A Value Recognition Algorithm for Pointer Meter Based on Improved Mask-RCNN

Peilin He, Lin Zuo*, Changhua Zhang and Zhehan Zhang
Intelligent Learning Institute for Science and Application
University of Electronic Science and Technology of China
Chengdu, Sichuan 611731, China
Email: peilinhe@std.uestc.edu.cn, (linzuo, zhangchanghua)@uestc.edu.cn and htzhangzhehan@gmail.com

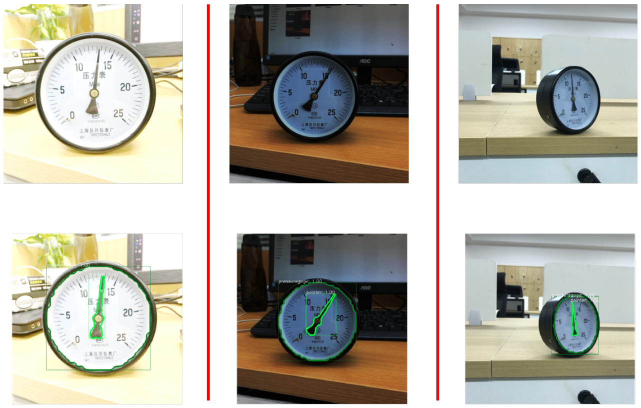
***Abstract*—The automatic reading recognition technology for pointer meter is widely used in the military, industrial, and aerospace fields. However, the accuracy of recognizing pointer meter is susceptible to some factors during the recognition process, such as uneven illumination, complex background, rotation angle, image blur, shooting angle, scale change and other factors, all of which may result in lower accuracy and usability. In this paper, in order to overcome these difficulties, we make improvements to the existing Mask-RCNN network, and propose a novel deep learning based algorithm using PrRoIPooling instead of RoiAlign to achieve the accurate segmentation to improve the accuracy of meter readings. The main procedure of the algorithm is to first use the Mask-RCNN instance segmentation network to segment the meter dial and the pointer area, and then classify and recognize the meter type while fitting the pointer. Finally, the angle reading method is used to calculate the pointer reading. Experiments show that the algorithm is robust, adaptable and effective to the natural environment.**

***Keywords*—Pointer Meter; Mask-RCNN; PrRoIPooling; Deep Learning**

I. Introduction

The pointer meter has the characteristics of simple structure, low cost, strong anti-interference ability and durability, which has been widely used in power equipment monitoring, industrial manufacturing, military, aerospace and other fields. Under normal circumstances, due to technical limitations and other reasons, data reading and recording of pointer meter has always relied on manual observation. However, the problems of manual observation have also been exposed throughout the working process, including large manual workload, poor immediacy, low efficiency, and what's more important – high error rate. As the automation and intelligentization of enterprises accelerates, manual readings can no longer meet the management and maintenance requirements of modern enterprises accelerates, it is necessary to collect, monitor, and record data in the production process in real time. Some researchers proposed the automated means of using computer vision processing techniques to solve the problem which use a camera to acquire images and then use digital image processing techniques to process the dial for reading. However, most algorithms can only run in a specific environment or fixed position, without high reliability, stability and long-term availability.

With the development of high-performance computing and mobile communication technologies, the pointer meter automatic reading recognition technology has become a new hotspot and cutting-edge technology in the field of machine vision and pattern recognition. Especially in recent years, the explosive development of AI(Artificial Intelligence) and the deep learning algorithm have made a major breakthrough, attracting scholars to use deep learning methods to solve the problems encountered in the pointer meter recognition process.



(a)overexposure **(b)underexposure** **(c) tilted**

Fig. 1. The effect diagram of extracting the pointer and dial under different conditions of illumination and meter tilt.

In this paper, we propose a novel pointer meter recognition algorithm based on deep learning. The technical principle of the algorithm is shown in Figure 1, which shows the detection effect of the dial and pointer under the special conditions of overexposure, underexposure and tilt. In general, our algorithm consists of four steps. First of all, we use Mask-RCNN [1] classification meter and extract the basic parameters of the corresponding meter from the database, as well as segment the dial and pointer binary mask. Secondly, we calculate the perspective transformation matrix based on the dial binary mask, and then use the perspective transformation matrix to correct the pointer binary mask. After that, we use PCA(Principal Component Analysis) [2] algorithm to fit the pointer binary mask and obtain the slope. Finally, we accurately calculate the reading based on the basic parameters of the meter, the slope of the pointer, and the angle method, A more detailed algorithm procedure is shown in Figure 2. In addition, in order

978-1-7281-2106-2/19/$31.00 @2019 IEEE

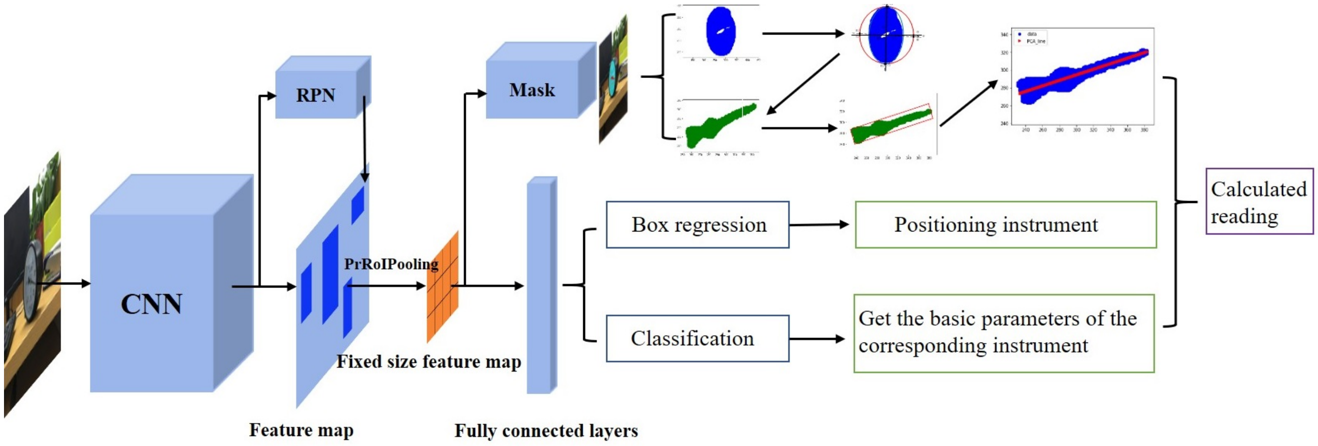


Fig. 2. The framework of the improved Mask-RCNN based for pointer meter reading recognition.

to simulate the real environment as much as possible, we have prepared overexposure, underexposure, Gaussian noise, salt and pepper noise, complex background and different scale pictures, which greatly improve the robustness, flexibility and adaptability of the model. The experimental results show that we have successfully solved the problem of the influence of uneven illumination, complex background and different scales on meter recognition.

II. Releated work

A. Pointer Meter Recognition

A lot of work have been done in pointer meter recognition in the past few years. In general, the existing pointer meter recognition algorithm can be divided into the traditional algorithms based on digital image processing technology and the modern algorithms based on machine learning and deep learning. Traditional algorithms have proposed to use the template matching and table lookup method to perform pointer recognition and reading the pointer meter's image [3]. Traditional algorithms also proposed to use the image subtraction method to extract the pointer [4] and use the Hough transform algorithm to detect the circle to complete pointer meter reading recognition [5]. Although these algorithms worked well in certain scenarios and could achieve high reading accuracy, they were not well adapted in the natural environment due to the extremely high illumination conditions required for image processing. In recent years, some scholars have also proposed novel modern algorithms for meter recognition. For example, Liu *et al.* [6] proposed to use the SVM (Support Vector Machine) machine learning algorithm to locate and separate the meter, and Liu *et al.* [7] proposed to use the Faster-RCNN object detection algorithm to locate and extract the meter. To a large extent, it solved the problems left by traditional algorithms, such as different scales, complex backgrounds, and difficulty in locating meter. However, there is no algorithm that can solve the problem of uneven illumination, large range of illumination variation, and tilting of the meter.

B. Object Detection Based on Deep Learning

Thanks to CNN's excellent performance in image classification tasks, a large number of computer vision algorithms based on CNN and deep learning have been proposed which is much better than the traditional algorithms. In recent years, the significant advances in object detection in the field of computer vision include the 2014 RCNN [8], the 2015 FCN [9], the 2015 Fast-RCNN [10] and the same year's Faster-RCNN [11]. In 2017, FPN [12] and Mask-RCNN [1] pushed image classification, semantic segmentation and instance segmentation to its highest point. Among them, Mask-RCNN has achieved the state-of-the-art performance, and then it is also widely promoted to the industry for application. For example, Nie *et al.* [13] proposed a new method based on Mask-RCNN for inshore ship detection and the experimental results on Google Earth showed the effectiveness of their method. Sorokin *et al.* [14] proposed a lesion analysis and diagnosis algorithm based on Mask-RCNN, etc. It can be seen that the maturely developed deep learning object detection algorithm has been applied to all walks of life. Therefore, in this paper, we also propose a novel deep learning algorithm based on Mask-RCNN to solve the problem of automatic reading of industrial field pointer meter.

III. Approach

A. PrRoIPooling

As discussed in the previous section, we improve the Mask-RCNN framework and use PrRoIPooling [15] to replace RoiAlign [1] to improve the instance segmentation accuracy. PrRoIPooling is a RoI average pool method based on bilinear interpolation. It avoids any quantization and has a continuous gradient on bounding box coordinates. It is different from the original RoIPooling proposed in Fast-RCNN. PrRoIPooling uses average pooling instead of max pooling for each bin and has a continuous gradient on bounding box coordinates. It is also different from the RoIAlign proposed in Mask-RCNN. PrRoIPooling uses a full integration-based average pooling instead of sampling a constant number of points.

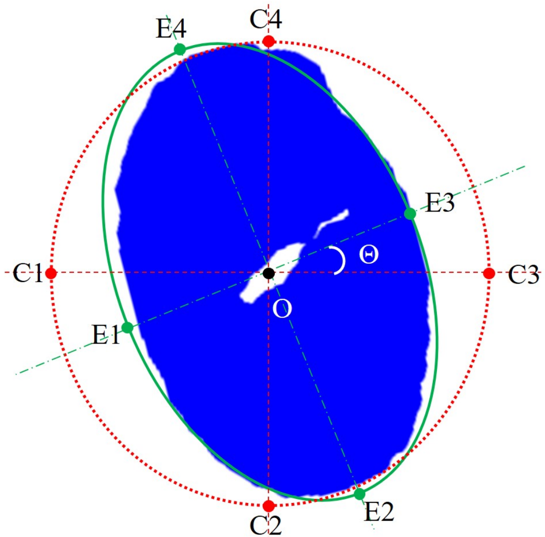


Fig. 3. Schematic diagram of perspective transformation algorithm based on irregular ellipse.

This makes the gradient w, r, t of the coordinates continuous and more details can be seen in [15]. After that, we can pass an image to the improved Mask-RCNN to get the meter type and the meter location, the dial binary mask denotes as $I = (W_i, Q_i), i = 1, 2, 3...n$ and the pointer binary mask denotes as $P = (G_j, P_j), j = 1, 2, 3...q$.

B. Perspective Transformation

We propose a novel perspective transformation algorithm based on irregular ellipse to solve the problem of tilting image due to the unfixed shooting angle. The algorithm can effectively correct the tilting image, and the schematic diagram can refer to Figure 3. In the previous section, after getting the binary mask of the dial through Mask-RCNN and fitting the minimum circumscribed circle and the ellipse of the meter binary mask respectively, we can get the parameters of the circle $(C_x, C_y), \mathrm{r}$ as well as the parameters of the ellipse $(E_x, E_y), (2\mathrm{a}, 2\mathrm{b}), \theta_\mathrm{e}$ through OpenCV [16]. After that, we can get 8 points $(C1 - C4, E1 - E4)$ in the four directions of horizontal, vertical, long axis and short axis, please refer to Eq. 1-6.

$$c = E_y - k \times E_x \tag{1}$$

$$A = 1 + k^2 \tag{2}$$

$$B = -2 \times (k^2 + 1) \times E_x \tag{3}$$

$$C = (1 + k^2) \times {E_x}^2 - f \tag{4}$$

$$X_i = \frac{-B \pm \sqrt{B^2 - 4 \times A \times C}}{2 \times A}, i = 1, 2, 3, 4 \tag{5}$$

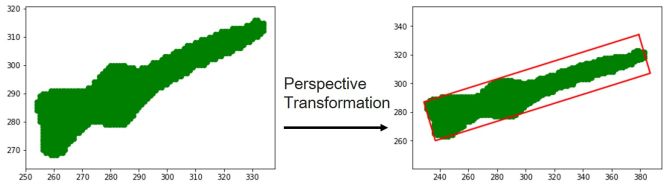


Fig. 4. The effect diagram of our perspective transformation algorithm to correct pointer meter binary mask.

$$Y_i = k \times X_i + c, i = 1, 2, 3, 4. \tag{6}$$

Where (c, A, B, C) represent a constant term, a quadratic coefficient, a first term coefficient, and a zero term coefficient, respectively. $(X_i, Y_i), i = 1, 2, 3, 4$ represents the four point coordinates of the ellipse intersecting the long axis and the short axis. $(E_x, E_y), (2\mathrm{a}, 2\mathrm{b})$ and θ_e indicate the central coordinate, the length of the long axis, the length of the short axis, and the deflection angle of the smallest circumscribed ellipse, respectively. (C_x, C_y) and r represent the center coordinates and the radius of the smallest circumscribed circle, respectively. When we calculate the intersection on the long axis, $k = \tan\theta_e, f = a$. On the contrary, when we calculate the intersection on the short axis, $k = -1/\tan\theta_e, f = b$. Therefore, the 8 coordinate points we get can be expressed as $(X_i, Y_i), i = 1, 2, 3, 4$ and $(C_x \pm r, C_y \pm r)$.

Finally, we bring the 8 coordinate points into Eq.9 and Eq.10 to solve the perspective transformation matrix $\mathbf{M}$. After that, the pointer binary mask can be corrected by the perspective transformation matrix $\mathbf{M}$. The effect diagram is shown in Figure 4.

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{M} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{8}$$

$$x' = \frac{X}{Z} = \frac{m_{11} * x + m_{12} * y + m_{13}}{m_{31} * x + m_{32} * y + m_{33}} \tag{9}$$

$$y' = \frac{Y}{Z} = \frac{m_{21} * x + m_{22} * y + m_{23}}{m_{31} * x + m_{32} * y + m_{33}} \tag{10}$$

Where $(x, y, 1)$ represents the three-dimensional coordinates before the meter perspective transformation; (X, Y, Z) represents the three-dimensional coordinates of the meter after the perspective transformation, $\mathbf{M}$ is the perspective transformation matrix; (x', y') is the coordinate converted into the two-dimensional space; $(m_{ij}, i, j = 1, 2, 3)$ represents the elements of the perspective transformation matrix.

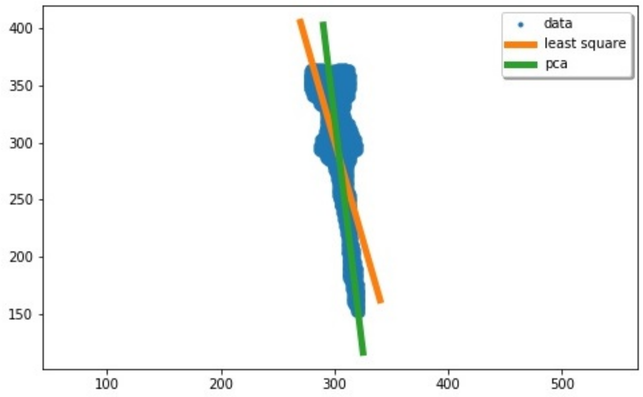


Fig. 5. The effect of the least squares and the PCA straight line fitting.The blue part is the pointer binary mask, the orange part is the least squares straight line fitting, and the green part is the PCA straight line fitting.

C. PCA Linear Fitting

We have obtained the corrected pointer binary mask in the previous section, and in order to get the slope of the pointer K we need to fit the binary mask of the pointer. Fitting the line with least squares is a common method we use to solve such problems [17], and the principle is to find the best function match for a set of data by minimizing the sum of the squares of the errors. Although this is a standard linear solution to the classical problem, since the error source of this process is the vertical distance between the point and the line, the method cannot represent a vertical line at all. And, the more dispersed the point is, the worse the fitting effect will be. In practice, we can fit the line with the actual distance from the point to the line, which is the PCA algorithm we used.

In order to verify that PCA performs better than least squares, we use these two algorithms to fit the pointer binary mask, respectively. It is not difficult to see from the experimental results that PCA is more superior and stable than least squares, please refer to Figure 5.

D. Calculate Reading

Although we have obtained the pointer slope K, we can't calculate the meter reading directly from the slope because there may be two different directions for the pointer meter under the same slope. For example, if the angle of the pointer is zero degrees, the slope is zero; if the angle of the pointer is 180 degrees, the slope is also zero, then we cannot be sure that the direction of the pointer depends only on the slope. So we design a judgment mechanism, the principle can be referred to Figure 6. The principle of the direction judging mechanism is very simple that the quadrant pointed by the current pointer is determined by the positional relationship of the two points $O = (O_x, O_y)$ and $C = (C_x, C_y)$. For example, $C_x > O_x$ and $C_y > O_y$, it can be judged that the pointer points to the first quadrant, which can be seen from Table I for more information.

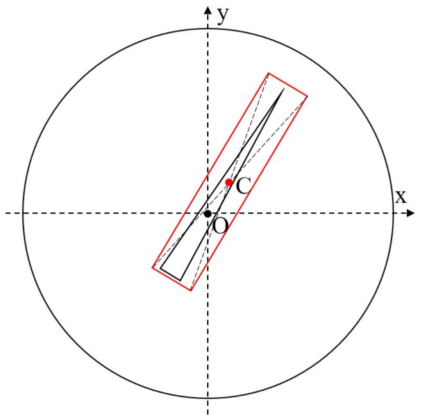


Fig. 6. The schematic of pointer direction judgment mechanism. the black point is marked as O in the center of the circle, the red point marked as C is the center point of the pointer binary mask, and the red diagonal rectangle is the smallest bounding rectangle of the pointer.

TABLE I
POINTER DIRECTION JUDGMENT MECHANISM

X	Y	Quadrant	Angle(rad)
$C_x > O_x$	$C_y = O_y$	Y positive axis	$\theta=\pi/2$
$C_x = O_x$	$C_y < O_y$	X negative axis	$\theta = \pi$
$C_x < O_x$	$C_y = O_y$	Y negative axis	$\theta= -\pi/2$
$C_x = O_x$	$C_y < O_y$	X positive axis	$\theta = 0$
$C_x > O_x$	$C_y > O_y$	First quadrant	$\theta = atan(K)$
$C_x > O_x$	$C_y < O_y$	Fourth quadrant	$\theta = atan(K)$
$C_x < O_x$	$C_y > O_y$	Second quadrant	$\theta = atan(K) + \pi$
$C_x < O_x$	$C_y < O_y$	Third quadrant	$\theta = atan(K) + \pi$

After that, we can obtain the angle θ of the pointer in the plane rectangular coordinate system. In addition, we have obtained the basic parameters of the meter in the previous section which include $\theta_{\min}$, $\theta_{\max}$, $v_{\min}$ and $v_{\max}$. Therefore, We can express the angular density as $\varepsilon = (\theta_{\min} - \theta_{\max})/(v_{\max} - v_{\min})$, and then judge whether the meter range is exceeded. If $\theta_{\min} * \pi/180 < \theta < \theta_{\max} * \pi/180$, the meter reading $v = 0$. Otherwise, the meter reading can be expressed as follows:

$$v = v_{\min} + \frac{(\theta_{\min} - \theta * 180/\pi)}{\varepsilon} \tag{11}$$

Where $\theta_{\min}$, $\theta_{\max}$, $v_{\min}$ and $v_{\max}$ represent the minimum angle, the maximum angle, the minimum value and the maximum value, respectively. v represents the reading value, and θ represents the pointer radians.

IV. EXPERIMENTS

A. Experimental Settings

Dataset. The dataset used is composed of 1430 RGB images collected by ourselves, which contains five different situations: original image, overexposure, underexposure, Gaussian noise, and salt and pepper noise, enhancing the adaptability of the model to illumination conversion. Each image is 600 × 600 pixels, and contains two types: pressure gauge and pointer.

TABLE II
THE MASK-RCNN INSTANCE SEGMENTATION RESULTS UNDER DIFFERENT METHODS

Method	Total train time(hr)	Bounding Box				Instance Segmentation			
		AP	AP50	AP75	AR	AP	AP50	AP75	AR
ResNet-50-C4	4.50	0.7292	0.9349	0.8064	0.7660	0.6298	0.8880	0.6570	0.6750
ResNet-50-FPN	1.27	0.7870	0.9652	0.8760	0.8200	0.6947	0.9399	0.6998	0.735
ResNet-101-FPN	1.60	0.8080	0.9550	0.9050	0.841	0.6990	0.9360	0.6870	0.741
ours. ResNet-50-C4-PrRoIPooling	4.68	0.7520	0.9405	0.8367	0.7890	0.6450	0.8947	0.6393	0.6840
ours. ResNet-50-FPN-PrRoIPooling	1.37	0.8136	0.9552	0.9123	0.8420	0.7060	0.9454	0.7098	0.7450
ours. ResNet-101-FPN-PrRoIPooling	1.57	**0.8250**	**0.9701**	**0.9290**	**0.8530**	**0.7130**	**0.9506**	**0.7270**	**0.7640**

And according to the ratio of 6 : 4, the data set is divided into 1000 pictures as the training set and 430 pictures as the verification set. In addition, to enhance the adaptability of the model to natural environment, the dataset covers complex backgrounds, multiple scales, various tilt angles, etc.

Experimental Environment. The software environment for the experiment code running in this paper is Pytorch1.0, Python3.6, and OpenCV3.1. The hardware environment is NVIDIA TITAN X(Pascal) GPU and Intel Xeon Processor E5-2650 CPU. It takes 1 hour and 34 minutes to train 15,000 iterations.

Implementation Details. In this paper, the convolutional neural network model is ResNet. Take ResNet50 as an example, first enter the convolution of $7 \times 7 \times 64$, then pass through $3+4+6+3 = 16$ building blocks and each block has 3 layers. Finally, there is a fully connected layer for classification, so there are a total of 50 layers. For more detailed parameters, please refer to [18]. In this experiment, the initial learning rate is 0.02, the weight decay is 0.0001, the maximum number of iterations is 15,000, and the learning rate begins to decay at 5,000 and 10,000. For more setting of hyperparameters, please refer to [1].

B. Experimental Results

Instance Segmentation Accuracy Assessment. We conduct three experiments under the framework of ResNet-50-4C, ResNet-50-FPN, ResNet-101-FPN as our baselines. Then we want to improve the accuracy of the instance segmentation, so we propose to replace the RoiAlign with PrRoIPooling to optimize the original model, and conduct the same experiment on the three models under the same conditions at the same time. We use AP, AP50, AP75 and AR as the evaluation criteria, where AP represents the average of all the precisions of the IoU(Intersection Over Union) threshold from 0.5 to 0.95, AP50 indicates the average precision in the case where the IoU threshold is greater than 0.5, AP75 indicates the average precision in the case where the IoU threshold is greater than 0.75 and AR represents the average of all recalls with an IoU threshold of 0.5 to 0.95.

TABLE III
EXPERIMENTAL COMPARISON OF CPU AND GPU TIME COSTS

Average	CPU Cost(s)	GPU Cost(s)
-	2.8724	**0.0931**

TABLE IV
READING RECOGNITION ACCURACY COMPARISON

Method	Average Relative Error(%)	Average Reference Error(%)
Chi *et al.*(2015) [19]	**2.135**	0.584
Zheng *et al.*(2016) [20]	9.95	0.954
Gao *et al.*(2018) [21]	2.433	0.44
ours.	2.35	**0.1708**

The experimental results show that by replacing the RoiAlign in the original model, we can obtain the higher bounding box performance and improve the performance of instance segmentation by using PrRoIPooling, as shown in Table II. Our bounding box classification AP can reach 0.8250, which is 2 to 3 percentage points higher than the original average, and our instance segmentation AP can reach 0.7130, which is 1 to 2 percentage points higher than the original average. Comparing the original model of the first three rows of Mask-RCNN and the improved model of the last three lines, we can find that the training time is not much different, indicating that our improved method does not bring more time overhead. Secondly, with more layers of ResNet and FPN, we can get more excellent classification and instance segmentation performance.

Compare CPU and GPU Time Cost. Our algorithm can run in GPU or CPU hardware environment. We test 120 sets of data to record the time cost of CUP and GPU respectively. The data after calculating the average value are recorded in Table III. The experimental results show that the GPU has a huge time advantage, and the average processing time per image is 0.0931 seconds. Although the average time overhead for running on the CPU is 2.8724 seconds, it will get better in the future as the model moves toward lighter weight.

Reading Recognition Accuracy Comparison. Other authors' methods may use different meter to do experiments, in order to make the comparison data more accurate, we choose the two parameters of the average relative error $\bar{\delta}$ and the average reference error $\bar{\gamma}$ as evaluation indicators, as shown in Eq. 12 and Eq. 13.

$$\bar{\delta}=\frac{\sum\limits_{i=1}^{n}\frac{|a_i-A_i|}{A_i}}{n}\times 100\% \tag{12}$$

$$\bar{\gamma} = \frac{\sum_{i=1}^{n} \frac{|a_i - A_i|}{S}}{n} \times 100\% \qquad (13)$$

Where a_i represents the algorithm predictive reading value, A_i represents the true reading value, S represents the meter's range, and n represents the total number of experimental data. As shown in Table IV, the experimental results show that although the average relative error 2.35% is slightly inferior to Chi *et al.* [19], the average reference error is still far ahead, which proves that our algorithm has higher reading recognition accuracy. In addition, our algorithm is different from the traditional algorithm, and it is completely considering the meter reading recognition from the perspective of deep learning. Therefore, the algorithm not only can achieve the accuracy of the traditional algorithm or even a slight lead, but also has a high adaptability to the natural environment, such as complex background, illumination changes, scale changes, image tilt and so on.

V. Conclusion

We propose a novel pointer meter recognition algorithm based on deep learning, which is completed by Mask-RCNN instance segmentation, perspective transformation, PCA line fitting and angle method. Owing to the utilization of deep neural network in the dial and pointer learning of our work, the characteristics of them can be segmented directly from the image together with avoiding the process of preprocessing the meter image in the traditional algorithm, thereby effectively solving the uneven illumination, complex background, and meter tilted, scale changes and other issues. At present, the algorithm has been successfully applied to substation inspection robots. In addition, the algorithm can be applied to the identification of digital meters with a little modification, which is the subject of our next research.

Acknowledgment

This work was supported in part by the National Science Foundation of China under Grants 61877009 and 61573081, Sichuan Provincial Science and Technology plan project under Grant 2018GZ0396 and 2019YFG0451.

References

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[2] I. Jolliffe, *Principal component analysis*. Springer, 2011.

[3] W. Mo, L. Pei, q. Huang, Y. Zhang, W. Fu, and Y. Zhao, "Development of automatic verification system for high precision pointer instrument based on template," *Electrical Measurement and Instrumentation*, vol. 54, no. 12, pp. 100–105, 2017.

[4] J. Shi, D. Zhang, J. He, C. Kang, J. Yao, and X. Ma, "Design of remote meter reading method for pointer type chemical instruments," *Process Automation Instrumentation*, vol. 35, no. 5, pp. 77–79, 2014.

[5] G. NI and B. YAN, "Pointer instrument image recognition based on priori characteristics of instrument structure," *Electronic Science and Technology*, vol. 26, no. 10, pp. 10–12, 2013.

[6] H. Liu, "An intelligent methods for pointer instrument reading based on machine vision," in *2016 International Conference on Modeling, Simulation and Optimization Technologies and Applications (MSOTA2016)*. Atlantis Press, 2016.

[7] K. Liu, "Recognition of the analog display instrument based on deep learning," Master's thesis, Huazhong University of Science and Technology, 2017.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[10] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[12] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, vol. 1, no. 2, 2017, p. 4.

[13] S. Nie, Z. Jiang, H. Zhang, B. Cai, and Y. Yao, "Inshore ship detection based on mask r-cnn," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 693–696.

[14] A. Sorokin, "Lesion analysis and diagnosis with mask-rcnn," *arXiv preprint arXiv:1807.05979*, 2018.

[15] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 784–799.

[16] A. Kaehler and G. Bradski, *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. " O'Reilly Media, Inc.", 2016.

[17] D. York, "Least-squares fitting of a straight line," *Canadian Journal of Physics*, vol. 44, no. 5, pp. 1079–1086, 1966.

[18] 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.

[19] J. Chi, L. Liu, J. Liu, Z. Jiang, and G. Zhang, "Machine vision based automatic detection method of indicating values of a pointer gauge," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[20] C. Zheng, S. Wang, Y. Zhang, P. Zhang, and Y. Zhao, "A robust and automatic recognition system of analog instruments in power system by using computer vision," *Measurement*, vol. 92, pp. 413–420, 2016.

[21] J. Gao, L. Guo, Y. Lv, Q. Wu, and D. Mu, "Research on algorithm of pointer instrument recognition and reading based on the location of the rotation center of the pointer," in *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2018, pp. 1405–1410.