Low Resolution Handwritten Digit String Recognition based on Object Detection Network

Yingjie Xu, Jun Guo
School of Data Science and Engineering
East China Normal University
Shanghai, China
jguo@cc.ecnu.edu.cn

***Abstract*—A novel object detection network is proposed in this paper for low resolution handwritten digit string recognition. It is composed of a convolutional neural network (CNN) and two independent output branches for classification and bounding box regression. The network is designed to effectively extract the features from low resolution images. Non-categorized non-maximum suppression (NMS) and mini-batch fine-tuning (MB-FT) are used to improve accuracy further. The experiments are conducted on a new dataset collected by a tablet and HDSRC 2014 benchmark datasets, and the high metrics are obtained. Furthermore, its prediction speed reaches 65 FPS achieving real-time recognition.**

***Index Terms*—object detection, convolutional neural network, handwritten digit string recognition, low resolution**

I. INTRODUCTION

Handwritten digit string recognition (HDSR) is one of the research hotspots in the area of computer vision. Unlike machine-generated digits, each person has a unique style of writing digits enhancing the difficulty of the task. In infinite ways of writing, a person may not be able to understand his manuscript after a period of time, not to mention letting someone else understand it [1]. HDSR has many industrial applications, for example, recognizing addresses in mail sorting and reading amounts in checks. Although HDSR has been studied for many years, getting a lot of methods and achievements [2]–[5], it does not seem to be easy due to the variability of handwritten digits and characters.

In the past few years, the methods [6]–[9] used to solve the HDSR problem were devoted to segment the strings into pieces and applied some machine learning algorithms to recognize every single digit. Sadri, Suen and Bui [6] proposed a genetic framework using contextual knowledge for segmentation, classifiers using neural network (NN) and support vector machine (SVM). Gattal, Chibani and Hadjadji [7] combined three segmentation methods to obtain isolated digits and employed a recognition module based on SVM classifiers. Chen and Guo [8] aimed at solving a classic segmentation problem on touching digit pairs by using spectral clustering (SC). Meanwhile, SVM is used to predict the affinity matrix of SC. Shao, Chen and Guo [9] designed an online HDSR algorithm using spectral clustering which can segment digit strings with the extraction of stroke information.

Thanks to the development of deep learning, more and more methods [10]–[12] had emerged. Goodfellow, Bulatov, Ibarz, Arnoud and Shet [10] presented an approach that can recognize multi-digit numbers from Street View Imagery based on deep CNNs. It is more suitable for door number or license plate number. Tian, Huang, He and Qiao [11] proposed a connectionist text proposal network (CTPN) that locates text lines in natural images by employing a vertical anchor mechanism. It can only detect but not recognize. Shi, Bai and Yao [12] proposed a novel neural network architecture called convolutional recurrent neural network (CRNN) for scene text recognition. It is an end-to-end system that depends on a blank mechanism to handle duplicate characters and the lexicon-based transcription layer. It is failed on HDSR since there are lots of connections between characters and no lexicon in existence.

In this paper, we propose a novel network based on object detection. There are two main paradigms. One is the two-stage approach represented by Faster R-CNN [13] which narrows down the number of candidate object locations to a small quantity in the first stage and goes to a further classification in the second stage. Another is the one-stage approach represented by YOLOv3 [14] which needs to process a large number of candidate object locations across an image. This approach gets faster processing time but losing accuracy because of the extreme class imbalance. If we use the above intuitive approaches to solve the HDSR problem, only magnifying the image can help us to complete the task but leading to more time consumption. We find that there is no well-designed object detector for low resolution HDSR problem. So we propose a solution.

As mentioned above, our proposed model is based on the one-stage approach as shown in Fig. 1. First, a revised ResNet as the backbone is used to generate a feature map. Then, it is fed into two subnetworks respectively. Each subnetwork is a small fully convolutional network (FCN) [15]. They are responsible for classification and bounding box regression. In addition, we employ the focal loss [16] on the output of the classification subnet to solve the class imbalance problem. During the post-processing stage, we use non-maximum suppression (NMS) without discriminating class. Finally, a training heuristics with smaller batch-size to fine-tune our model called mini-batch fine-tuning (MB-FT). These two heuristics totally increase the accuracy by about 3% to 7%.

In the experimental stage, we use a new dataset called hand-

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

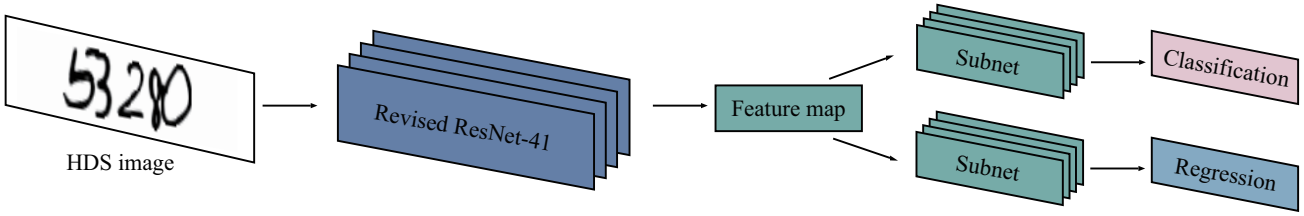


Fig. 1. The architecture of the proposed model. It consists of a backbone network and two parallel output branches.

written digit strings five (HDS5) to evaluate our model which includes synthetic digit strings and a large number of handwritten digit strings from tablets. By using non-categorized NMS and MB-FT, our model gets the highest accuracy compared with the mainstream object detection models. Subsequently, we extend the proposed model on benchmark HDSRC2014 and get encouraging experimental results by training on the enhanced datasets.

II. METHODOLOGY

We will elaborate on the proposed method in this section. The overview of the architecture is described in Section II-A. Then, we will show the backbone network for feature extraction in detail in Section II-B. The subnetworks and the loss function are introduced in Section II-C. At the end of this section, it briefly describes non-categorized NMS and MB-FT in Section II-D and Section II-E.

A. Overview

The architecture of our network is shown in Fig. 1. It is a one-stage object detection network that includes a backbone network and two subnetworks. The backbone network originates from ResNet-50 [17] and goes through some modifications called revised ResNet-41. The subnetworks are the RetinaNet's [16] recognizers and detectors.

The pipeline of our network is as follows. To begin with, a handwritten digit string image converted to the grey-scale is fed into a revised ResNet-41 CNN model to generated a feature map. After that, the feature map passes through two parallel subnetworks which are small FCNs. One performs convolutional object classification, the other performs convolutional bounding box regression. Obviously, the proposed network is an end-to-end model. After mini-batch fine-tuning, we obtain the final experimental results.

B. Backbone Network

Since our network is dedicated to solving the HDSR problem in low resolution, the key point is the backbone network to generate a rich semantic feature map. Imagining that if the feature maps created by RetinaNet [16] which uses feature pyramid network (FPN) [18] without any modifications. We can get a multi-scale feature pyramid with feeding a 160×32 size image, the largest size of feature maps is 20×4. The experimental result shows that the final accuracy is only 76.8% on HDS5. From here we can see that the feature pyramid can not provide rich semantic information for the next network to achieve classification and regression. Therefore, the backbone network for the low resolution HDSR problem needs to be specially designed.

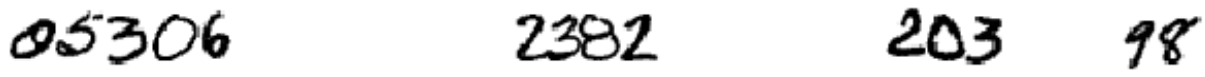
TABLE I
CONFIGURATION OF THE REVISED RESNET-41

Layer	Input	Configuration	Output
Conv1	160×32	f64, k7 $\times$ 7, s2 $\times$ 2	80×16
MaxPooling	80×16	k3 $\times$ 3, s2 $\times$ 2	40×8
Conv2_x	40×8	$\begin{bmatrix} \text{f64, k1} \times 1 \\ \text{f64, k3} \times 3 \\ \text{f256, k1} \times 1 \end{bmatrix} \times 3$, s1 $\times$ 1	40×8
Conv3_x	40×8	$\begin{bmatrix} \text{f128, k1} \times 1 \\ \text{f128, k3} \times 3 \\ \text{f512, k1} \times 1 \end{bmatrix} \times 4$, s1 $\times$ 1	40×8
Conv4_x	40×8	$\begin{bmatrix} \text{f256, k1} \times 1 \\ \text{f256, k3} \times 3 \\ \text{f1024, k1} \times 1 \end{bmatrix} \times 6$, s1 $\times$ 1	40×8

'f', 'k', 's' stand for filter, kernel and stride respectively.

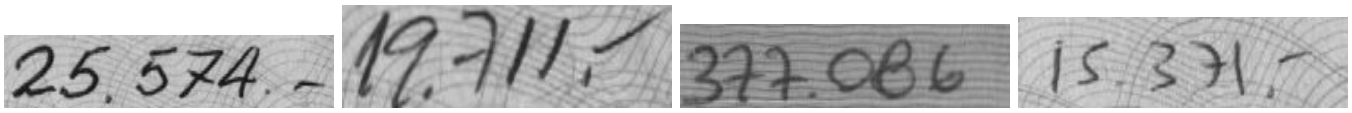
FPN [18] enhances the standard convolutional network by a top-down pathway and lateral connections which effectively constructs a rich semantic feature pyramid from a single image. In our opinion, its design aims at discriminating different objects at a different scale by merging multi-scale feature maps. In this paper, we focus on handwritten digit objects which are almost identical in scale. We boldly design a backbone network with only one feature map instead of the feature pyramid. So we propose the revised ResNet-41 which originates from ResNet-50 to replace FPN.

We still use the convolutional block and identity block in ResNet [17] to extract more features with a deeper network and avoid gradient exploding problem. There are two modifications for ResNet-50. Considering the size of the feature map is too small, we try to turn off some down-sampling layers. Keeping the output size of the ResNet in 40×8. Besides, on account of the ResNet-50 is very time-consuming, we intend to remove some stages in ResNet-50 to speed up. After some relevant experiments shown in Section III-C, we know that stage 5 can be removed without affecting the recognition

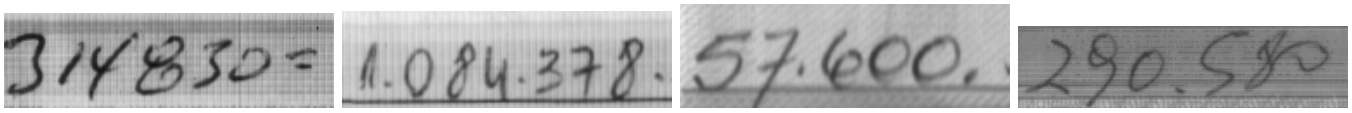


(a) Strings from HDS5

(b) Strings from CVL HDS



(c) Strings from ORAND-CAR-A



(d) Strings from ORAND-CAR-B

Fig. 2. Samples from HDS5 and HDSRC 2014.

accuracy. The configuration of the revised ResNet-41 is shown in Table I.

C. Subnetworks and Loss Function

The subnetworks are responsible for recognition and detection which are composed of two parallel small FCNs. The architecture and parameter setting of subnetworks are identical to the RetinaNet [16]. Unlike RetinaNet, we only use two anchors. The size of a single digit is similar to a square or a rectangle so that we design the aspect ratios to 21×24 and 12×24 by using the K-means algorithm to count.

Since the key problem in the one-stage approach is an extreme class imbalance between the foreground and background, we use focal loss proposed on RetinaNet [16] for classification. In addition, The standard smooth $\mathcal{L}_1$ loss is used for regressing the bounding box. The sum of the above two loss functions is our training loss.

D. NMS

In the field of object detection, a post-processing approach called NMS [19] is an important step to eliminate redundant bounding boxes and find the best one for the objects to be detected. It will divide bounding boxes into several groups by their classes. After sorting by scores (as in Faster R-CNN [13]) or confidence (as in YOLOv3 [14]) within the group, all bounding boxes calculate Intersection-over-Union (IoU) with the maximum score box and whose IoU larger than a threshold will be deleted. This process is intended to detect the objects belong to different classes which are overlapped seriously.

But in our HDSR problem, there is a rare existence of serious overlapping between two adjacent digits. Only a few connections and touch will happen. We conduct NMS on all classes called non-categorized NMS. By using these heuristics, the accuracy will be significantly improved.

E. Training with Mini-batch Fine-tuning

Mini-batch gradient descent (MBGD) is commonly used for training, which is equivalent to the compromise between the batch gradient descent (BGD) and stochastic gradient descent (SGD): dividing the training set into several mini-batches, training one mini-batch iteratively and updating the weights according to the loss of the batch data.

Reference [20] shown that increasing batch-size is helpful to the stability of convergence in a certain range of epochs, but with the increase of batch-size, the performance of the model will decline. The reason is that large batch-size converges to a sharp minimum, while small batch-size converges to a flat minimum which has better generalization ability. So we start training with a large batch-size of 512 to converge steadily to an optimal point. Then, using a small batch-size of 16 and initializing with previous network weight to fine-tune our model called mini-batch fine-tuning (MB-FT). This training heuristics strengthens the generalization ability of the model. The model performs better on the test sets and improves our accuracy by about 2%.

III. EXPERIMENTS

Our experiments are run on an NVIDIA Tesla P100 GPU to evaluate the proposed model. The criterion of accuracy is used as the main comparison measure which means no matter the predicted digits are error, even less or more than the ground-truth labels, it will be determined to a wrong recognition. Furthermore, mean average precision (mAP) is used for comparison as it is prevalent in object detection.

A. Datasets

There are two datasets called HDS5 and HDSRC 2014 [21]. HDS5 shown in Fig. 2.(a) is an abundant dataset which is collected for us to research HDSR, especially in low resolution. HDSRC 2014 as a benchmark dataset, includes CVL HDS and ORAND-CAR-A & B shown in Fig. 2.(b),

TABLE II
THE DISTRIBUTION OF DATASETS

Dataset	Training set	Test set
HDS5	200000	8000
CVL HDS	1262	6698
CAR-A	2009	3784
CAR-B	3000	2926

(c) and (d), helps us prove the validity of our model better. CVL HDS is collected from more than 300 different writers and ORAND-CAR is collected from the real bank checks. The size of the images in HDSRC 2014 is not exactly the same. Besides, it does not offer ground-truth but only has sequence labels. Thus, it cost us much time to relabel the training sets of the dataset with the help of an annotation tool called LabelImg. The distribution of the two datasets is shown in Table II.

It deserves to be mentioned that the HDS5 dataset has significance for the research on HDSR. It makes up of two parts. One is $200,000$ synthetic digit strings by using MNIST database called S_{mnist}. We connect single digits to a digit string which has 2 to 5 digits. The other is a large number of handwritten digit strings whose lengths are also from 2 to 5 by using the tablets with the help of college students and employees. In this part, $48,125$ digit strings called R_{hsd5} are real handwritten strings and $351,875$ digit stings synthesized from different single digits by handwriting called S_{hds5}. All synthetic operations control a random degree of connection between the two digits. Both of the two parts generate annotations automatically.

From Fig. 2.(a), we can see that all images are fixed to the same size in exactly 160×32 px when filling in the blanks with white. To increase the diversity of our samples, the handwritten digit strings are placed anywhere of the images.

We randomly select $60,000$ samples from S_{mnist}, $40,000$ samples from R_{hsd5} and $100,000$ samples from S_{hds5} which have a total of $200,000$ images as the training set. We believe that such rich samples are sufficient to train the model well. Another step is selecting $8,000$ samples randomly as the test set. The mentioned above is the composition of our HDS5 dataset.

B. Data Augment

It is foreseeable that HDSRC 2014 does not have as many samples as HDS5 for us to train. Because of the lack of sample diversity, we decide to use data augment. The transformation methods are divided into two groups. One includes the stretching of the height or width of a digit and the scaling of it, the other includes rotation, piecewise affine and perspective transformation to an image. The transformed image is obtained by choosing a transformation method from any group or combining two transformations from each group randomly at a time. We get 15 styles of transformations from one image. After five loops with random parameters, the training sets have been extended to $95,912$ for CVL HDS, $152,684$ for CAR-A and $228,000$ for CAR-B. With the help of a Python package named 'imgaug', the transformed ground-truth can be obtained simultaneously after a transformation.

C. Ablation

We tried a lot of experiments to determine the design of our backbone network. Training on HDS5 using RetinaNet without any modification as our baseline model. It adopts ResNet-50 and FPN as the backbone network which generates five different scales of feature maps. The anchors are set to three different sizes, each with three aspect ratios. The images are fixed to 256 px on the short side by keeping the aspect ratio and fed into the network.

To design our backbone, we remove the FPN and the reason is shown in Section II-B. First, we consider whether to use ResNet-50 or VGG-16 as the backbone. Model 1 and Model 2 are designed to make a wise decision. The last three layers of down-sampling are closed down so that the output is a 40×8 feature map. The aspect ratios of anchors are set to 21×24 and 12×24. From the Table III, we can see that Model 1 is better. Although the testing time of model 2 with VGG-16 is faster than Model 1, the accuracy as the first criterion declines by about 2% so that we choose ResNet as our backbone.

In addition, considering the little complexity of the HDSR problem, further experiments are carried out to verify how many stages can we removed to speed up our model. We remove the last three stages of ResNet-50 in Model 3 and other settings are the same as Model 1. Unfortunately, the results are far from our baseline model which is only 91.85% accuracy. Then, Model 4 and Model 5 remove the last two stages and the last one. According to the results in Table III, we decide to use ResNet-41, as it keeps the same accuracy with ResNet-50 but reduces the detection time. This proves that excessive convolutional layers are unnecessary to make the model performs better.

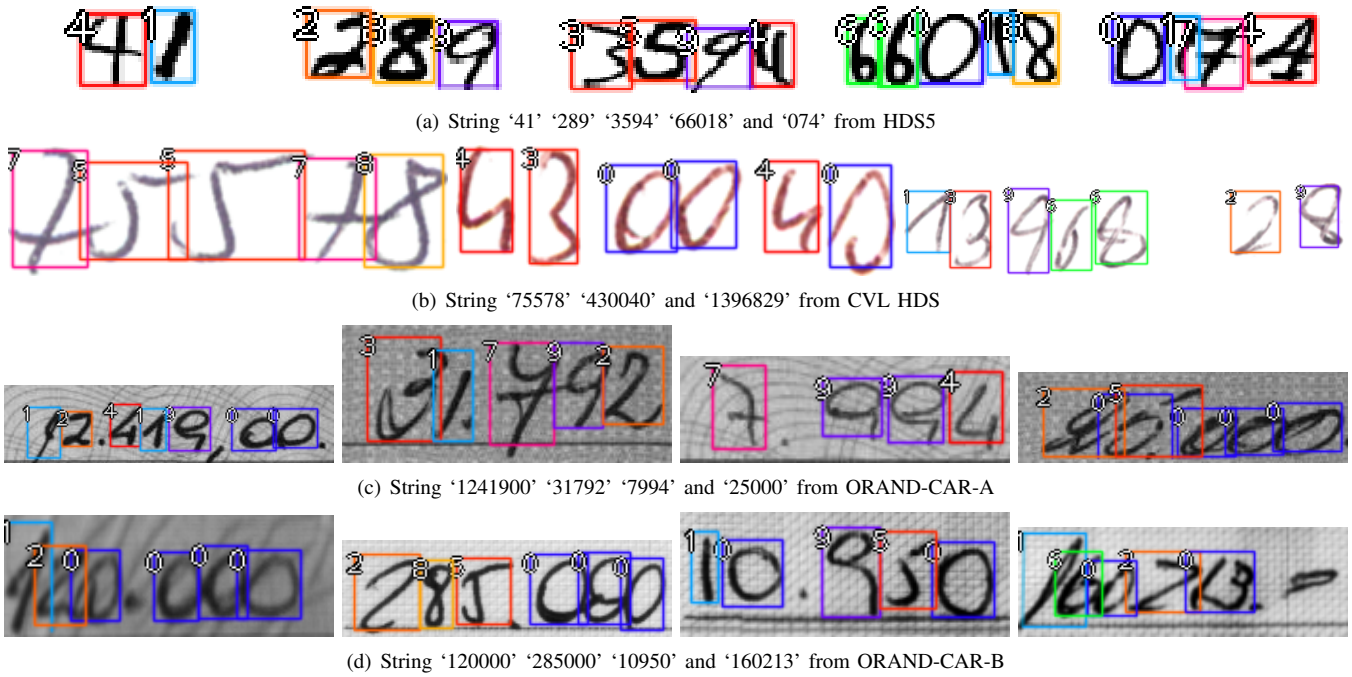
D. Results and Analysis

a) Experiments for HDS5: After determining the network structure, we use a training heuristics named MB-FT to fine-tune our model. The results are compared with the mainstream object detection methods which are Faster R-CNN for two-stage and YOLOv3-tiny for one-stage, non-categorized NMS is also employed. Similarly, in order to obtain satisfactory results in low resolution, it is necessary to revise the down-sampling of the two models to get rich semantic feature maps. Finally, The down-sampling ratio is

TABLE III
ABLATION EXPERIMENTS FOR BACKBONE DESIGN

Model	Backbone	Accuracy	mAP	FPS
RetinaNet	ResNet-50+FPN	96.43	**99.93**	21
Model 1	ResNet-50	96.65	99.43	44
Model 2	VGG-16	94.62	99.19	**120**
Model 3	ResNet-11	91.85	98.49	115
Model 4	ResNet-23	94.19	98.97	96
Model 5(ours)	ResNet-41	**96.66**	99.48	65

The metrics of Accuracy and mAP is %.



(a) String ‘41’ ‘289’ ‘3594’ ‘66018’ and ‘074’ from HDS5

(b) String ‘75578’ ‘430040’ and ‘1396829’ from CVL HDS

(c) String ‘1241900’ ‘31792’ ‘7994’ and ‘25000’ from ORAND-CAR-A

(d) String ‘120000’ ‘285000’ ‘10950’ and ‘160213’ from ORAND-CAR-B

Fig. 3. Results from HDS5 and HDSRC 2014. This is a demonstration of the correct predictions except the last one of each row is a misclassified sample.

TABLE IV
THE ACCURACIES (%), MAPS (%) AND FPS ON HDS5

Model	Accuracy	mAP	FPS
Faster R-CNN	94.75	99.21	5
YOLOv3-tiny	95.57	99.23	**257**
Ours+MB-FT	**97.30**	**99.57**	65

set to $1/2$ for Faster R-CNN and get an 80×16 feature map. However, since YOLOv3-tiny applies the residual mechanism whose prediction depends on two feature maps, the down-sampling ratios are set to $1/4$ and $1/8$.

We train the model about 200 epochs and used MB-FT within 50 epochs. From the Table IV, we can see that our model has already exceeded the other two models. Although YOLOv3-tiny has the fastest speed, our model performs better than it in accuracy and has impressive detection speed as well. Meanwhile, it is encouraging that the accuracy has exceeded the baseline model obviously.

In HDS5, most errors occurred when one part of a digit is recognized as another, just like the part of the digit ‘7’ in the last image of Fig. 3.(a) is recognized as ‘1’. We are afraid that this kind of mistake will also happen in human eye recognition.

b) Experiments for HDSRC 2014: To prove the robustness of our model, we train our model on the benchmark HDSRC 2014. The first problem we met is that the size of images in HDSRC 2014 is quite different, which makes it impossible for us to train by MBGD. So we resize the height of images to 32 px while maintaining the aspect ratio of each image. Besides, statistics show that the widths of the images do not exceed 256 px after resizing the height of the images. Finally, all images are padded to the size of 256×32 with white.

Another problem we need to overcome is that the model obtains high accuracy in training sets but failed in test sets when we train with the original training sets. In our opinion, the reason is the lack of training samples. So data augment is utilized on the HDSRC 2014 to enhance the diversity of our training sets. Furthermore, transfer learning can be employed since it is the same HDSR problem as solving the recognition of HDS5. Thanks to the weights of HDS5 to initialize the model, we get the best model within a few epochs.

After finishing off these two problems, we train the model with MB-FT using the training sets without data augment and the best results are shown in Table V. The results from Tébessa I to Shanghai are presented in ICFHR 2014, Saabni to RNN+CTC come from recent papers. Comparative experiments are also done on Faster R-CNN and YOLOv3-tiny shown in the next two rows.

Recent state-of-the-art algorithms mostly consider HDSR as a segmentation problem or a sequence prediction problem, but they all have some shortcomings. The experiment from Table V demonstrates that the method Beijing [21] based on traditional segmentation performs the best in CVL HDS because of the pure background of images. However, when it comes to the complicated background like the images of CAR-A & B datasets, it underperforms our model.

For methods based on sequence, they need large and diverse samples. Once the training samples are monotonous, it will fail. For example, RNN-CTC [23] gets the best accuracies of 89.75% and 91.14% on CAR-A & B but it gets into trouble

TABLE V
THE ACCURACIES (%) ON HDSRC 2014

Model	CVL HDS	CAR-A	CAR-B
Tébessa I [21]	59.30	37.05	26.62
Tébessa II [21]	61.23	39.72	27.72
Singapore [21]	50.40	52.30	59.30
Pernambuco [21]	58.60	78.30	75.43
Beijing [21]	**85.29**	80.73	70.13
Shanghai [21]	48.93	49.50	28.09
Saabni [22]	-	85.80	
CRNN [12]	26.01	88.01	89.79
RNN+CTC [23]	27.07	**89.75**	**91.14**
Faster R-CNN	70.11	75.38	76.81
YOLOv3-tiny	65.86	72.51	76.13
Our model	78.67	83.17	84.79

in CVL HDS due to the lack of sample diversity. There are 300 writers that contribute to this dataset. For each writer, 26 different digit strings were collected. Only 10 kinds of strings occur in the training set. But for object detection methods, this is not a problem because they treat digits as different objects. In this framework, the digit strings are separated into some single digits so that the complex string recognition is reduced to a limited category classification problem. Under data augment and transfer learning, our accuracy has greatly exceeded it on CVL HDS and reaches 78.67%.

The predicted results of HDSRC 2014 are shown in Fig. 3. It can be found that our model can recognize very well whether the backgrounds of the images have lots of noise or the digital color is changeable. From the wrong image in Fig. 3.(b), both of digits '8' are misclassified due to the lack of training samples. One of the main problems of ORAND-CAR-A & B is the lack of samples, and the other is that, like HDS5, one digit is considered to be different digits or two digits are regarded as one. For example, the '0' of the wrong predicted image from Fig. 3.(c) is a part of the digit '5'. The digits '1' and '3' are recognized as one digit '0' in Fig. 3.(d).

By the way, Faster R-CNN is superior to YOLOv3-tiny on three benchmark datasets but its accuracy is still lower than our model. Although our model can not reach the state-of-the-art on all three datasets simultaneously, the method based on object detection is a novel train of thought on solving the HDSR problem. As long as we have enough labeled datasets, this method is worthy of further study.

IV. CONCLUSIONS

In this paper, a model is proposed to solve the HDSR problem in low resolution based on object detection. It mainly contributes a revised ResNet-41 as the backbone network to extract features. Meanwhile, with the help of two heuristics, we achieve excellent accuracy and reduce the recognition time compared with traditional object detection methods. Although our model does not achieve state-of-the-art in HDSRC2014, it is unlike CRNN and RNN-CTC based on sequence labeling which will perform badly once lacking the diversity of samples. Our method can avoid this problem fairly because of the existence of bounding boxes.

REFERENCES

[1] A. Bischoff and P. S. P. Wang, "Handwritten digit recognition using neural networks," *Spie Intelligent Robots & Computer Vision X*, vol. 1608, pp. 460 – 464, 1992.

[2] O. Matan, C. Burges, Y. Lecun, and J. Denker, "Multidigit recognition using a space displacement neural network," in *International Conference on Neural Information Processing Systems*, 1992, pp. 488–495.

[3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] U. Pal, A. BelaïD, and C. Choisy, "Touching numeral segmentation using water reservoir concept," *Pattern Recognition Letters*, vol. 24, no. 1-3, pp. 261–272, 2003.

[5] L. S. Oliveira and R. Sabourin, "Support vector machines for handwritten numerical string," in *International Workshop on Frontiers in Handwriting Recognition*, 2004, pp. 39–44.

[6] J. Sadri, C. Y. Suen, and T. D. Bui, "A genetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings," *Pattern Recognition*, vol. 40, no. 3, pp. 898–919, 2007.

[7] A. Gattal, Y. Chibani, and B. Hadjadji, "Segmentation and recognition system for unknown-length handwritten digit strings," *Pattern Analysis and Applications*, vol. 20, no. 2, pp. 307–323, 2017.

[8] C. Chen and J. Guo, "A general approach for handwritten digits segmentation using spectral clustering," in *IAPR International Conference on Document Analysis & Recognition*, 2017, pp. 547–552.

[9] R. Shao, C. Chen, and J. Guo, "An online handwritten numerals segmentation algorithm based on spectral clustering," in *International Conference on Neural Information Processing*, 2018, pp. 506–516.

[10] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," *Computer Science*, 2014.

[11] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *European Conference on Computer Vison*, 2016, pp. 56–72.

[12] 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 Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks." *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[14] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2018, pp. 1–6.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision & Pattern Recognition (CVPR)*, 2015.

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2017, pp. 2999–3007.

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

[18] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2017.

[19] A. Neubeck and L. J. V. Gool, "Efficient non-maximum suppression," in *International Conference on Pattern Recognition*, 2006, p. 850–855.

[20] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," in *International Conference on Learning Representations*, 2016.

[21] M. Diem, S. Fiel, F. Kleber, R. Sablatnig, J. M. Saavedra, D. Contreras, J. M. Barrios, and L. S. Oliveira, "Competition on handwritten digit string recognition in challenging datasets (hdsrc 2014)," in *International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 779–784.

[22] R. Saabni, "Recognizing handwritten single digits and digit strings using deep architecture of neural networks," in *International Conference on Artificial Intelligence & Pattern Recognition*, 2016, pp. 1–6.

[23] H. Zhan, Q. Wang, and L. Yue, "Handwritten digit string recognition by combination of residual network and rnn-ctc," in *International Conference on Neural Information Processing*, 2017, pp. 583–591.