Improving Offline Handwritten Chinese Character Recognition by Iterative Refinement

Xiao Yang, Dafang He, Zihan Zhou, Daniel Kifer, C. Lee Giles
The Pennsylvania State University
xuy111@psu.edu duh188@psu.edu zzhou@ist.psu.edu dkifer@cse.psu.edu giles@ist.psu.edu

***Abstract*—We present an *iterative refinement module* that can be applied to the output feature maps of any existing convolutional neural networks in order to further improve classification accuracy. The proposed module, implemented by an attention-based recurrent neural network, can iteratively use its previous predictions to update attention and thereafter refine current predictions. In this way, the model is able to focus on a sub-region of input images to distinguish visually similar characters (see Figure 1 for an example). We evaluate its effectiveness on handwritten Chinese character recognition (HCCR) task and observe significant performance gain. HCCR task is challenging due to large number of classes and small differences between certain characters. To overcome these difficulties, we further propose a novel convolutional architecture that utilizes both low-level visual cues and high-level structural information. Together with the proposed iterative refinement module, our approach achieves an accuracy of 97.37%, outperforming previous methods that use raw images as input on ICDAR-2013 dataset [1].**

I. Introduction

Handwritten Chinese character recognition (HCCR) has been a long-standing research problem. A successful HCCR module can support various practical systems such as mail sorting, paycheck processing, documents digitalization and retrieval. The main difficulties for existing approaches include a large number of class labels (e.g. an educated Chinese knows 6000 to 8000 characters [2]), mispredictions between visually similar characters (such as "已" which means "already" and "己" which means "self"), and distinct handwriting styles across individuals.

Depending on the representation of the input data, HCCR problem can be further categorized into two sub-problems: 1) online HCCR, where input characters are represented by the trajectories of pen tip movements; and 2) offline HCCR, where images containing isolated character are the input. This work focuses on improving the second, since for many applications (e.g. mail sorting) pen trajectories are not available.

Early work on offline HCCR often relies on hand-crafted features, while recent advances in deep convolutional neural networks (CNN) [3] enable direct learning of visual representations from raw data which has led to state-of-the-art results. However, it is still difficult to distinguish certain characters from those that are visually similar. For this we have created an iterative refinement module that takes as input the feature maps learned by a convolutional neural network. The iterative refinement module is implemented with an attention-based recurrent neural network (RNN) [4], [5]. Given an input image x depicting a single handwritten Chinese character, we first learn a convolutional visual representation $V(x)$, then apply the iterative refinement module to obtain an initial prediction y_1. Here y_t $(t = 1, 2, \cdots, T)$ is a vocabulary-sized vector representing the probability of yielding each character. Based on y_1, the module is expected to concentrate its attention on a sub-region of $V(x)$ that is hypothetically more informative to distinguish visually similar characters (see Figure 1). Consequently, a refined prediction y_2 is outputted. The aforementioned process can repeat T times until we are satisfied with the last prediction. The intuition behind is that instead of trying to solve a complicated classification problem in one shot, we decompose the problem using a coarse-to-refined approach.

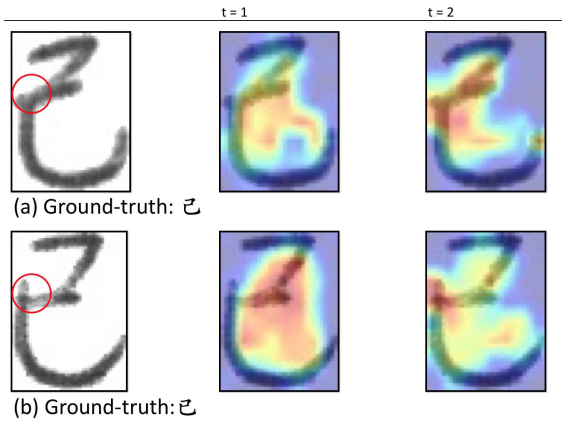

Fig. 1. Two handwritten Chinese characters that look very similar to each other. The major difference is shown in the red circle of the input images. The visualized attention maps are explained in Section III-A.

Figure 1 shows two Chinese characters that can be easily confused for each other. As shown, the attention (learned by our model) used for outputting y_1 is roughly evenly spread, while that used for outputting y_2 is concentrated on the circled region. Therefore, the model is more likely to tell the difference between these two characters.

As both low-level visual cues (e.g. small strokes) and high-level structural knowledge are beneficial for HCCR task, we propose a novel convolutional architecture to utilize a hierarchy of visual representations. It adopts residual blocks [6] to facilitate training deep networks and shortcut connections to aggregate multi-scale information. Therefore the proposed architecture is termed as *multi-scale residual block cascade*. Figure 2 illustrates the pipeline of our model while Figure 4

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.11

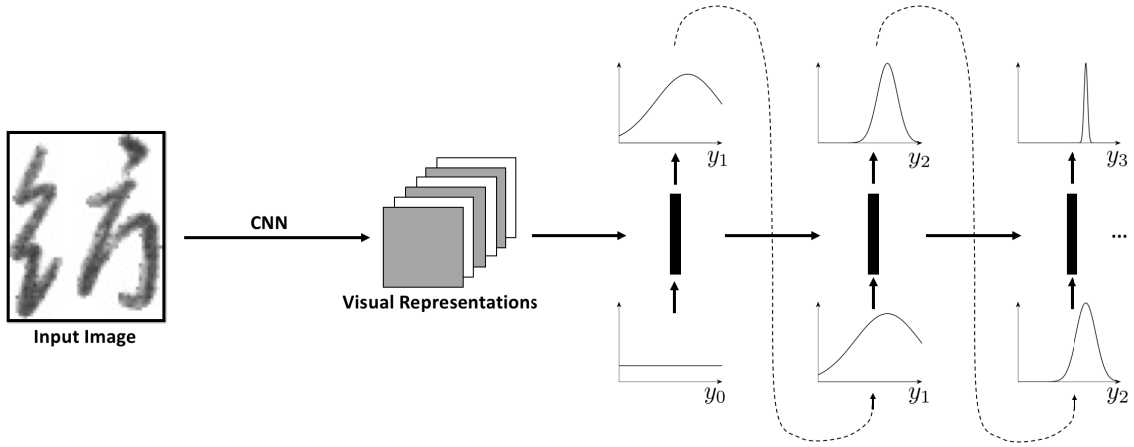

Fig. 2. The proposed model consists of two parts: 1) a multi-scale residual block cascade that learns a hierarchy of visual features from the input image; and 2) an iterative refinement module that iteratively updates attentions and refines current predictions. y_0 is set as a uniform distribution while the most likely character from the last refined prediction y_T is the model's final output.

shows the architecture of the multi-scale residual block cascade.

The proposed method is evaluated on the ICDAR-2013 offline HCCR dataset [1], which contains 224,419 test images covering 3,755 frequently used Chinese characters. Our method achieves a prediction accuracy of 97.37%, outperforming existing methods that use raw images as input. The DirectMap+ConvNet+Adaption method [7] achieved the same accuracy, however it requires a time-consuming pre-processing step while our method can be directly applied on raw images. Overall our method is about 4 times faster than [7]. The contributions of our work can be summarized as follows:

- We present an iterative refinement module which can be trained with a convolutional neural network in an end-to-end manner. It is shown in experiments that the proposed module can robustly increase classification accuracy by iteratively update attentions and predictions.
- We propose a novel convolutional architecture to utilize both low-level visual cues and high-level information. Together with the iterative refinement module, we achieve an accuracy of 97.37% on ICDAR-2013 dataset, outperforming other methods that use raw images as input.

II. Related Work

Handwritten Chinese Character Recognition: Based to the types of input data, HCCR can be divided into online and offline problems. For online HCCR, researchers often follow a conventional pipeline where they first extract features from the recorded sequence of coordinates, then apply classification models to obtain predictions. For example, [8], [9] first detected "key" coordinates that indicate changes in strokes (e.g. corners and segment ends), then employed a nearest neighbor classifier to match from a pre-defined template database. In [10], [11], stroke types were first recognized using finite state automaton, then fed into a nearest-neighbor based model for classification. Recently, [12] proposed using an RNN to simultaneously learn representations from raw sequence of coordinates and perform classification, circumventing the need for feature engineering.

Methods for online HCCR suffer from various stroke orders and computational expensiveness when processing very long sequences. For offline HCCR, these challenges no longer exist. Early work on offline HCCR focus more on the design of visual features for classification models. With the recent success of deep CNN in vision tasks, a number of CNN-based models have been proposed, leading to rapid improvements in performance. In [13], a multi-column CNN model was presented which ensembled multiple parallel networks to enlarge model capacity. [14] proposed a spatially-sparse CNN model to speedup training time for deeper networks. Despite the different architectures in use, their models are still in the form of a convolutional representation learning part followed by a softmax classifier. Our model, on the other hand, replaces the softmax classifier with an iterative refinement module to iteratively update attentions and predictions. The proposed module is general in the sense that it can be applied to the output feature maps of any pre-trained CNN-based models to further improve performance.

Knowledge Distillation: The proposed iterative refinement module can be seen as an iterative version of *knowledge distillation*, a concept introduce by Hinton et al. [15] in 2015. Hinton et al. attempted to train a small neural network model by matching its class probabilities to the output of an already trained large model, instead of the ground-truth labels. In this way, the knowledge of the large model is transfered to the small model which are more suitable for deployment. The proposed iterative refinement module differs from [15] in that it does not require a pre-trained large model to "distill" knowledge. Instead, the module is self-guided: it relies on its previous prediction to update attentions and current prediction. Besides, the goal of the proposed module is to boost performance, rather than to compress model size as in [15]. Our work is loosely connected to *sequential knowledge distillation* [16], which also proposed using an RNN to distill knowledge for sequential predictions. In [16], a pre-trained

large model was used to predict *next* object (word) given previous translated words for machine translation task. On the contrary, our module iteratively refine the prediction of the *same* object (character), without any pre-trained external models.

III. Method

This section describes the details of our handwritten Chinese character recognition model. Overall, it takes as input an image x depicting an isolated character, and outputs the prediction C.

As shown in Figure 2, our model consists of two parts: 1) a multi-scale residual block cascade (M-RBC) that learns a hierarchy of visual features, and 2) an iterative refinement (IR) module implemented by an attention-based RNN that repeatedly refine predictions. Section III-A describes the formulation of the iterative refinement module while Section III-B describes in detail the architecture of the multi-scale residual block cascade.

A. Iterative Refinement Module

Concretely, assume we have a visual representation $V(x)$ which is a set of K-dimensional vectors (K denotes the number of channels) learned from x using a convolutional representation learning module, and a previous prediction y_{t-1} which is a vocabulary-sized vector. Then, the refined prediction y_t can be calculated by:

$$y_t = \text{IR}(V(x),\ y_{t-1}) \tag{1}$$

This process repeats T times and the last predicted character $\hat{C}_T = \text{argmax}(y_T)$ will be announced as the model's final output. y_0 is set to a uniform distribution.

A usual deep CNN-based model for classification is equivalent to the case where the iterative refinement module is implemented by a feed-forward neural network (e.g. a multi-layer perceptron (MLP)) and T is set to 1:

$$y_1 = \text{MLP}(V(x),\ y_0) \tag{2}$$

In this case, since y_0 is an input-independent uniform distribution, it does not convey any information. When T is larger than 1, the iterative refinement module is in the form of a recurrent neural network. With inspiration from the attention mechanism of human vision system, we implement the iterative refinement module with an attention-based recurrent neural network. Therefore, the module is able to focus on a sub-region of $V(x)$. Formally, we decompose Equation 1 into two steps. First, a context vector ctx_t which is a dynamic representation of the relevant region is learned based on $V(x)$ and y_{t-1}:

$$\text{ctx}_t = f_{att}(V(x),\ y_{t-1}) \tag{3}$$

where f_{att} is an attention model. Then, we compute the refined prediction y_t using a recurrent neural network:

$$y_t = \text{RNN}(\text{ctx}_t,\ y_{t-1}) \tag{4}$$

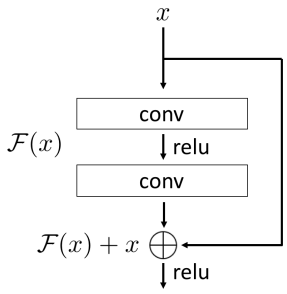

Fig. 3. A convolutional residual block where the operator $\oplus$ denotes element-wise addition. Each convolution layer has a kernel size of 3×3 and is followed by a Batch Normalization [18] layer, which for brevity is omitted here.

Multiple choices exist for the attention model f_{att} such as "hard" attention and "soft" attention [17]. Here we adopt the "soft" attention mechanism where context vector ctx_t is defined as the weighted sum of the learned features $V(x)$:

$$\text{ctx}_t = \sum_{n=1}^{N} \alpha_t^n V^n(x) \tag{5}$$

where $V^n(x)$ is one of the K-dimensional vectors in visual representation $V(x)$:

$$V(x) = \{V^1(x), V^2(x), \cdots, V^N(x)\} \tag{6}$$

$$V^n(x) \in \mathbb{R}^K, \quad |V(x)| = N \tag{7}$$

and weight α_t^n for $V^n(x)$ can be computed by:

$$\alpha_t^n = \frac{\exp(e_t^n)}{\sum_{i=1}^{N} \exp(e_t^i)} \tag{8}$$

Here the alignment score e_t^n indicates how relevant $V^n(x)$ is to the prediction y_t. Following Bahdanau et al. [5], we model e_t^n with a single layer perceptron (SLP) such that:

$$e_t^n = \text{SLP}(y_{t-1},\ V^n(x)) \tag{9}$$

$$= v^\intercal \text{Tanh}(W y_{t-1} + U V^n(x)) \tag{10}$$

with v, W and U being weight matrices to be learned.

It has been observed by [19] that training a plain recurrent neural network is difficult owning to the gradient vanishing or exploding issue during back-propagation. To overcome this, we adopt a variant of recurrent neural network: Gated Recurrent Unit (GRU) [20] to implement the proposed iterative refinement module. GRU contains two gating units that modulate the flow of information: a reset gate r, and a update gate z. Intuitively, r determines how to incorporate the new input with the previous state, while z controls how much of the previous states to keep around. The hidden state y_t at time-step t is therefore computed as follows:

$$r = \sigma(x_t W_r + y_{t-1} U_r) \tag{11}$$

$$z = \sigma(x_t W_z + y_{t-1} U_z) \tag{12}$$

$$\hat{y} = \text{Tanh}(x_t W_{\hat{y}} + (y_{t-1} \circ r) U_{\hat{y}}) \tag{13}$$

$$y_t = (1 - z) \circ \hat{y} + z \circ y_{t-1} \tag{14}$$

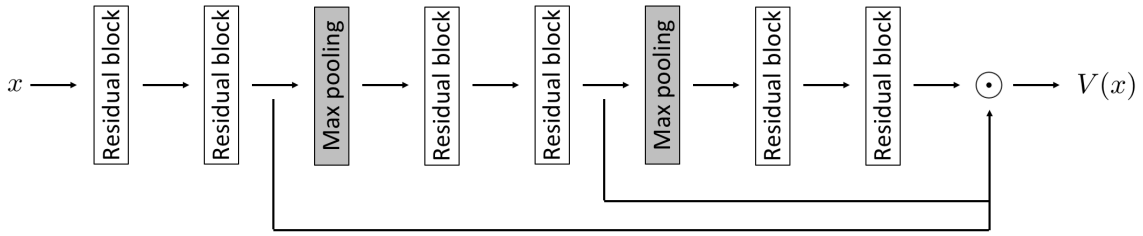

Fig. 4. The proposed multi-scale residual block cascade contains 6 residual blocks and 2 Max-pooling layers whose pooling size is 2×2. The operator $\odot$ denotes aggregation among activations at different scales (see Equation 16).

with σ being a sigmoid function and $\circ$ being the element-wise multiplication. Here x_t is the input at time-step t and $W_r, U_r, W_z, U_z, W_{\hat{y}}, U_{\hat{y}}$ are weight matrices to be learned. Consequently, Equation 4 can be rewritten as

$$y_t = \text{GRU}(\text{ctx}_t,\ y_{t-1}) \tag{15}$$

B. Multi-scale Residual Block Cascade

Convolutional neural networks have seen a gradual increase of the number of layers in the past few year, together with remarkable improvements in many computer vision tasks such as object recognition. In spite of superiority of deep architectures in representation learning, training deep neural networks suffers from several challenges including gradient vanishing or exploding. To overcome these difficulties, various solutions have been proposed such as layer-by-layer pre-training [21], different parameter initialization strategies [22], and new optimization methods [23], [24].

He et al. [6] proposed residual blocks to enable training of very deep neural networks and achieved a large success in ImageNet [25] competition. Figure 3 shows the structure of a residual block. The intuition is that instead of attempting to fit a desired underlying mapping function $\mathcal{H}(x)$, it is hypothetically easier to fit a residual mapping function $\mathcal{F}(x) = \mathcal{H}(x) - x$. Therefore, the desired mapping can be rewritten as $\mathcal{H}(x) = \mathcal{F}(x) + x$.

We also adopt residual blocks in our convolutional representation learning part to facilitate training. However, for handwritten Chinese character recognition task, we observe that both low-level visual cues (e.g. small strokes as shown in Figure 1) and high-level structural knowledge are necessary for successful predictions. This is contrary to normal object recognition task such as that in ImageNet [25] competition where highly semantic information is the primary concern. Therefore, we introduce shortcut connections that aggregate activations of "lower" layers with those of "higher" layers. Formally, for activation a_i and a_j that have the same number of channels K but different height and width, the aggregation operation $\odot$ is defined as the union of feature vectors:

$$a_i \odot a_j \doteq \{a_i^m\} \cup \{a_j^n\} \tag{16}$$

$$m = 1, 2, \cdots, H_i W_i, \quad n = 1, 2, \cdots, H_j W_j \tag{17}$$

where H_i, W_i, H_j, W_j are the height and width of the corresponding activations. In this way, information at different scales can be utilized.

Figure 4 illustrates the architecture of the multi-scale residual block cascade. Assume the output activations of each building block i in Figure 4 is named as a_i sequentially, then the learned visual representation $V(x) = a_2 \odot a_5 \odot a_8$ will be fed into our iterative refinement module.

IV. Experiments

In this section, we systematically investigate the effectiveness of the proposed multi-scale residual block cascade and iterative refinement module. Our model is trained on CASIA-HWDB1.0 [26] and CASIA-HWDB1.1 [26] training set and evaluated on ICDAR-2013 [1] dataset.

A. Datasets

CASIA-HWDB1.0 collected 1.6 million handwritten Chinese character samples from 420 persons. Each person wrote 4,037 characters. It contains 3,866 Chinese characters as well as 171 alphanumeric and symbols. Among the 3,866 Chinese characters, 3,740 characters are in the key official character set GB2312-80 level-1 [30] which includes 3,755 characters in total.

CASIA-HWDB1.1 collected 1.2 million handwritten Chinese character samples from 300 persons. Each person wrote 3,755 characters that are from GB2312-80 level-1 set [30].

ICDAR-2013 was the evaluation dataset for ICDAR 2013 Chinese Handwriting Recognition Competition. It contains 3,755 handwritten Chinese characters collected from 60 persons who did not contribute to CASIA-HWDB datasets. The total number of samples is 224,419.

B. Implementation Details

The architecture of our multi-scale residual block cascade is shown in Figure 4. Each convolution layer has a kernel size of 3×3 and a stride size of 1×1, followed by a Batch Normalization [18] layer and a Rectify Linear Unit (ReLU). Max-pooling size is 2×2. The proposed iterative refinement module is implemented with an attention-based GRU whose hidden size is 512. The maximum iteration T is set to 4 empirically. Larger T did not show any significant performance gain so we set $T = 4$ to save computation cost. The proposed model is trained in an end-to-end fashion using stochastic gradient decent with a momentum of 0.9. The learning rate is initially set to 0.001 and decreases by half for every epoch.

TABLE I

ACCURACIES FOR DIFFERENT METHODS FOR HANDWRITTEN CHINESE CHARACTER RECOGNITION FOR THE ICDAR-2013 DATASET. ALL ARE TRAINED ON THE CASIA-HWDB1.0 AND CASIA-HWDB1.1 DATASETS. "PREPROCESSING" INDICATES WHETHER RAW IMAGES OR MANUALLY PREPROCESSED SAMPLES ARE USED AS INPUT. IF ENSEMBLE LEARNING STRATEGY IS USED, THE NUMBER OF MODELS IN THE ENSEMBLE IS LISTED IN THE PARENTHESES. NOTE THAT [7] RELIES ON A TIME-CONSUMING PREPROCESSING STEP (SEE SECTION IV-E).

Method	Accuracy (%)	Training data	Preprocessing	Ensemble
Human Performance [1]	96.13	-	-	-
DFE-DLQDF [27]	92.72	1.0 + 1.1	No	No
HKU [28]	89.99	1.0 + 1.1	No	No
Gabor+HCCR-GoogLeNet [29]	96.35	1.0 + 1.1	No	No
HCCR-Ensemble-GoogLeNet(4) [29]	96.64	1.0 + 1.1	No	Yes(4)
HCCR-Ensemble-GoogLeNet(10) [29]	96.74	1.0 + 1.1	No	Yes(10)
Our ConvNet	95.97	1.0 + 1.1	No	No
Our ResNet	96.34	1.0 + 1.1	No	No
Our M-RBC	96.84	1.0 + 1.1	No	No
Our ConvNet + IR	96.64	1.0 + 1.1	No	No
Our ResNet + IR	97.04	1.0 + 1.1	No	No
Our M-RBC + IR	**97.37**	1.0 + 1.1	No	No
DirectMap + ConvNet [7]	96.95	1.0 + 1.1	Yes	No
DirectMap + ConvNet + Adaptation [7]	**97.37**	1.0 + 1.1	Yes	No
DirectMap + ConvNet + Ensemble [7]	97.07	1.0 + 1.1	Yes	Yes(2)
DirectMap + ConvNet + Ensemble [7]	97.12	1.0 + 1.1	Yes	Yes(3)

C. Effectiveness of the Multi-scale Residual Block Cascade

In this section, we compare the proposed multi-scale residual block cascade with a vanilla CNN similar to that in Zhang et al. [7] and a regular residual network. The vanilla CNN contains 12 convolutional layers. A fully connected layer is stacked on top the convolutional part to output final predictions. The regular residual network shares same architecture as our multi-scale residual block cascade as shown in Figure 4, except that the shortcut connections are removed. For both the regular residual network and our multi-scale residual block cascade, a fully connected layer is further applied for classification purpose. Note that despite their differences in architectures, all these three models contain 12 convolutional layers and 1 fully connected layer.

From Table I we can see that the vanilla CNN (denoted as Our ConvNet) and the regular residual network (denoted as Our ResNet) achieves an accuracy of 95.97% and 96.34% respectively, while the proposed multi-scale residual block cascade (denoted as Our M-RBC) achieves a highest accuracy of 96.84%. We can conclude that adding residual blocks leads to a more effective model learning. The proposed multi-scale residual block cascade yields slightly better results than the regular residual network, suggesting the advantages of utilizing low level features for HCCR task.

D. Effectiveness of the Iterative Refinement Module

From Table I we can see that, adding an iterative refinement module leads to performance boosts of 0.67%, 0.70% and 0.53% in accuracy for the vanilla CNN, the regular residual network and the proposed multi-scale residual block cascade, respectively. Taking into consideration the large size of the test dataset, such level of boosts means that we can correctly recognize 1503, 1570 and 1189 more test images, respectively. Since we did not find any publicly available pre-trained models for previous methods, it is difficult to investigate how well those models can be improved by our iterative refinement module. However, the results above suggest that the iterative refinement module can provide extra performance gain after being incorporated into an existing CNN-based model.

Figure 5 shows several examples where the proposed iterative refinement module repeatedly updates predictions and finally yields correct class labels. As we can see, although the model makes wrong predictions at an initial attempt, it is able to correct itself. While the attentions at each time-step are not shown due to space constrain, we do observe that the attentions are shifted to sub-regions that can are more informative to differentiate groundtruth characters from others.

E. Comparisons with State-of-the-art Methods

In this section, we compare our approach with other state-of-the-art methods. As shown in Table I, Our M-RBC method outperforms all other methods that use raw images as input, showing the superiority of using residual blocks with multi-scale shortcuts. After incorporating the iterative refinement module, both Our ResNet+IR and Our M-RBC+IR can achieve higher classification accuracies compared with other methods that do not rely on preprocessing. The DirectMap+ConvNet+Adaptation method ties Our M-RBC+IR method and also achieves an accuracy of 97.37%. However, their method requires shape normalization [31] and direction decomposition [32] to obtain a 8-channel DirectMap during testing. Such kinds of pre-processing steps are very time-consuming. For example, in [7] it takes 1.997 ms to calculate DirectMap and 0.464 ms to perform a forward pass of a deep CNN for each image. In total it takes 2.461 ms. On the contrary, Our M-RBC+IR directly processes raw images and takes 0.623 ms for each image[1], four times faster than the DirectMap+ConvNet+Adaptation method.

[1]Using a Nvidia K40 GPU.

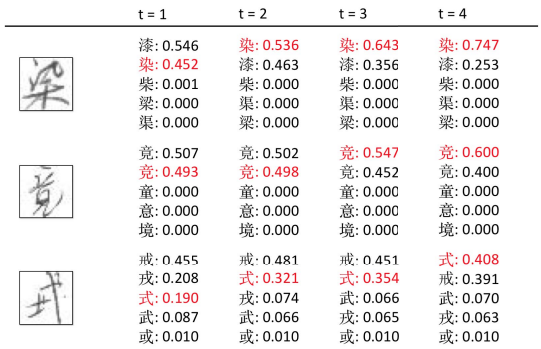

Fig. 5. Examples where the iterative refinement module repeatedly updates its predictions and finally outputs the correct characters. At each time-step t, the top 5 predictions are listed. Each character is followed by the corresponding probability. Characters in red are the correct ones.

V. Conclusions

We have created an iterative refinement module that increases the classification performance for deep convolutional neural networks for handwritten Chinese character recognition (HCCR) task. The iterative refinement module is implemented with an attention-based recurrent neural network, that iteratively uses its previous prediction to update attention and to refine current predictions. A multi-scale residual block cascade that utilizes both low-level visual cues and high-level structural information was specifically designed for HCCR task. Together with the iterative refinement module, we achieve state-of-the-art results on ICDAR-2013 dataset. The proposed model is completely end-to-end, avoiding any pre-processing or post-processing steps. A future direction would be to investigate the effectiveness of the iterative refinement module for other vision tasks such as object recognition.

Acknowledgments

We gratefully acknowledge partial support from NSF grant CCF 1317560 and a hardware grant from NVIDIA.

References

[1] F. Yin, Q.-F. Wang, X.-Y. Zhang, and C.-L. Liu, "Icdar 2013 chinese handwriting recognition competition," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 1464–1470.

[2] N. Jiang, *Advances in Chinese as a second language: Acquisition and processing*. Cambridge Scholars Publishing, 2014.

[3] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network, 1989," in *Neural Information Processing Systems (NIPS)*.

[4] S. El Hihi and Y. Bengio, "Hierarchical recurrent neural networks for long-term dependencies." in *Neural Information Processing Systems (NIPS)*, vol. 409, 1995.

[5] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[7] X.-Y. Zhang, Y. Bengio, and C.-L. Liu, "Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark," *Pattern Recognition*, vol. 61, pp. 348–360, 2017.

[8] W.-T. Chen and T.-R. Chou, "A hierarchical deformation model for online cursive script recognition," *Pattern Recognition*, vol. 27, no. 2, pp. 205–219, 1994.

[9] M. Kobayashi, S. Masaki, O. Miyamoto, Y. Nakagawa, Y. Komiya, and T. Matsumoto, "Rav (reparameterized angle variations) algorithm for online handwriting recognition," *IJDAR*, vol. 3, no. 3, pp. 181–191, 2001.

[10] Y. Liu and J. Tai, "A structural approach to online chinese character recognition," in *Pattern Recognition, 1988., 9th International Conference on*. IEEE, 1988, pp. 808–810.

[11] H. A. Rowley, M. Goyal, and J. Bennett, "The effect of large training set sizes on online japanese kanji and english cursive recognizers," in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*. IEEE, 2002.

[12] A. Graves, M. Liwicki, H. Bunke, J. Schmidhuber, and S. Fernández, "Unconstrained on-line handwriting recognition with recurrent neural networks," in *NIPS*, 2008, pp. 577–584.

[13] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *CVPR*. IEEE, 2012, pp. 3642–3649.

[14] B. Graham, "Spatially-sparse convolutional neural networks," *arXiv preprint arXiv:1409.6070*, 2014.

[15] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[16] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," in *EMNLP*, 2016, pp. 1317–1327.

[17] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention." in *ICML*, vol. 14, 2015, pp. 77–81.

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of The 32nd International Conference on Machine Learning*, 2015, pp. 448–456.

[19] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[20] K. Cho, B. v. M. C. Gulcehre, D. Bahdanau, F. B. H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," 2014.

[21] J. Schmidhuber, "Learning complex, extended sequences using the principle of history compression," *Neural Computation*, vol. 4, no. 2, pp. 234–242, 1992.

[22] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks." in *AISTATS*, 2010.

[23] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning." *ICML (3)*, vol. 28, pp. 1139–1147, 2013.

[24] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[26] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, "Casia online and offline chinese handwriting databases," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 37–41.

[27] C.-L. Liu, F. Yin, and D.-H. Wang, "Online and offline handwritten chinese character recognition: benchmarking on new databases," in *Pattern Recognition*, 2013.

[28] C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang *et al.*, "Chinese handwriting recognition contest 2010," 2010.

[29] Z. Zhong, L. Jin, and Z. Xie, "High performance offline handwritten chinese character recognition using googlenet and directional feature maps," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, 2015, pp. 846–850.

[30] https://en.wikipedia.org/wiki/GB_2312.

[31] C.-L. Liu and K. Marukawa, "Pseudo two-dimensional shape normalization methods for handwritten chinese character recognition," *Pattern Recognition*, vol. 38, no. 12, pp. 2242–2255, 2005.

[32] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: benchmarking of state-of-the-art techniques," *Pattern recognition*, vol. 36, no. 10, pp. 2271–2285, 2003.