Reading Digital Numbers of Water Meter with Deep Learning Based Object Detector

Shirong Liao[1], Pan Zhou[1], Lianglin Wang[2], and Songzhi Su[1(✉)]

[1] School of Informatics, Xiamen University,
Xiamen 361005, People's Republic of China
ssz@xmu.edu.cn
[2] Ropeok Technology Group Co., Ltd., Xiamen, China

Abstract. Automatically reading water meter is a classical OCR problem, typical method includes four major components: region of interests (ROIs) detection, skew correction of bounding boxes, single digital character segmentation, and digital classification. Disadvantage of the traditional method is that the pipeline is too complex and coupled to the accuracy of the final recognition result. Deep learning based object detection has achieved promising results on many computer vision tasks. As one of the representatives of the deep learning object detection framework, YOLOv3 perform detection task quickly and accurately. Inspired by this, we formulate the water meter reading problem as a detection problem, which is a true end-to-end solution. In order to attack the half-character problem of water meter, we proposed a heuristic rule to guarantee that there is only one bounding box in the vertical direction within a grid. Experimental results on our own built XMU-W-M dataset showed that the 0-error recognition rate reaches 96.67% and the 1-error recognition rate is up to 99.81%, which outperforms the traditional water meter recognition system in both time and precision. Both the code and dataset are available: https://github.com/sloan96/water-meter-recognition.

Keywords: Water meter recognition · Traditional method · Deep learning · YOLOv3 · Rule

1 Introduction

Reading the water meter automatically makes our daily life more convenient. The general meter-reading process is as follows: the camera is fixed above the water meter waiting for power-on command, the picture is taken and its binary code is sent to the terminal immediately once the command is received. Terminal collects the binary code and sends data to the platform software while issuing a power-off command to the camera. The platform software decodes the received binary data into an image format, recognizes the water meter number and analyzes the result.

Supported by organization x.

Z. Lin et al. (Eds.): PRCV 2019, LNCS 11857, pp. 38–49, 2019.
https://doi.org/10.1007/978-3-030-31654-9_4

Optical Character Recognition (OCR) refers to converting text on an image into computer-editable text content. There are already large numbers of studies in this area, such as license plate recognition. Digital character recognition is a traditional research topic of pattern recognition and it's still studied by many researchers and widely used in many domains. However, due to the specific application scenarios, the situation is different and problems are of great diversity. In this paper, we focus on the image digital recognition part of the above process and regard the recognition problem as a detection problem.

As we can see from Fig. 1, the collected images mainly lead to four challenges for recognition. (1) Camera installation and uneven illumination of the light source lead to image distortion. (2) Irrelevant characters existed in the dial will affect the recognition. (3) The rotation of the water meter makes the numbers not in a horizontal line. (4) Uncertainty caused by digital rotation changes.

Fig. 1. The examples of water meter images. All the three sub-images have difficulty (1) and (2), the second and third sub-image significantly have difficulty (3) and (4).

In this work, we turn to a deep learning framework – YOLOv3. You only look once (YOLO) [1] is a state-of-the-art, real-time object detection system. We no longer need to perform tedious preprocessing such as digital segmentation and skew correction thanks to the one stage pipeline of YOLOv3 [3]. However, one mainly trouble is the half-word problem, namely, more than one detected bounding boxes in a digital grid caused by the above mentioned difficulty (4). In order to handle this situation, we add a heuristic rule to the network.

The main contributions of this paper are: (1) recognizing multiple digits without digital segmentation, (2) modeling ROIs localization as detection problem, so that the slope is not required to be estimated by basic image processing techniques, (3) proposing a heuristic method to tackle the half-word problem. We also introduce a new dataset of both realistic and virtual water meter dial images generated by using GAN [12], and experimentally evaluate our adjusted model.

2 Related Work

The traditional water meter recognition system generally includes five modules: (1) water meter digital area detection, (2) digital rectangle area location, (3) rectangular box skew correction, (4) digital segmentation, and (5) digital recognition.

As Fig. 2 shows, inputting an image containing a water meter, after jpeg compression, thresholding, tilt angle detection and correction, throw the image at this time into the trained Support Vector Machine (SVM) classifier [10], where the HOG [9] feature is usually used. Getting the water meter digital area, scaling it to the specified size, generally using the method based on the maximum interval width of adjacent characters for digital segmentation, sometimes we discard this step but regard single-character recognition as end-to-end multi-label classification and throw it into the trained Convolutional Neural Networks, finally we could obtain the recognition result.

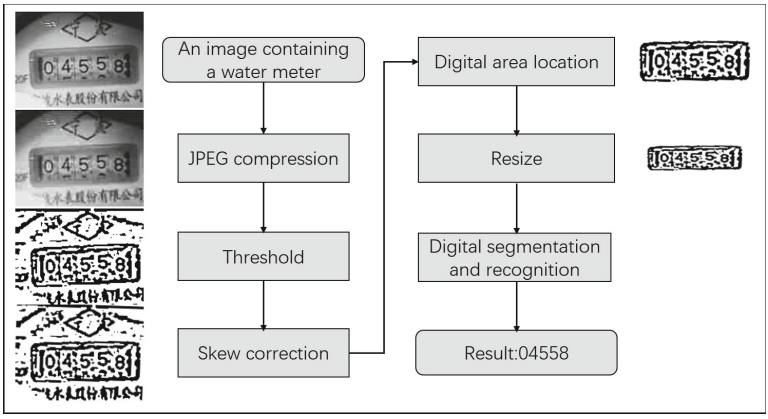


Fig. 2. The pipeline of traditional water meter recognition, mainly including region of interests (ROIs) detection, skew correction of bounding boxes, single digital character segmentation, and digital recognition

Over the years, a large number of digital recognition methods have been put forward. The traditional approach is to design and extract features, and input them to the classifier, then a digital classifier model could be established. However, feature design is very time-consuming and single designed features tends to result in low generalization ability, therefore [11] proposed to replace hand crafted features with features learned by unsupervised algorithms like K-means [13]. Another method is character template matching, generally including digital template definition, digital area segmentation and digital matching. However, it is too simple to be applied to a slightly complicated situation. Lately, the emergence of powerful deep learning techniques has led to plenty of digital

recognition methods based on neural network. For example, the famous LeNet-5 [4] proposed by Yann LeCun, which has 7 layers. The input 2D image is first passed through the convolution layer to the pooling layer, then through the fully connected layer, and finally using the softmax classification as the output layer. Based on [4,5] presents a feed-forward network architecture for recognizing an unconstrained handwritten multi-digit string. Lately, Qiang Guo proposed a method to combine the Hidden Markov Model (HMM) and deep learning methods to locate and identify the numbers in the natural scene [6]. But a problem is that training networks needs a large number of data and better hardware condition.

In our paper, we are committed to water meter digital recognition. For the purpose of achieving a high accuracy within a really short time, we take inspiration from YOLOv3 framework to regard the localization of ROIs as a detection problem and simplify the pipeline. In addition, we specially design several rules to iron out problems arising from the detection process. Surprisingly, under our attempts, the final model could hit a high accuracy level.

3 Self-built Water Meter Dataset

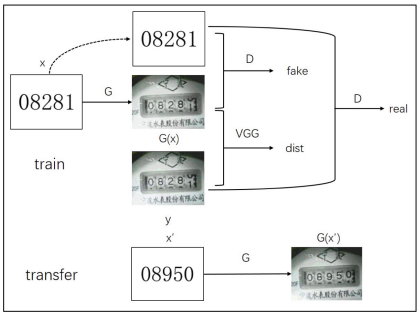
3.1 Data Generation

The original idea was to use pix2pixHD [7] open source framework to simulate the generation of water meter data.

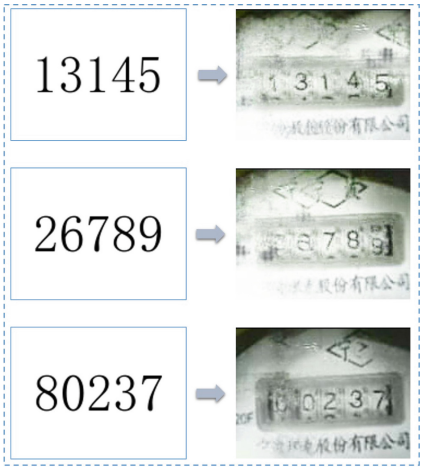
pix2pixHD is a variant of GAN whose input consists of a digital map x and a true label y corresponding to x. Training generator G and generating $G(x)$ to make it realistically true to the true label y. D is a discriminator, the input x and the generated image $G(x)$ are determined to be false as much as possible, in contrast, the input x and the real label image y are determined to be true as much as possible. VGG is used for calculating the perceptual reconstruction loss [8] between the real label image and the generated image.

Using pygame and font library to generate a label image corresponding to the real water meter number, i.e. '08281' image with white background. The image generated by pygame rendering should be the same size as the real water meter image. The corresponding white background label image is rendered according to the label of the real water meter, and the generator $G(x)$ is trained by the pix2pixHD framework. The training and transfer process is shown in Fig. 3(a). We only need to generate the digital image X' we want, that is, we can render the corresponding water meter digital image $G\,(x')$.

As shown in Fig. 3(b), we find that there is still a gap between the data generated by this method and the real data. The improvement of the effect requires a large number of real samples, which is of little significance for our training, but can be considered for image noise rendering. Considering that the training requires a large number of real samples, we simulate the actual scene to capture the water meter image, as shown in Fig. 3(c). These data are more in line with the real scene, followed by data labeling issues.

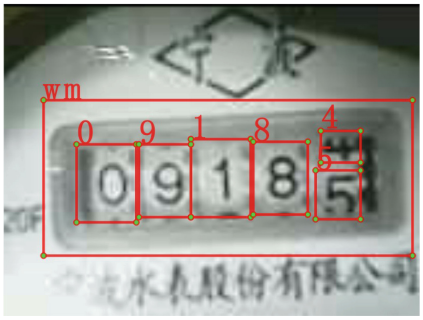


(a) The general training and transfer process of data generation



(b) Actual generation effect

(c) Real samples obtaining



(d) Water meter data annotation

Fig. 3. Data generation and annotation

3.2 Data Annotation

We use the open source tool labelImg[1] for data annotation. As shown in Fig. 3(d), we mark each number in the water meter image and also add a 'wm' category, which is convenient for post-processing to distinguish the numbers inside and outside the box. For the area where the digital rotation changes, the numbers appearing above and below should both be marked, and the marked box should be as close as possible to the number, which can ensure that the predicted bounding box can also fit the number as much as possible, which helps us to apply rules to reduce errors. When we encounter a blurred digital area that is unrecognizable to the human eye, no marking is required, because this type of area contains too much noise. Once labeled, it is easy to make the model learn the wrong information and increase the background false detection rate.

[1] Tzutalin. LabelImg. Git code (2015). https://github.com/tzutalin/labelImg.

4 Proposed Method

In this section, we first briefly introduce the principles and framework of YOLOv3, then introduce how to regard digital recognition as a detection problem, and finally detail how the additional rules are to reduce the error rate step by step.

4.1 YOLOv3

YOLO divides the input image into $S \times S$ grids, and each grid unit is responsible for detecting targets falling into it. Each grid unit predicts a confidence score corresponding to the B bounding boxes and the bounding box, the confidence reflecting whether the bounding box contains the likelihood of the target. As Eq. (1) shows, the confidence is defined as Pr (Object) $*\mathrm{IOU}_{\mathrm{pred}}^{\mathrm{truth}}$. If no object exists in that cell, the confidence scores should be zero. Otherwise the confidence score should equal the intersection over union (IOU) between the predicted box and the ground truth.

$$Confidence\,score = P_r(Object) \times IOU_{pred}^{truth} \tag{1}$$

If the grid cell does not contain a target, the confidence should be 0, otherwise the confidence is equal to the prediction box and the Ground Truth's IOU. Each bounding box contains 5 predicted values: x, y, w, h, confidence. The (x, y) coordinates represent the center point of the bounding box, and w and h represent the width and height of the bounding box. Each grid unit also predicts C (the number of categories) conditional category probabilities $P_r(Class|Object)$, and the prior condition is that the grid unit contains the target. Regardless of how many bounding boxes are predicted, each grid unit only predicts a set of category probabilities. If there are 20 categories, then each grid unit will only predict a set of 20 categories of probabilities, so a map corresponds to a predicted value of $S \times S \times (B * 5 + C)$.

YOLOv3 uses the cluster center as the anchor box. But it uses logistic regression instead of the previous softmax, which effectively improves the case where a bounding box predicts only one category and the near-small target detection rate is not high. YOLOv3 predicts the bounding box at three different scales, where the author uses a similar feature pyramid network. At the same time, a hybrid method for Darknet-19 and novel residual network is proposed to realize feature extraction, which is named Darknet-53 because it has 53 convolution layers. The specific structure is shown in Fig. 4.

4.2 Regard as a Detection Problem

An example of recognition is shown in Fig. 5. We have no needs to do any splitting on the numbers on the image but input the full image to the trained YOLOv3 model, then locations of ROIs will be directly detected and presented with bounding boxes sorted from small to large according to the x coordinate of

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig. 4. Darknet-53

Fig. 5. Water meter recognition flowchart in this paper, YOLOv3 for RoIs regression and classification, then additional algorithm to solve half-word problem

the upper left corner. At the same time, classification results are also generated. When the rules are formulated, the converted string is finally outputted. See Algorithm 1 for details.

4.3 Additional Rules

As we all know, there are five digits in the water meter. Predictably, three scenarios are predicted, as shown in Fig. 6:

a. The number of predicted bounding boxes is greater than five except 'wm' category
b. The number of predicted bounding boxes is equal to five except 'wm' category
c. The number of predicted bounding boxes is less to five except 'wm' category

In the second case, the digital length is 5, however, there may be a situation in which the prediction is wrong or even there are two bounding boxes in the nearly vertical position while some positions have no bounding box. This kind of

Algorithm 1. Detection into recognition

Require: Water meter image IMAGE
Ensure: Water meter number NUMBER
1: r = YOLOv3.detect(IMAGE) //r contains the predicted N categories C[N] and their corresponding bounding box position information P[N] and score S[N],where $P^{(i)} = [x_{min}^{(i)}, y_{min}^{(i)}, w^{(i)}, h^{(i)}]$ $i \in [1, N]$;
2: Construct a new empty list of boxes;
3: **for** $i \leftarrow 1$ to N **do**
4: **if** $C^{(i)} \neq$ "wm" and $IoU(P^{(i)}, P^{wm}) > threshold$ **then**
5: $boxes^{(i)} = [C^{(i)}, x_{min}^{(i)}, y_{min}^{(i)}, w^{(i)}, h^{(i)}, S^{(i)}]$
6: **end if**
7: **end for**
8: $sorted\_boxes = sorted(boxes, key = lambda\ x : x[2])$ //sorted from small to large according to the x coordinate of the upper left corner;
9: $checked\_boxes = check(sorted\_boxes)$ //the check function will be introduced in section 4.3;
10: $detect\_number = [checked\_boxes^{(i)}[1]]$ $i \in [1, length(checked\_boxes)]$;
11: $NUMBER = ListToString(detect\_number)$ //convert list to string;
12: **return** $NUMBER$;

situation is likely to occur in the case of blurred images with rotating numbers. At this time, strengthening the training corresponding to the error sample can effectively reduce the prediction error, and the second case is likely to become the first case. The third case means there are misses, If it is a blurred image that is unrecognizable to the human eye, this situation can be ignored. If not, a simple way is to lower the threshold and perform the YOLOv3 detection again, which can add some new predicted bounding boxes, but the time cost increases. One feasible way is to increase the number of unpredicted digital samples to join the training. We focus on applying the rules to solve the first case, which is also the case with the most exceptions.

The digital rotation changes have 36 cases like xxx09-xxx10, xx099-xx100, x0999-x1000, 09999-10000. We strictly label the digital appearing in the digital

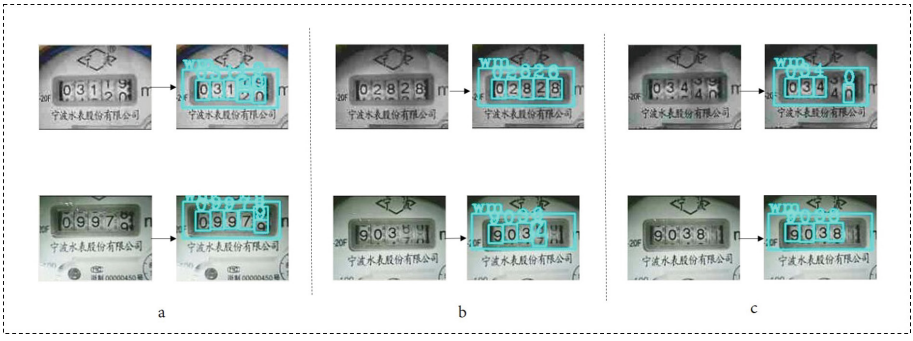


Fig. 6. Three main situations with different number of predicted bounding boxes

area of the water meter when marking, so almost all of the following 36 cases can predict more than five bounding boxes except 'wm' category, and in most cases, the predicted bounding boxes fit the digital well. Since the last step of YOLOv3 has added non-maximum suppression, we do not need to consider the case where the two predicted bounding boxes have a large overlap. We first find out if two or more predicted bounding boxes appear in the same vertical area, which is the so-called half-word problem, then suppress this situation, and finally ensure that there is only one predicted bounding box for each vertical area. We propose a suppression strategy (Algorithm 2): finding the closest two boxes of X_{min} in each loop, then proposing an evaluation function value_func(), comparing the scores of the two boxes found, and suppressing the predicted box with a lower score, simultaneously recording whether the above or below box is reserved. The subsequent loop only needs to compare the y_{min} of the two boxes, keeping the same as the previous record. Exit the loop until the number of predicted bounding boxes is less than or equal to five. For the evaluation function value_func(), we propose three scheme comparisons, which are comparing the height of the predicted bounding box, the score, and the combination of height normalization and score. See Eqs. 3, 4 and 5 for details. box_i is the i_{th} box, $height_{box_i}$ is the height of i_{th} box, and $score_{box_i}$ is the score of i_{th} box.

Algorithm 2. check() function

Require: sorted boxes SBOXES
Ensure: checked boxes CBOXES
1: $flag \leftarrow -1$;
2: **while** $(length(SBOXES) - 5) \neq 0$ **do**
3: Find the box_{idx} and box_{idx+1} of the current x_{min} minimum difference in SBOXES;
4: **if** $flag = -1$ **then**
5: **if** $value\_func(box_{idx}, box_{idx+1}) > value\_func(box_{idx+1}, box_{idx})$ **then**
6: Remove box_{idx+1} from SBOXES;
7: **else**
8: Remove box_{idx} from SBOXES;
9: **end if**
10: Record flag is 1 or 0, corresponding to whether the reserved box is above or below;
11: **else**
12: Compare the y_{min} of box_{idx+1} and box_{idx}, and keep the box consistent with the flag;
13: **end if**
14: **end while**
15: $CBOXES_i \leftarrow SBOXES_i$ $i \in [1, 5]$;
16: **return** $CBOXES$;

$$value\_func(box_i, box_j)_1 = height_{box_i} \tag{2}$$

$$value\_func(box_i, box_j)_2 = score_{box_i} \tag{3}$$

$$value\_func(box_i, box_j)_3 = \lambda * \frac{height_{box_i}}{height_{box_i} + height_{box_j}} + (1 - \lambda) * score_{box_i} \quad (4)$$

5 Experiments

In this section, we compare the performance of the traditional version with the version of YOLOv3 combined with the rules on the test set. In order to compare the three evaluation functions, we also prepared 3,000 more fuzzy water meter images for evaluation.

5.1 Experimental Settings

The traditional version uses OpenCV's[2] own contour algorithm to extract digital regions as positive samples, and other regions as negative samples, training SVM classifiers. The recognition part uses a simple convolutional network of three convolutional layers and two fully connected layers, using maximum pooling and dropout. We prepared 10, 510 water meter images that have been labeled, divided into training and test sets in a ratio of 8:2 after data cleaning, and use the official YOLOv3-voc network structure to train, modify the number of categories to 11, and add random transformations. All of our experiments are on Intel Core i7 8700K, 16G Memory, 1T HDD, Ubuntu 16.04, a GeForce GTX 1080Ti graphics card with 11G memory.

5.2 Comparison Results

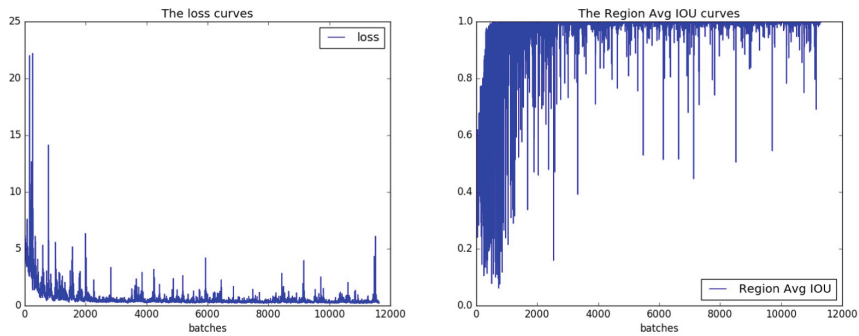


Fig. 7. Training loss and average IoU

Figures 7 and 8 show that training 10,000 batches basically leads to converge, and the 11 categories of training have achieved super-high AP, and the mAP of 11 categories is up to 0.9893.

[2] https://opencv.org/.

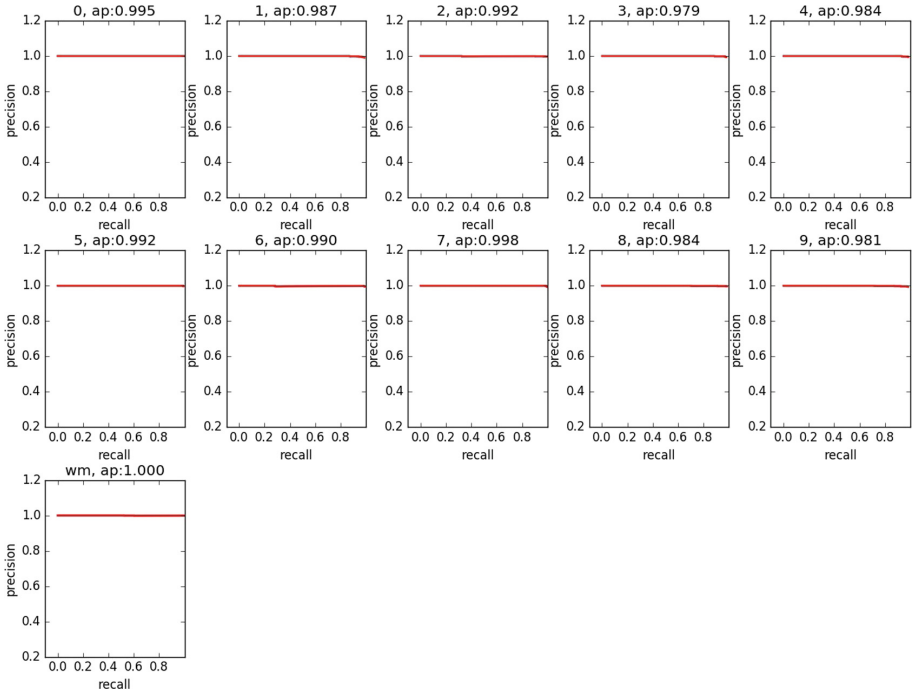


Fig. 8. PR curve for each category

On 2102 test sets, it can be found that the proposed method improves the 0-error accuracy by about 11.2% and the 1-error accuracy by about 4%. With 3000 blurred images as test sets, the performance gap is even greater. The proposed method improves the 0-error accuracy by about 26.8% and the 1-error accuracy by about 12.1%. The speed of the traditional method is about 100 ms/image, and our proposed method takes about 30 ms/image and 450 ms/images respectively under GPU and CPU. We also trained a tiny structure, with a 1-error of 99.57% on test2012 and a CPU time of 150 ms. Taking actual product demand into account, customers often tend to accept an error of 1 cubic meter, so the accuracy of 1-error may be more important. Based on the performance of the evaluation function on the two test sets, we can find that the combination of height normalization and score performs better (Tables 1 and 2).

Table 1. Accuracy on two types of test sets.

Method	Test 2102		Test 3000	
	0-error	1-error	0-error	1-error
Traditional method	0.8692	0.9600	0.7500	0.8877
Proposed method	**0.9667**	**0.9981**	**0.9570**	**0.9953**

Table 2. Accuracy in three evaluation functions.

Value function	Test 2102		Test 3000	
	0-error	1-error	0-error	1-error
Function(1)	0.9662	0.9981	0.9667	0.9940
Function(2)	0.9577	0.9972	0.9433	0.9950
Function(3), $\lambda = 0.65$	**0.9667**	**0.9981**	0.9570	**0.9953**

6 Conclusion

In this paper, we have established a water meter image dataset for training our model, as well as a novel water meter digital recognition method to tackle the recognition problem as a detection task. In contrast to traditional approaches, our work gets rid of time-consuming feature design thanks to the deep learning technology, instead it is a simple pipeline that directly receives images as input data and detect the location of ROIs, as well as classification. Detailed experiments evidence the benefit of our YOLOv3-based framework, it is a really accurate and real-time system, which has met the commercial standard. In particular, due to the wide application of water meter in both industries and our daily life, our water meter recognition work is of great practicability.

References

1. Redmon, J., Ali, F.: You only look once: unified, real-time object detection. IEEE (2015)
2. Redmon, J., Ali, F.: YOLO9000: better, faster, stronger. IEEE (2017)
3. Redmon, J., Ali, F.: YOLOv3: an incremental improvement. IEEE (2018)
4. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. IEEE (1998)
5. Matan, O., Burges, C.J.C., LeCun, Y., Denker, J.S.: Multi-digit recognition using a space displacement neural network. In: NIPS (1991)
6. Guo, Q., Lei, J., Tu, D., Li, G.: Reading numbers in natural scene images with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014)
7. Wang, T.C., Liu, M.Y., Zhu, J.Y.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
8. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision (2016)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. IEEE (2005)
10. Vapnik, V.N.: An overview of statistical learning theory. IEEE (1999)
11. Netzer, Y., et al.: Reading digits in natural images with unsupervised feature learning. In: NIPS (2011)
12. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M.: Generative adversarial nets. In: NIPS (2014)
13. Jain, A.K.: Data clustering: 50 years beyond K-means. In: ECML/PKDD (2008)