more credible is expectation estimation. So only when the variance is smaller than a certain threshold (1 in our work), the most aligned frame of the character can be used for training. Otherwise, if the decoding result with pseudo-label distribution is consistent with the ground truth, the probability distribution r^t is confident and then the most aligned frames of this text sample can also be used for training. A schematic diagram of the method can be seen in Fig. 3.

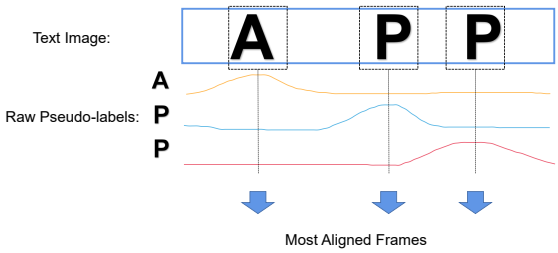


Fig. 3: An illustration of most aligned frame selection

In the training process, for high confidence of most aligned frames, we use the selected most aligned frame with the one-hot label of the corresponding class. Besides, to reduce the influence of other frames and *blank* frames, we still use the pseudo-label distribution but multiply by the weakening coefficients γ before the loss. The new CE loss function can be written as,

$$Loss_{CE} = -\gamma \sum_{(t,k)\notin\Omega} z^t_k \log y^t_k - \sum_{(t,k)\in\Omega} \log y^t_k, \tag{15}$$

where Ω indicates the set of the most aligned frames with labels. In the experiment, we choose 0.5 as the weakening coefficient γ of the non-most aligned frame.

4 Experiments

4.1 Datasets

In our experiment, four public datasets are used to evaluate our handwritten text recognition method. Two of them are handwritten digit strings named ORAND-CAR [5] and Computer Vision Lab Handwritten Digit String (CVL HDS, or CVL

for short) [5], the third is an English text line dataset named IAM [21] and the last one is a Chinese handwritten dataset named ICDAR-2013 [39].

ORAND-CAR consists of 11719 images obtained from the Courtesy Amount Recognition (CAR) field of real bank checks. It can be divided into two sub-datasets CAR-A and CAR-B with different data sources. CAR-A has 2009 images for training and 3784 images for testing, while CAR-B consists of 3000 training images and 2926 testing images.

CVL HDS has been collected mostly amongst students of the Vienna University of Technology. 7960 images from 300 writers are collected, where only 1262 images are in the training set and the other 6698 images are for testing.

IAM contains unconstrained handwritten texts collected from 657 writers. As an English handwritten dataset, there are 79 categories, including numbers, letters, and punctuation. It contains 6,482 lines in the training set, 976 lines in the validation set, and 2,915 lines in the test set.

As for the Chinese handwritten dataset, we set the training set as **CASIA-HWDB** [18], which is divided into six sub-datasets. CASIA-HWDB1.0-1.2 consists of individual character samples, while CASIA-HWDB2.0-2.2 samples are handwritten text line images. There are 3118477 character images with 7356 classes and 41781 text lines with 2703 categories in the training set (816 writers out of 1020). For the test dataset **ICDAR-2013**, there are 3432 text line images. In our experiments, training sets of CASIA-HWDB are used for the model training.

All datasets are the only line labeled without character boundaries. So for the alignment experiment, we use MNIST handwritten digital dataset [12] for string synthesis and model evaluation.

4.2 Implementation Details

We use the sliding-window-based model [40] as the baseline, where the text image is divided into multiple windows equidistantly and then directly recognized by CNN. We use the same network structure as [40], but choose different sizes of windows according to different databases. In digit string recognition task, images are resized and padded to 32×256, and multi-scale windows are used with the size of 32×24, 32×28, 32×32. As for IAM dataset, image height is scaled to 32, width is scaled proportionally and multi-scale window sizes are set to 32×24, 32×32, 32×40. Models shift with step 4 in both experiments. For the Chinese handwritten dataset, we use character images in CASIA-HWDB1.0-1.2 training set to synthesize 1,250,000 text images and train the network together with the real text samples in CASIA-HWDB2.0-2.2. We scale the image to a width of 64, the multi-scale window to be 64×48, 64×64, 64×80, and the window step size to 8.

In the training process of digit string recognition, we first use the Adam optimizer [10] to train our network with a batch size of 32. The initial learning rate is 3×10^{-4}. After trained for 50 epochs, we switch to Stochastic gradient descent (SGD) with a learning rate 1×10^{-4}. After another 50 epochs, the learning rate is reduced by 0.3 times and then trained again for 50 epochs. For CVL HDS, due

Table 1: String accuracies of different models on the handwritten digital dataset.

Methods	CAR-A	CAR-B	CVL HDS
Pernambuco [5]	0.7830	0.7543	0.5860
BeiJing [5]	0.8073	0.7013	**0.8529**
FSPP [29]	0.8261	0.8332	0.7923
CRNN [24]	0.8801	0.8979	0.2601
ResNet-RNN [42]	0.8975	0.9114	0.2704
DenseNet [41]	0.9220	0.9402	0.4269
Sliding-Window [40]	0.9337	0.9357	0.8010
Sliding-Window + CPN	0.9430	0.9445	0.8425
Sliding-Window + MAFS	0.9447	0.9425	0.8356
Sliding-Window + CPN + MAFS	**0.9483**	**0.9470**	0.8512

to the lack of training samples, a model with random initialization is not easy to converge. So we use the model trained on CAR-A for ten epochs as initialization. In the handwritten English and Chinese recognition task, we only use the Adam optimizer with the initial learning rate of 3×10^{-4}, randomly initialized model can converge well.

4.3 Comparison with the State-of-the-art Methods

For handwritten digital strings, we compare our methods with state-of-the-art approaches in Table 1. On the ORAND-CAR dataset, our method achieves the best performance and can reduce the error rate by 25% in the best case. On the CVL dataset, based on careful initialization, our end-to-end framework has almost the same performance as Beijing [5] and achieves state-of-the-art recognition accuracy among the deep-learning-based methods. The Beijing method manually segments each training text image into characters for classifier training, while our method can train the model with only line labels and so is more practical for practical application.

Since the digit string datasets do not have context information, it is not suitable for attention-based methods. But for English text recognition, the attention-based model is also listed in Table 2. In the comparison, we only scale each image to a height of 32 but achieve better performance. As shown in Table 2, our proposed method has achieved the best recognition results in both character error rate (CER) and word error rate (WER). We visualize some recognition results in Fig. 4.

We also conducted experiments on the Chinese handwriting dataset, using the 5-gram statistical model trained by Wu et al. [36]. The experimental results are shown in Table 3. We can also find that in large-category database, the training strategy based on the most aligned frame and the character classifier based on the convolution prototype can still improve the recognition performance of the model, which also verifies the effectiveness of our methods.

Table 2: Results of different models on the dataset IAM. (CER: character error rate; WER: word error rate.)

Methods	CER	WER
Salvador et al. [6]	9.8	22.4
Bluche [2]	7.9	24.6
Sueiras et al. [28]	8.8	23.8
Zhang et al. [43]	8.5	22.2
DAN [32]	6.4	19.6
Sliding-Window	6.6	18.8
Sliding-Window + CPN	6.1	18.1
Sliding-Window + MAFS	6.2	17.9
Sliding-Window + CPN + MAFS	**5.8**	**17.8**

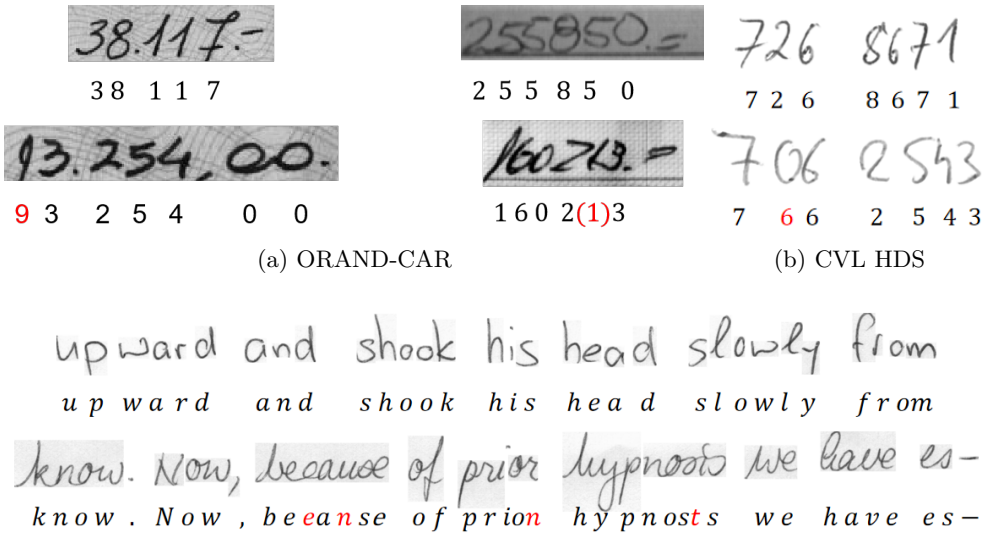
Table 3: Results on the Chinese handwritten text dataset ICDAR-2013. (CR: correct rate; AR: accurate rate [35])

Methods	Without LM		With LM	
	CR	AR	CR	AR
Wu et al. [35]	87.43	86.64	-	92.61
Wang et al. [31]	90.67	88.79	95.53	94.02
Wu et al. [36]	-	-	96.32	96.20
Wang et al. [34]	89.66	-	96.47	-
Wang et al. [33]	89.12	87.00	95.42	94.83
Sliding-Window	89.03	88.65	95.89	95.35
Sliding-Window + MAFS	90.71	90.16	96.43	96.15
Sliding-Window + CPN + MAFS	**90.92**	**90.30**	**96.64**	**96.23**

We believe that our convolutional prototype classifier is more reasonable than linear classifiers, where *blank* is the rejection of character classes. Besides, the model can converge better due to more reliable confidence. That is why CPN can improve recognition performance in different datasets. As for our proposed MAFS, sparser pseudo-label distribution can reduce the error caused by pseudo-label estimation in the training process. Besides, most aligned frames can be paid more attention in training by reducing the impact of other frames. So our work can improve recognition performance.

4.4 Robustness and Alignment Evaluation

In this part, we analyze the effect of CPN and MAFS on model alignment and robustness. As far as we know, there is almost no evaluation standard of alignment effect in the field of text recognition, so we propose an indirect experiment for



(a) ORAND-CAR (b) CVL HDS

(c) IAM

Fig. 4: Visualization results of the datasets used in our experiments.

Table 4: Recognition rates of different models on the datasets MNIST.

	MNIST string	MNIST character
Sliding-Window + CNN	0.934	0.942
Sliding-Window + MAFS	0.932	0.956
Sliding-Window + CPN	0.939	**0.983**

alignment comparison. We train the model with string samples synthesized by MNIST digital images and compare the character classification accuracy on the MNIST test set. We believe the higher accuracy of character classification, the more character-aligned frames are classified correctly in sequence recognition. It also indirectly describes the alignment effect of the model. In the experiment, we randomly select samples in the MNIST dataset and splice them into strings with a length of 5 to 8. When the recognition accuracy with sequence samples is similar, character classification accuracy can be a standard for comparing model alignment effects. As shown in Table 4, with comparable recognition accuracy, CPN and MASF have higher classification accuracy than CNN, which shows that they lead to better alignment performance.

We also visualize the feature representation learned by CPN on the CAR database. Although the ground truth per frame is not available, we choose the category predicted by pseudo-label distribution for each frame as the label and draw a scatter plot. In Fig. 5, different colors represent different classes. The

black dots represent the coordinate in feature dimension of prototypes, and *blank* frames are not in this figure. It can be seen from Fig. 5 that samples cluster near prototypes in the feature dimension, which proves that CPN has a robust feature representation.

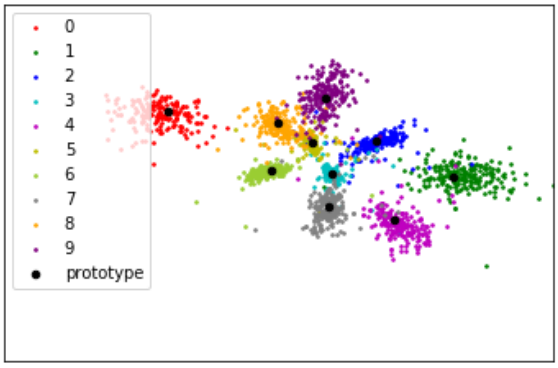


Fig. 5: Feature representation learned by CPN model on CAR

5 Conclusions

In this paper, we propose a method for handwritten text recognition using convolutional prototype network for character classification and most aligned frame based CTC training. Different from previous CTC-based methods, we regard *blank* as the rejection of character classes and design a one-vs-all prototype classifier. The training strategy is based on the most aligned frame selection so as to improve the accuracy of character location and classification. Experiments on four handwritten text datasets confirm that our proposed methods can effectively improve text recognition performance, and the valuation on MNIST also verifies that CPN is beneficial for better alignment and model robustness. The proposed framework will be applied to more recognition scenarios (including scene text recognition) for further evaluation and improvement.

Acknowledgements

This work has been supported by the National Key Research and Development Program Grant 2020AAA0109702, the National Natural Science Foundation of China (NSFC) grants 61936003, 61721004.

References

[1] Bartz, C., Bethge, J., Yang, H., Meinel, C.: Kiss: Keeping it simple for scene text recognition. arXiv preprint arXiv:1911.08400 (2019)

[2] Bluche, T.: Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. arXiv preprint arXiv:1604.08352 (2016)

[3] Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., Zhou, S.: Focusing attention: Towards accurate text recognition in natural images. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5076–5084 (2017)

[4] Cheng, Z., Xu, Y., Bai, F., Niu, Y., Pu, S., Zhou, S.: Aon: Towards arbitrarily-oriented text recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5571–5579 (2018)

[5] Diem, M., Fiel, S., Kleber, F., Sablatnig, R., Saavedra, J.M., Contreras, D., Barrios, J.M., Oliveira, L.S.: Icfhr 2014 competition on handwritten digit string recognition in challenging datasets (hdsrc 2014). In: Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition. pp. 779–784. IEEE (2014)

[6] Espana-Boquera, S., Castro-Bleda, M.J., Gorbe-Moya, J., Zamora-Martinez, F.: Improving offline handwritten text recognition with hybrid hmm/ann models. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(4), 767–779 (2010)

[7] Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning. pp. 369–376 (2006)

[8] Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(5), 855–868 (2008)

[9] Huang, Y.S., Liu, K., Suen, C.Y., Shie, A., Shyu, I., Liang, M., Tsay, R., Huang, P.: A simulated annealing approach to construct optimized prototypes for nearest-neighbor classification. In: Proceedings of 13th International Conference on Pattern Recognition. vol. 4, pp. 483–487. IEEE (1996)

[10] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[11] Kohonen, T.: The self-organizing map. Proceedings of the IEEE **78**(9), 1464–1480 (1990)

[12] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)

[13] Lee, S.W., Song, H.H.: Optimal design of reference models for large-set handwritten character recognition. Pattern Recognition **27**(9), 1267–1274 (1994)

[14] Liu, C.L.: Classifier combination based on confidence transformation. Pattern Recognition **38**(1), 11–28 (2005)

[15] Liu, C.L., Eim, I.J., Kim, J.H.: High accuracy handwritten chinese character recognition by improved feature matching method. In: Proceedings of the Fourth International Conference on Document Analysis and Recognition. vol. 2, pp. 1033–1037. IEEE (1997)
[16] Liu, C.L., Nakagawa, M.: Evaluation of prototype learning algorithms for nearest-neighbor classifier in application to handwritten character recognition. Pattern Recognition **34**(3), 601–615 (2001)
[17] Liu, C.L., Sako, H., Fujisawa, H.: Effects of classifier structures and training regimes on integrated segmentation and recognition of handwritten numeral strings. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(11), 1395–1407 (2004)
[18] Liu, C.L., Yin, F., Wang, D.H., Wang, Q.F.: Casia online and offline chinese handwriting databases. In: Proceedings of the International Conference on Document Analysis and Recognition. pp. 37–41. IEEE (2011)
[19] Liu, H., Jin, S., Zhang, C.: Connectionist temporal classification with maximum entropy regularization. In: Proceedings of the Advances in Neural Information Processing Systems. pp. 831–841 (2018)
[20] Ly, N.T., Nguyen, K.C., Nguyen, C.T., Nakagawa, M.: Recognition of anomalously deformed kana sequences in japanese historical documents. IEICE Transactions on Information and Systems **102**(8), 1554–1564 (2019)
[21] Marti, U.V., Bunke, H.: The iam-database: an english sentence database for offline handwriting recognition. International Journal on Document Analysis and Recognition **5**(1), 39–46 (2002)
[22] Sato, A., Yamada, K.: Generalized learning vector quantization. Advances in Neural Information Processing Systems pp. 423–429 (1996)
[23] Sato, A., Yamada, K.: A formulation of learning vector quantization using a new misclassification measure. In: Proceedings of the Fourteenth International Conference on Pattern Recognition. vol. 1, pp. 322–325. IEEE (1998)
[24] Shi, B., Bai, X., Yao, C.: 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 (2016)
[25] Shi, B., Wang, X., Lyu, P., Yao, C., Bai, X.: Robust scene text recognition with automatic rectification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4168–4176 (2016)
[26] Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X.: Aster: An attentional scene text recognizer with flexible rectification. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(9), 2035–2048 (2018)
[27] Su, B., Lu, S.: Accurate scene text recognition based on recurrent neural network. In: Proceedings of the Asian Conference on Computer Vision. pp. 35–48. Springer (2014)
[28] Sueiras, J., Ruiz, V., Sanchez, A., Velez, J.F.: Offline continuous handwriting recognition using sequence to sequence neural networks. Neurocomputing **289**, 119–128 (2018)
[29] Wang, Q., Lu, Y.: A sequence labeling convolutional network and its application to handwritten string recognition. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 2950–2956 (2017)

[30] Wang, Q.F., Yin, F., Liu, C.L.: Handwritten chinese text recognition by integrating multiple contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(8), 1469–1481 (2011)
[31] Wang, S., Chen, L., Xu, L., Fan, W., Sun, J., Naoi, S.: Deep knowledge training and heterogeneous cnn for handwritten chinese text recognition. In: Proceedings of the 15th International Conference on Frontiers in Handwriting Recognition. pp. 84–89. IEEE (2016)
[32] Wang, T., Zhu, Y., Jin, L., Luo, C., Chen, X., Wu, Y., Wang, Q., Cai, M.: Decoupled attention network for text recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 12216–12224 (2020)
[33] Wang, Z.X., Wang, Q.F., Yin, F., Liu, C.L.: Weakly supervised learning for over-segmentation based handwritten chinese text recognition. In: Proceedings of the 17th International Conference on Frontiers in Handwriting Recognition. pp. 157–162. IEEE (2020)
[34] Wang, Z.R., Du, J., Wang, W.C., Zhai, J.F., Hu, J.S.: A comprehensive study of hybrid neural network hidden markov model for offline handwritten chinese text recognition. Proceedings of the International Journal on Document Analysis and Recognition **21**(4), 241–251 (2018)
[35] Wu, Y.C., Yin, F., Chen, Z., Liu, C.L.: Handwritten chinese text recognition using separable multi-dimensional recurrent neural network. In: Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition. vol. 1, pp. 79–84. IEEE (2017)
[36] Wu, Y.C., Yin, F., Liu, C.L.: Improving handwritten chinese text recognition using neural network language models and convolutional neural network shape models. Pattern Recognition **65**, 251–264 (2017)
[37] Yang, H.M., Zhang, X.Y., Yin, F., Liu, C.L.: Robust classification with convolutional prototype learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3474–3482 (2018)
[38] Yang, H.M., Zhang, X.Y., Yin, F., Yang, Q., Liu, C.L.: Convolutional prototype network for open set recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, early access (2020)
[39] Yin, F., Wang, Q.F., Zhang, X.Y., Liu, C.L.: Icdar 2013 chinese handwriting recognition competition. In: Proceedings of the 12th International Conference on Document Analysis and Recognition. pp. 1464–1470. IEEE (2013)
[40] Yin, F., Wu, Y.C., Zhang, X.Y., Liu, C.L.: Scene text recognition with sliding convolutional character models. arXiv preprint arXiv:1709.01727 (2017)
[41] Zhan, H., Lyu, S., Lu, Y.: Handwritten digit string recognition using convolutional neural network. In: Proceedings of the 24th International Conference on Pattern Recognition. pp. 3729–3734. IEEE (2018)
[42] Zhan, H., Wang, Q., Lu, Y.: Handwritten digit string recognition by combination of residual network and rnn-ctc. In: Proceedings of the International Conference on Neural Information Processing. pp. 583–591. Springer (2017)
[43] Zhang, Y., Nie, S., Liu, W., Xu, X., Zhang, D., Shen, H.T.: Sequence-to-sequence domain adaptation network for robust text image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2740–2749 (2019)