Research on the Hyperparameter Optimization Method of Mask RCNN Based on DARTs

Gang Hao [1], Ling Cao [2], Peng Liang [1,*]

[1]*School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China*

[2]*Information Engineering School, Guangzhou Vocational College of Technology & Business, Guangzhou 511442, China*

**Corresponding Author.*

Abstract

When dealing with specific target detection tasks, the target detection algorithms based on deep learning are often due to the lack of training samples or expert experience, resulting in the training and verification of the network requires a lot of computing resources, and cannot achieve better detection performance. This paper proposes a hyperparameter optimization method and training method of Mask RCNN Based on DARTs. Experiments show that, this method can optimize the model with only a small number of training samples and without expert experience, and improve the accuracy of the model by 4.83% when IoU is 0.75.

Keywords: *Specific target detection, DARTs, character detection, mask RCNN*

I. Introduction

With the continued development of Internet technology, researchers have found that using the deep learning model of Convolutional Neural Networks (CNN) to perform target detection on large-scale data sets can achieve better recognition results. But for specific target detection tasks, the deep learning algorithm still has some shortcomings [1]:

1) The data set is seriously insufficient. Deep learning algorithms is too dependent on the amount of training data. In order to obtain better target detection accuracy, a massive of data is needed for training, but specific tasks often have the characteristics of small number of training samples, high recognition requirements and so on. For example, in industrial production, a massive of data samples are often hard to obtain, so it is often impossible to construct new massive training data sets;

2) For specific recognition tasks, there are often different shapes of target images under complex background, so it is impossible to meet the industrial requirements by directly using the existing general depth recognition model, but the optimization of network structure and hyperparameters of existing models requires a lot of computing resources and expert experience. Taking the general Mask RCNN [2] (Mask Region-CNN) for mold coded character recognition as an example, the structure involved in the optimization operation includes backbone network, Feature Pyramid Network (FPN), Region Proposal Network (RPN), classification regression branch, Mask branch and so on. If only optimizing a certain layer of classification regression branch network as an example, assuming that the training convergence time of the model is 2GPU DAYs, then without using expert experience, 80GPU DAYs is needed to directly optimize the network structure and hyperparameters to obtain the optimal network structure. Even with the guidance of expert experience, the time required for network structure and hyperparameters optimization is also long, and the computing cost of optimization increases exponentially with the increase of network layers.

In response to the above problems, Neural Architecture Search (NAS) technology has been proposed in recent years, including Evolutionary algorithms Architecture Search [3], Reinforcement learning Architecture Search [4,5]

and Differentiable Architecture Search (DARTs) [6] three methods. These methods provide a fast and intelligent method for network structure and hyperparameter optimization. However, many studies have shown that the NAS-based networks can beat artificially designed networks on some small classical data sets, but they also need to consume a lot of computing resources for the training.

In view of this, this thesis proposes a hyperparameter optimization method and training method of Mask RCNN Based on DARTs. Experiments show that this method can achieve fast and high-precision specific target recognition tasks with only a few training samples and no expert experience.

The remaining organizational structure of the thesis is as follows: the second part is related work, the third part is the description of the proposed method, the fourth part is the introduction of the data set, the fifth part is the experiment and analysis, and the sixth part is the summary and prospect.

II. Related Work

At present, the common method for target detection for specific tasks is to simply modify the existing network model, but in character detection, due to the lack of training samples and expert experience needed to modify the network model, it is hard to achieve the desired results. In order to better classify common foods, Charles et al. introduced the Mask RCNN model to train on the data set constructed by the author, and used expert experience to optimize the parameters of the model, and finally made the model better than the model FCN nearly 8 percentage points when the IOU was 0.7 [7]. Yang Yu et al. proposed the MRSD model, which is based on the pre-training Mask RCNN model based on the COCO data set, and trained on the strawberry fruit image sample set. Through continuous updating of network hyperparameters and loss functions, the accuracy rate finally reached 98.41% in an unstructured environment [8]. Qinghui Zhang et al. proposed a model for detection and segmentation of accidental vehicle damage areas. By reducing the number of layers of ResNet network structure, adjusting the order of various layers of backbone, and updating the parameters and loss function of Anchor Box, the detection accuracy of the new model was improved by 2.15% [9]. All the above algorithms are based on the original model, and use known or rare strategies to improve the existing functional model to achieve better detection results, but for unknown specific tasks, it is difficult for us to have the corresponding expert experience to improve the existing model.

Therefore, in the absence of expert experience, the industry proposed a neural network architecture search technology to realize the automatic search and optimization of neural network architecture for specific tasks.

Neural network architecture search technology is divided into three categories: Evolutionary algorithms Architecture Search, Reinforcement learning Architecture Search and Differentiable Architecture Search. Chen Yukang et al. proposed a method based on evolutionary algorithm to search for the feature extraction network in the target detection network, named DetNAS [10]. In the ImageNet classification task, the result of the algorithm was only 0.03% lower than that of the artificial design network, and the best result at that time was obtained on the COCO target detection data set, and the average accuracy is 1.2% higher than that of the previous optimal model. Ghiasi et al. proposed a method to search for the FPN network in target detection network by using Reinforcement learning Architecture Search technology, named NAS-FPN [11]. By designing a search space that covered all possible cross-scale connections, using RNN as the controller, and searching for the optimal FPN structure through reinforcement learning, the algorithm achieved the highest accuracy on the COCO data set. Sirui Xie et al. proposed a differentiable neural network architecture search method called Stochastic Neural Architecture Search (SNAS) [12]. This algorithm optimized network parameters and architecture hyperparameters in back propagation while keeping the network search differentiable. Although the above techniques can realize the optimization of network structure and hyperparameters, but the consumption of computing resources is also huge if they directly replace the whole general function model.

III. The Hyperparameter Optimization Method of Mask RCNN Based on DARTs

In order to increase the target detection performance of specific tasks under small samples, and without expert experience and a lot of computing resources, we propose a hyperparameter optimization method and training method of Mask RCNN Based on DARTs.

Figure 1 is the flow chart of the whole framework of the mold coded character recognition model based on Mask RCNN. The mold coded image passes through one network of feature extraction formed by Resnet101 and FPN to obtain a feature map, and then a candidate regions are generated through RPN. RoI Align is performed on the candidate regions to obtain 7×7 feature maps. Then we divide the network into upper and lower branches: the upper branch is the classification regression branch for classification of target and detection of bounding box; the lower branch is the Mask branch for target pixel-level segmentation; finally, the two branches are integrated to obtain the recognition and segmentation results of the mold coded characters. Through experimental analysis, it is found that in the target detection of coded character recognition for specific task, the low recognition rate is mainly due to the low detection performance of its classification regression branch (shown in the red box in Figure 1). Therefore, we use DARTs to locally optimize the network structure and hyperparameters of the classification regression branch.

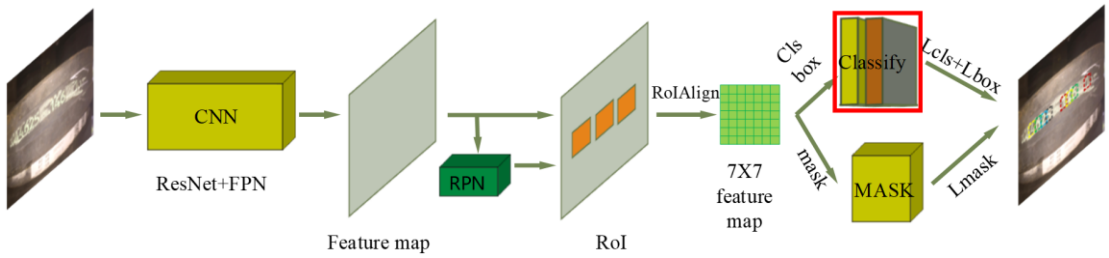


Fig 1: The flow chart of the whole framework of the mold coded character recognition model based on Mask RCNN

Specifically, the proposed method firstly uses DARTs to construct the mold coded character classification model, and uses the mold character data set to train the classification model to get the structure of cell [13]. Then, the cells are stacked to reconstruct the classification regression branch in the Mask RCNN, and the mold coded image data set is used to train the whole Mask RCNN network, so as to realize the automatic optimization of the Mask RCNN network hyperparameters, and finally achieve the fast and high-precision recognition effect of the mold encoded characters.

3.1 The setting of cell structure in network

The structure of cell which used to reconstruct the classification regression branch network is shown in Figure 2. In this structure, *S0*, *S1* are the two nodes for input; *T1*, *T2*, *T3* are the three middle nodes, in which *T1* is connected with *S0* and *S1*, *T2* is connected with *S0*, *S1* and *T1*, *T3* is connected with *S0*, *S1*, *T1* and *T2*; *Cout* is the node for output. The nodes are connected by directed acyclic, in which the line of gray represents an optional set of operation Θ = {3 x 3, 5 x 5 separable convolution, 3 x 3, 5 x 5 empty convolution, 3 x 3 max pooling, 3 x 3 avg pooling, skip connect, zero connection}, the line of black represents the connect operation (Concat), and the output is gained by connecting many middle nodes.

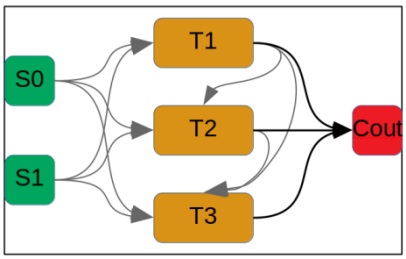


Fig 2: Cell structure diagram

On the basis of the above cell structure, after completing the cell structure search, the mapping of operational connection of the cell nodes is maximized by formula (1):

$$o(i,j) = argmax_{0\in\Theta}(\alpha_o^{(i,j)}) \tag{1}$$

Among them, $o(i,j)$ represents the connect operation of the T_i and T_j nodes, Θ represents the operation set of the $o(i,j)$ operation acting on $T(i)$, $o(\cdot)$ represents multiple operations belonging to the operation set Θ, and $\alpha_o^{(i,j)}$ represents a certain operation connection weight between nodes Ti and Tj .
Figure 3 shows a schematic diagram of the connections between nodes after the maximization operation. The final cell structure is achieved by retaining the two largest operational connections (red lines) and discarding the other operational connections.

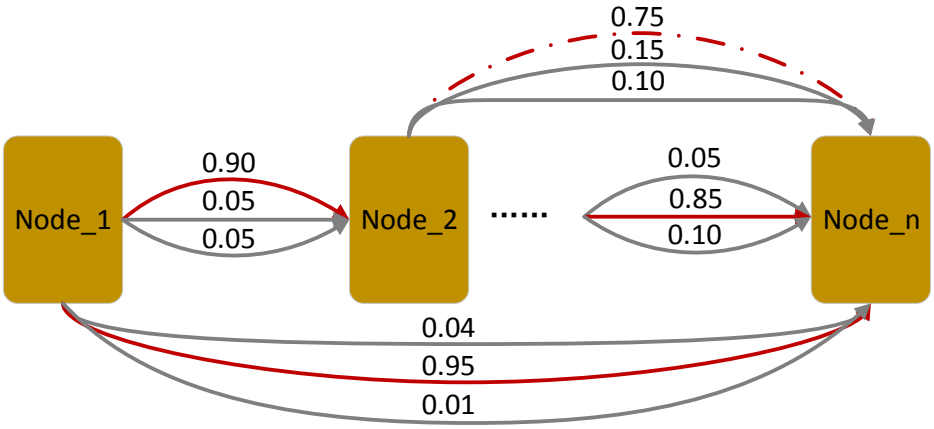


Fig 3: Connections mode of nodes after maximization

3.2 Construction and training of mold coded character classification model

On the basis of the above cell structure setting, we stack multiple cells to get a final network structure [14]. As shown in Figure 4, the input of the network model is a 28 x 28 image, while the *S0* and *S1* of the subsequent cells are the outputs of the first two cells respectively. Among them, (L1, L2, L3, L4, L5, L6, L7) are scale-invariant layers constructed using cells, and the network connects the two layers of completely connected network finally.

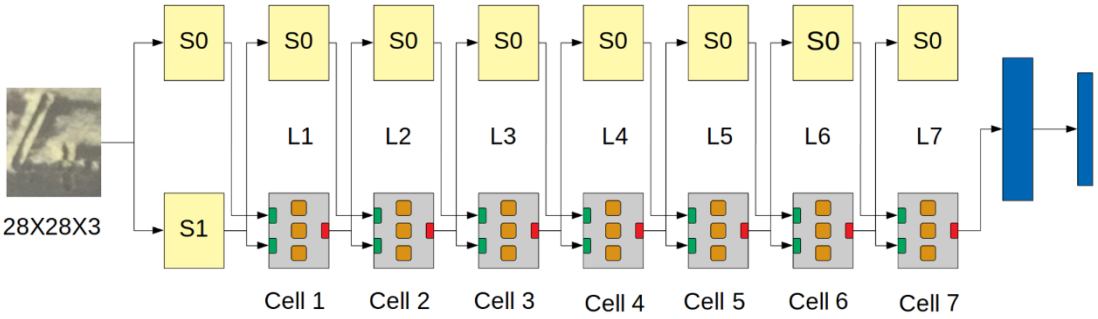


Fig 4: Network structure diagram of DARTs mold coded character classification model

By using character images to train the classification network, the cell structure and parameters with better loss rate can be obtained. The complete training process is as follows:

Init: Initialization, including operation set, cell structure, model structure and data set;
S1: Using gradient descent to update cell network weights w: $\nabla_w \mathcal{L}train(w, \alpha)$;
S2: Using gradient descent to update cell structure weights α: $\nabla_w \mathcal{L}val(w - \xi\nabla train(w, \alpha), \alpha)$;
S3: The model converges, maximizing the weight $\alpha^{(i,j)}$ of network structure for every edge, selecting operation $o^{(i,j)}$, so as to obtain the structure of network. If it does not converge, return to S1.

3.3 Reconstruct classification regression branches by cell stacking

Based on the cells trained in Section 3.2, the classification regression branch in the Mask RCNN network is reconstructed by means of cell stack. Figure 5 shows the classification regression branch network structure formed by stacking two cells. The network uses the feature map calibrated by RoI Align as initial input for *S0* and *S1*, and the final output passes through the two layers of completely connected network.

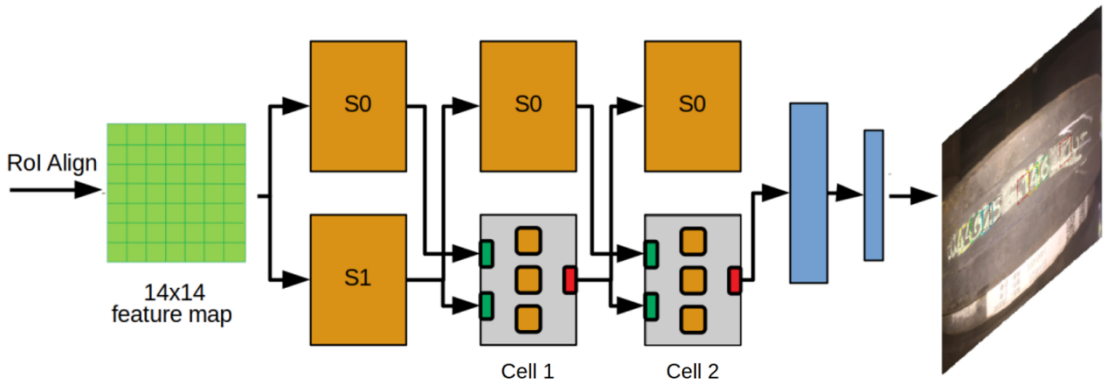


Fig. 5. The flow chart of the character classification regression branch network structure based on DARTs reconstruction in the Mask RCNN model

The RCNN network structure in the Mask RCNN network improved by the DARTs algorithm is shown in Figure 6.

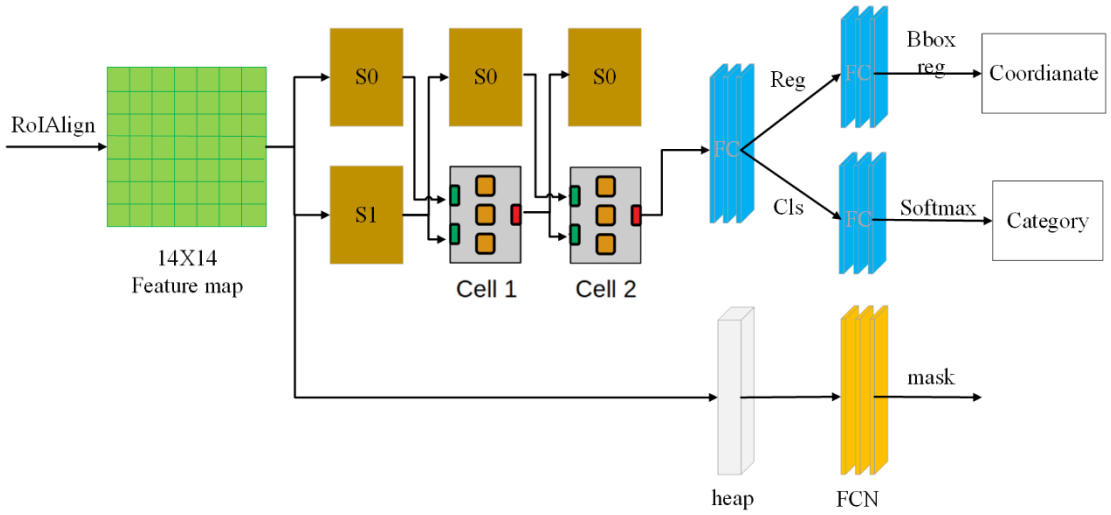


Fig 6: The flow chart of RCNN network structure in Mask RCNN model

IV. Experimental Data Set

The mold coded character data set used in this experiment has a total of 7065 images, which are divided into three folders of train, val and test. The folder for test includes 943 pictures, the folder for val includes 1020 pictures, and the folder for train includes 5102 pictures [15]. The resolution of the image has five sizes: 4032x3016, 3264x2448, 4000x3000, 1445x1080 and 1529x1095, and there is a serious sample imbalance. We intercept a total of 47832 character images from the mold images, including [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G, H, J, K, L, M, N, P, R, S, T, U, V, W, X, Y, Z] 33 categories. The partially cropped character pictures are shown in Figure 7.

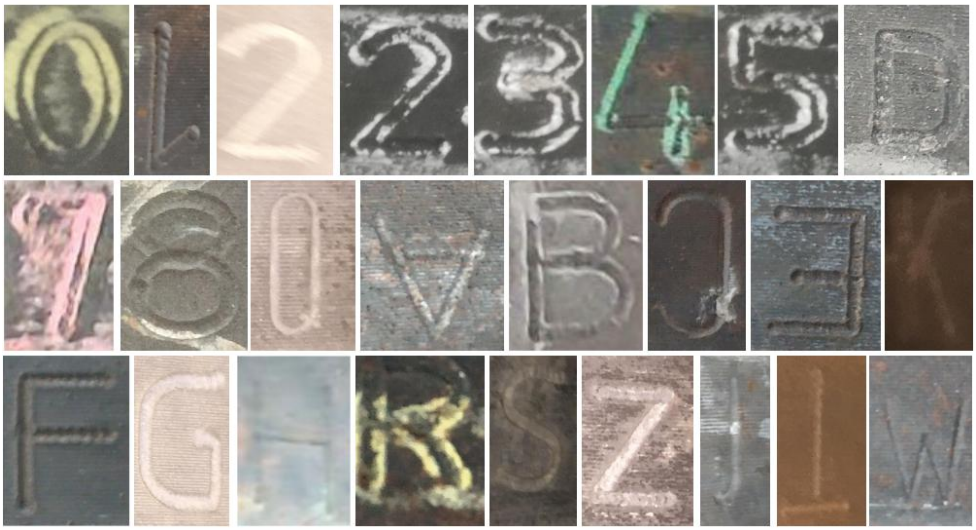


Fig 7: Partially cropped character data set legend

V. Experimental Design and Analysis

Experimental simulation environment: the CPU processor is i7 8700, the memory is 16GB, the graphics card is GTX 1080Ti X2, the hard disk is 2T, and the experimental platform is Ubuntu 18.04. The open source architecture used in the experiment is Tensorflow deep learning framework.

5.1 Experimental parameter settings

The parameters of the network are shown in Table 1.

Table 1The training parameters of Mask RCNN mold coded recognition network based on DARTs

Parameter	Value
Input dimension	1024×1024
Backbone	Resnet101
Scales	[32, 68, 128, 256]
Aspect ratio	[1, 2, 3]
Learning rate	0.003
NMS threshold	0.3
Prediction result threshold	0.65
Epochs	100

The feature extraction network uses the Resnet101 pre-training model, and selects the conv4 of Resnet101 as the feature output map. The scales is set to [32, 68, 128, 256], the aspect ratio is set to [1, 2, 3], the threshold of IoU is set to 0.3 for NMS frame fusion, and the prediction result threshold is set to 0.65. The learning rate is set to 0.0005 and the iterations number is set to 100.

5.2 Experimental evaluation index

In order to verify the effectiveness of the proposed method in various aspects, five evaluation indexes are used to evaluate the performance of the model under the IoU (Intersection over Union) of 0.5 and 0.75 respectively:
1) The average character detection accuracy rate is *AC*, that is, the proportion of the total characters that the predicted detection box classification is the same as the actual characters.

$$ac = correc\_num/total\_num \tag{2}$$

Among them, *correct_num* represents the number of correct check boxes predicted, and *total_num* represents the total number of check boxes in the actual mold data set.

2) The average character error detection rate is *error*, that is, the proportion of the total characters that are different between the predicted detection box classification and the actual characters.

$$error = error\_num/total\_num \tag{3}$$

Among them, *error* represents the number of detection boxes that predict errors.

3) The average character omission rate is *miss*, that is, the proportion of the undetected characters on the mold to the total characters.

$$miss = miss\_num/total\_num \tag{4}$$

Among them, *miss_num* represents the number of missed detection boxes.

4) The average character multi-detection rate is *extra*, that is, the proportion of the total characters that the detection frame does not exist on the actual mold is predicted.

$$extra = extra\_num/total \tag{5}$$

Among them, *extra_num* represents the number of additional prediction detection boxes.

5.3 Experimental results and analysis

The Mask RCNN mold coded recognition model based on DARTs is retrained by using the complete mold coded character recognition data set and the network training parameters shown in Table 1, and the experimental results are shown in Table 2. In Table 2, *(1 cell)*, *(2 cell)* and *(3 cell)* respectively represent the optimized model that uses 1, 2 and 3 cells to improve the classification regression branch in the Mask RCNN model, and the bold values in the table represent the optimal results.

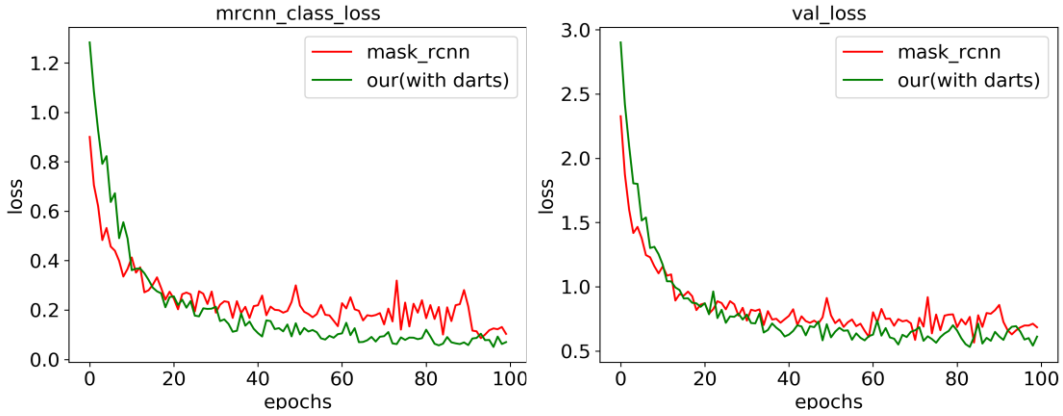
Table 2 Comparison of experimental results of Mask RCNN mold coded recognition model based on DARTs

Model	IoU	Ac	Error	Miss	Extra
Faster RCNN [4]	0.5	93.67	2.95	3.38	4.12
Mask RCNN [2]	0.5	95.27	2.03	2.70	2.63
1 cell	0.5	95.98	2.11	1.91	2.97
2 cell	0.5	97.23	1.94	0.83	2.34
3 cell	0.5	96.08	1.98	1.94	2.11
Faster RCNN [4]	0.75	70.97	2.41	26.62	32.15
Mask RCNN [2]	0.75	76.44	1.77	21.79	20.70
1 cell	0.75	79.67	1.80	18.53	18.67
2 cell	0.75	81.27	1.66	17.07	17.96
3 cell	0.75	80.40	1.88	17.72	17.76

It is not difficult to see from the data in Table 2 that when IoU = 0.5, in the process of improving the classification regression branch model by DARTs, when the number of stacked cells is 1, the model detection effect is similar to that of Mask RCNN, when the number of cell stacks is 2, the model detection effect is obviously better than that of Mask RCNN. This shows that the detection effect of the model may increase with the increase of the number of stacked cells. However, when the number of stacked cells is 3, except for the improvement of the "Extra" evaluation index, the other 3 indicators all begin to decline, which indicates that the model has been overfitted, probably because the number of training samples in the data set is not enough to support the overly complex network structure. The same situation also exists when IoU = 0.75. Even so, the experimental data is sufficient to show that the method of reconstructing the classification regression branch in the Mask RCNN model based on DARTs can effectively improve the performance of Mask RCNN on the mold coded character recognition data set without a large number of parameter adjustment experiments [16].

In addition, comparing IoU = 0.5 and 0.75, with the increase of character recognition requirements, the *AC* evaluation of the original Mask RCNN model decrease from 95.27% to 76.44%, a decrease of 18.83 percentage points, while the *AC* evaluation of the model whose classification regression branch is reconstructed using two cell stacks decrease from 97.23% to 81.27%, a decrease of only 15.96 percentage points. The optimized model is still superior to the original Mask RCNN model by 4.83% [17]. This shows that the stricter the character recognition requirements are, the more significant the proposed method is to improve the performance of the model, which proves the effectiveness of the optimization method.

Figure 8 shows the convergence curve of the classification regression branch and the total loss of training in Mask RCNN network. Among them, (a) represents the classification regression branch loss curve, and (b) represents the model evaluation total loss curve, green line represents the improved model based on DARTs, red line represents the original Mask RCNN model.



(a) (b)

Fig 8: Network total loss training and classification regression branch convergence curve

From Figure 8, it is not difficult to find that comparing the original model and the improved model, the volatility and final convergence results of the total loss convergence curve are almost consistent with those of the classification regression branch loss convergence curve, and the improved model is better than the original model, which further proves that the proposed optimization method is effective [18,19].

5.4 Visualization of results

The visualization of prediction results of the Mask RCNN mold coded recognition model based on DARTs is shown in Figure 9, where the marked boxes of different colors in the picture represent different character codes on the mold [20]. We sort the characters by getting the coordinates of the upper left corner of the predicted character bounding boxes to obtain the actual mold picture coding.

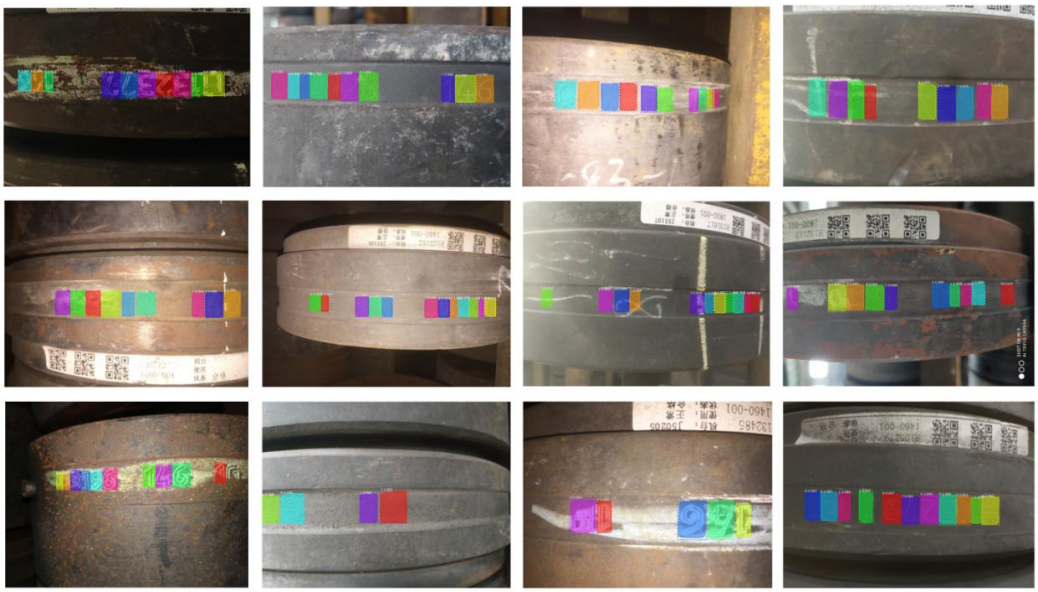


Fig 9: Partial results of model prediction

VI. Conclusions

This thesis proposes a hyperparameter optimization method of Mask RCNN Based on DARTs. Based on the mold coded character recognition method based on Mask RCNN, DARTs technology is used to reconstruct the

classification regression branch in the Mask RCNN network, which realizes the automatic optimization of the model network structure and hyperparameters in the absence of training samples and expert experience, and greatly improves the target detection performance of the model for specific tasks. The next step is to try how to make the algorithm connect the whole network framework while determining the cell structure.

References

[1] B. Liu, "Specific identification detection for small samples," Civil Aviation University of China, 2019.
[2] K. He, Gkioxari. G, Piotr. Dollár, et al., "Mask R-CNN," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 386-397, 2020.
[3] Real. E, Aggarwal. A, Y. Huang, et al., "Regularized evolution for image classifier architecture search," arXiv preprint arXiv: 1802.01548, 2018.
[4] S. Ren, K. He, Girshick. R, et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, vol. 39, no. 6, pp. 1137-1149.
[5] Zoph. B, Q.V. Le, "Neural architecture search with reinforcement learning," arXiv preprint arXiv: 1611.01578, 2016.
[6] H. Liu, Simonyan. K, Y. Yang, "Darts: Differentiable architecture search," arXiv preprint arXiv: 1806.09055, 2018.
[7] Freitas. C.N.C, Cordeiro. F.R, Macario. V, "MyFood: A Food Segmentation and Classification System to Aid Nutritional Monitoring," 2020.
[8] Y. Yu, K. Zhang, L. Yang, et al., "Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN," Computers and Electronics in Agriculture, vol. 163, pp. 104846, 2019.
[9] Q. Zhang, X. Chang, S.B. Bian, "Vehicle-Damage-Detection Segmentation Algorithm Based on Improved Mask RCNN," IEEE Access, vol. 8, pp. 6997-7004, 2020.
[10] Y. Chen, T. Yang, X. Zhang, et al., "DetNAS: Backbone search for object detection," Advances in Neural Information Processing Systems, pp. 6638-6648, 2019.
[11] Ghiasi. G, T.Y. Lin, Q.V. Le, "Nas-fpn: Learning scalable feature pyramid architecture for object detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7036-7045, 2019.
[12] S. Xie, H. Zheng, C. Liu, et al., "SNAS: stochastic neural architecture search," arXiv preprint arXiv: 1812.09926, 2018.
[13] J. Ma, W. Shao, H. Ye, et al., "Arbitrary-oriented scene text detection via rotation proposals," IEEE Transactions on Multimedia, vol. 20, no. 11, pp. 3111-3122, 2018.
[14] Y. Luo, B.Y. Wang, X. Chen, "Research summary of target detection technology based on deep learning," Semiconductor Optoelectronics, vol. 41, no. 01, pp. 1-10, 2020.
[15] C.J. Yang, "Research on Text Detection and Recognition Technology Based on Deep Learning," Harbin Institute of Technology, 2019.
[16] Pham. H, M.Y. Guan, Zoph. B, et al., "Efficient neural architecture search via parameter sharing," arXiv preprint arXiv: 1802.03268, 2018.
[17] Johnson. J.W, "Adapting mask-rcnn for automatic nucleus segmentation," arXiv preprint arXiv: 1805.00500, 2018.
[18] Y.C. Luo, "Overview of scene character recognition," Modern Computer, 2020, no. 04): 32-36.
[19] C. Liu, L.C. Chen, Schroff. F, et al., "Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation," arXiv preprint arXiv: 1901.02985, 2019.
[20] Q. Liu, J.W. Zhai, Z.C. Zhang, et al., "Summary of Deep Intensive Learning," chinese journal of computers, vol. 41, no. 1, pp. 1-27, 2018.