4th Asia Pacific Conference of the Prognostics and Health Management,
Tokyo, Japan, September 11 – 14, 2023 R02-08

Automatic Generation of Seven-Segment Display Image for Machine-Learning-Based Digital Meter Reading

Kota Gushima[1], and Takahiro Kashima[2]

[1,2]*Information Technology R&D Center, Mitsubishi Electric Corporation*

5-1-1 Ofuna, Kamakura, Kanagawa 247-8501, Japan.

Gushima.Kota@df.MitsubishiElectric.co.jp
Kashima.Takahiro@dc.MitsubishiElectric.co.jp

ABSTRACT

This paper presents a novel approach for automating the reading of seven-segment displays using machine learning, specifically addressing the concern of acquiring training data. By developing an algorithm that can automatically generate training images, the need for seven-segment display digit image acquisition was significantly reduced, making the process more efficient and cost-effective. In addition, by automatically generating images, a large amount of training data can be acquired. The training images include noise, such as sunlight reflections, shadows, and blurring due to camera shaking. The proposed method employs a machine-learning model trained on a diverse dataset of synthetic images generated by an algorithm. This dataset includes various fonts and styles, enabling the model to predict the meter values displayed on various fonts of the seven-segment liquid crystal display. By leveraging this auto-generated image set, the model effectively eliminates the labor-intensive process of manually capturing and annotating real-world meter images. The experimental results demonstrated the effectiveness of the proposed approach, with a reading accuracy of 96.8%.

1. INTRODUCTION

The Mitsubishi Electric Building Solutions Corporation (MEBS) was used to improve the quality of meter-reading operations and shorten the work time required. We developed a system capable of automatically reading meter information from smartphone camera images to reduce work errors and time. In this study, we aimed to improve the system's accuracy using machine learning for the automatic reading of meter values. The meters used in this study were M1PM-R, M2PM-R, and M8FM-S1R, manufactured by MITSUBISHI ELECTRIC. These digital meters display values on a seven-segment liquid crystal display (LCD) with decimal points. To develop an automatic reading system, we aimed to include a feature that realizes the function of automatically reading the meter values of the seven-segment LCD.

In the field of building maintenance, low-specification devices, such as iPhone7, are still in use. Furthermore, there are constraints such as the inability to use server-client type applications due to meter reading operations in areas where radio waves do not reach. Therefore, it is necessary to develop a system that can operate even on low-specification devices.

In this study, we propose an algorithm that automatically generates training data, which is a digit image displayed on a seven-segment LCD. We built a light digit estimation model and a light decimal point estimation model, learning from automatically generated training image data, and discussed the results of our evaluation, focusing on the system's accuracy. Chapter 2 presents related work. Section 3 introduces the proposed method, Chapter 4 explains the algorithm that automatically generates training data, Chapter 5 details the training process for the estimation models, Chapter 6 reports the evaluation results and discusses the findings, and Chapter 7 concludes the study.

2. RELATED WORK

Recently, image recognition accuracy using deep learning has advanced and been applied to reading dial-type analog meters (Salomon et al., 2020, Liu et al., 2020) and gas meters (Vanetti, 2013). It was similarly applied to the Seven-segment LCD value estimations. In deep learning, models are trained using a large amount of data, enabling them to recognize data features accurately. This technique has also been applied to digit and character estimation represented in images, often using a dataset of labeled handwritten number images from the MNIST database to learn digits. (He, 2016). However, because MNIST is a dataset of handwritten digits, it is not suitable for seven-segment LCD value estimation, which is the subject of this study. Machine-learning methods specialized for Seven-segment LCD value estimation were also investigated. For example, Wannachai et al. (2020) used

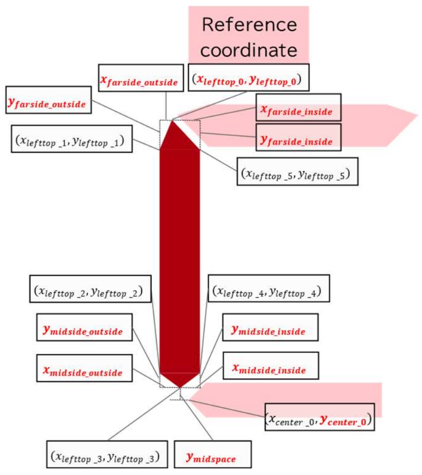

Reference coordinate
$x_{farside\_outside}$
$(x_{lefttop\_0}, y_{lefttop\_0})$
$y_{farside\_outside}$
$x_{farside\_inside}$
$(x_{lefttop\_1}, y_{lefttop\_1})$
$y_{farside\_inside}$
$(x_{lefttop\_5}, y_{lefttop\_5})$
$(x_{lefttop\_2}, y_{lefttop\_2})$
$(x_{lefttop\_4}, y_{lefttop\_4})$
$y_{midside\_outside}$
$y_{midside\_inside}$
$x_{midside\_outside}$
$x_{midside\_inside}$
$(x_{center\_0}, y_{center\_0})$
$(x_{lefttop\_3}, y_{lefttop\_3})$
$y_{midspace}$

Figure 3.1 each coordinate of the upper-left vertical segment.

a convolutional neural network (CNN), a machine-learning method, to train 537 digits images for digit estimation. They reported that they could estimate the values with 91.1% accuracy. Imran et al. (2022) used the YOLO v3 framework to achieve 99% accuracy in recognizing the meter position and 77% accuracy in predicting values with seven-segment LCD digits in fieldwork. The previous studies ignored the function of decimal point location estimation.

In this study, we aimed to develop a system capable of predicting the position of each digit and decimal point of the target meters. The previous studies ignored the function of decimal point location estimation. In addition, previous studies did not mention how to acquire Seven-segment LCD images for training, which are costly to capture in the field. In this study, we aimed to develop a system capable of predicting the position of each digit and decimal point of the target meters. We propose an algorithm that automatically generates images to reduce the cost of acquiring training data and build a digit estimation model and a decimal point position estimation model with the generated images.

3. Algorithm Automatically Generating Seven-Segment LCD Images

Seven-segment LCD digits are designed in styles (fonts) based on the width, aspect ratio, and tilt, which are set independently by various manufacturers. Therefore, we propose an algorithm for generating images of various styles. The process for automatically generating a seven-segment LCD image is as follows. For convenience, the left-right axis is referred to as the x-axis, and the up-down axis is the y-axis. The coordinates in the upper-left corner of the image were considered as the origin, with the rightward orientation as the positive x-axis orientation and the downward orientation as the positive y-axis orientation. The following seven steps were used to calculate the vertices of each segment.

1. *Determination of each coordinate of the upper-left vertical segment (Figure 3.1)*

- The x- and y-coordinates $(x_{\mathrm{lefttop\_0}}, y_{\mathrm{lefttop\_0}})$ of the top vertex are specified as the starting points of the entire segment.

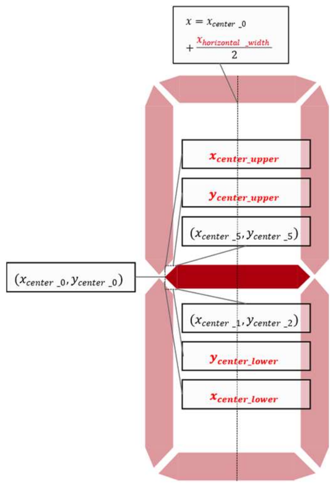

$x = x_{center\_0} + \frac{x_{horizontal\_width}}{2}$
x_{center_upper}
y_{center_upper}
$(x_{center\_5}, y_{center\_5})$
$(x_{center\_0}, y_{center\_0})$
$(x_{center\_1}, y_{center\_2})$
y_{center_lower}
x_{center_lower}

Figure 3.2 Coordinates of the upper horizontal segment.

- The distances in each axis direction from the starting point to the outer vertex, the inner vertex, and the $(x_{\mathrm{lefttop\_1}}, y_{\mathrm{lefttop\_1}})$ and $(x_{\mathrm{lefttop\_5}}, y_{\mathrm{lefttop\_5}})$ coordinates are specified.

$$\boldsymbol{x}_{\mathbf{lefttop\_1}} = \boldsymbol{x}_{\mathbf{lefttop\_0}} - \boldsymbol{x}_{\mathbf{farside\_outside}} \quad (1)$$
$$\boldsymbol{y}_{\mathbf{lefttop\_1}} = \boldsymbol{y}_{\mathbf{lefttop\_0}} + \boldsymbol{y}_{\mathbf{farside\_outside}} \quad (2)$$
$$\boldsymbol{x}_{\mathbf{lefttop\_5}} = \boldsymbol{x}_{\mathbf{lefttop\_0}} + \boldsymbol{x}_{\mathbf{farside\_inside}} \quad (3)$$
$$\boldsymbol{y}_{\mathbf{lefttop\_5}} = \boldsymbol{y}_{\boldsymbol{lefttop\_0}} + \boldsymbol{y}_{\mathbf{farside\_inside}} \quad (4)$$

- The vertices of the common x-coordinate are assigned.

$$\boldsymbol{x}_{\mathbf{lefttop\_2}} = \boldsymbol{x}_{\mathbf{lefttop\_1}} \quad (5)$$
$$\boldsymbol{x}_{\mathbf{lefttop\_4}} = \boldsymbol{x}_{\mathbf{lefttop\_5}} \quad (6)$$

- $y_{lefttop\_3}$ is calculated by specifying y_{midspace}, which is the distance along the y-axis between the coordinates of the leftmost vertex of the central horizontal segment and the bottom vertex of the upper-left vertical segment.

$$\boldsymbol{y}_{\mathbf{lefttop\_3}} = \boldsymbol{y}_{\mathbf{center\_0}} - \boldsymbol{y}_{\mathbf{midspace}} \quad (7)$$

- $x_{\mathrm{lefttop\_3}}, y_{\mathrm{lefttop-2}}, y_{\mathrm{lefttop-4}}$ are calculated by specifying the width from the bottom vertex to the outer vertex and the height to the inner vertex.

$$\boldsymbol{x}_{\mathbf{lefttop\_3}} = \boldsymbol{x}_{\mathbf{lefttop\_2}} + \boldsymbol{x}_{\boldsymbol{midside}\mathbf{\_outside}} \quad (8)$$
$$\boldsymbol{y}_{\mathbf{lefttop\_2}} = \boldsymbol{y}_{\mathbf{lefttop\_3}} - \boldsymbol{y}_{\mathbf{midside\_outside}} \quad (9)$$
$$\boldsymbol{y}_{\mathbf{lefttop\_4}} = \boldsymbol{y}_{\mathbf{lefttop\_3}} - \boldsymbol{y}_{\mathbf{midside\_inside}} \quad (10)$$

2. *Determination of the coordinates of the lower-left vertical segment*

- Each coordinate of the lower-left vertical segment is calculated such that the coordinates of the upper-left vertical segment are line-symmetrical, with the line $y = y_{\mathrm{center\_0}}$ as the target axis.

3. *Determination of the coordinates of the central horizontal segment (Figure 3.2)*

- The distance in each axis direction from the leftmost coordinate $(x_{\text{center\_0}}, y_{\text{center\_0}})$ to the upper coordinate and lower coordinates are specified, and the coordinates are calculated.

$$\boldsymbol{x}_{\textbf{center\_1}} = \boldsymbol{x}_{\textbf{center\_0}} + \boldsymbol{x}_{\textbf{center\_lower}} \quad (11)$$
$$\boldsymbol{y}_{\textbf{center\_1}} = \boldsymbol{y}_{\textbf{center\_0}} + \boldsymbol{y}_{\textbf{center\_lower}} \quad (12)$$
$$\boldsymbol{x}_{\textbf{center\_5}} = \boldsymbol{x}_{\textbf{center\_0}} + \boldsymbol{x}_{\textbf{center\_upper}} \quad (13)$$
$$\boldsymbol{y}_{\textbf{center\_5}} = \boldsymbol{y}_{\textbf{center\_0}} + \boldsymbol{y}_{\textbf{center\_upper}} \quad (14)$$

- The width of the center horizontal segment is specified and $(x_{\text{center\_2}}, y_{\text{center\_2}})$, $(x_{\text{center\_3}}, y_{\text{center\_3}})$, $(x_{\text{center\_4}}, y_{\text{center\_4}})$ are calculated such that these coordinates are line symmetrical, with the line $x = x_{\text{center\_0}} + \frac{x_{\text{horizontal\_width}}}{2}$ as the target axis based on the coordinates on the left side.

4. *<u>Determination of the coordinates of each of the upper right and lower right vertical segments</u>*

The coordinates of each of the upper-right and lower-right vertical segments are calculated such that the straight line $x = x_{\text{center\_0}} + \frac{x_{\text{horizontal\_width}}}{2}$ is line-symmetrical with the target axis based on the coordinates of the upper-left and lower-left vertical segments.

5. *<u>Determination of the coordinates of each of the upper horizontal segments (Figure 3.3)</u>*

- The distances x_{farspace}, y_{farspace}, $x_{\text{farmid\_outside}}$, $y_{\text{farmid\_outside}}$ from each coordinate system are specified to determine the location of the upper segment.

$$\boldsymbol{x}_{\textbf{top\_0}} = \boldsymbol{x}_{\textbf{\textit{lefttop}\_0}} + \boldsymbol{x}_{\textbf{farspace}} \quad (15)$$
$$\boldsymbol{y}_{\textbf{top\_0}} = \boldsymbol{y}_{\textbf{lefttop\_0}} - \boldsymbol{y}_{\textbf{farspace}} \quad (16)$$
$$\boldsymbol{x}_{\textbf{top\_1}} = \boldsymbol{x}_{\textbf{top\_0}} + \boldsymbol{x}_{\textbf{farmid\_inside}} \quad (17)$$
$$\boldsymbol{y}_{\textbf{top\_1}} = \boldsymbol{y}_{\textbf{top\_0}} + \boldsymbol{y}_{\textbf{farmid\_inside}} \quad (18)$$
$$\boldsymbol{x}_{\textbf{top\_5}} = \boldsymbol{x}_{\textbf{top\_0}} + \boldsymbol{x}_{\textbf{farmid\_outside}} \quad (19)$$
$$\boldsymbol{y}_{\textbf{top\_5}} = \boldsymbol{y}_{\textbf{top\_0}} - \boldsymbol{y}_{\textbf{farmid\_outside}} \quad (20)$$

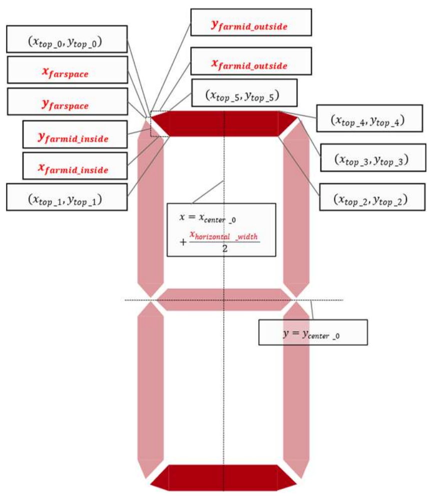
- The three coordinates on the right side are calculated such that the line $x = x_{\text{center\_0}} + \frac{x_{\text{horizontal\_width}}}{2}$ is symmetrical to the target axis based on each vertex on the left side, as shown in Step 4.

6. *<u>Calculating the lower horizontal segment in the same way as the upper horizontal segment</u>*

- Each coordinate of the lower horizontal segment is calculated as a line target on the axis of the straight line$y = y_{\text{center_0}}$ based on the upper horizontal segment.

7. *<u>Add tilt to the entire image</u>*

- By specifying the tilt value and using $(x_{\text{lefttop\_0}}, y_{\text{lefttop\_0}})$ as the origin, the x coordinates of each coordinate $(x_{\text{origin}}, y_{\text{origin}})$ are updated according to the tilt. Note that $(x_{\text{origin}}, y_{\text{origin}})$ is the coordinate before adding the tilt, and x_{tilt} is the x-coordinate after the tilt.



(x_{top_0}, y_{top_0})
$x_{farspace}$
$y_{farspace}$
y_{farmid_inside}
x_{farmid_inside}
(x_{top_1}, y_{top_1})
$y_{farmid\_outside}$
$x_{farmid\_outside}$
(x_{top_5}, y_{top_5})
(x_{top_4}, y_{top_4})
(x_{top_3}, y_{top_3})
(x_{top_2}, y_{top_2})
$x = x_{center\_0} + \frac{x_{horizontal\_width}}{2}$
$y = y_{center\_0}$

$$x_{\text{tilt}} = \frac{y_{\text{lefttop\_0}} - y_{\text{origin}}}{\text{tilt}} + x_{\text{origin}} \quad (21)$$

Figure 3.3 Coordinates of the upper horizontal segment.

These seven steps were used to calculate the coordinates of each segment. Generating seven-segment LCD images from 0 to 9 by filling a specific segment based on the coordinate information generated using this method is possible. Additionally, by varying the specified parameters, various styles of seven-segment characters can be generated.

4. Automatic Image Generation Flow of Training Image Data

The method described in Chapter 3 allows for seven-segment LCD image generation; however, the training image should contain noise, such as sunlight reflections, shadows, and blurring due to camera shaking. It should also include an adjacent digit to build noise tolerance estimation models. This section explains the procedure for generating the digit images, including noise. This generation program was implemented using Python 3.9 and OpenCV4.5.5.

4.1. Specifying parameters for image generation

This section describes the specifications of the parameters used for digit image generation, as mentioned in Chapter 3. In this study, we determined a range of parameter values and implemented the program such that each parameter was determined randomly from that range because generating a complete set for all parameter patterns would cause an enormous number of images, and the computational resources for training would be immense. Table 4.1 indicates the ranges for each parameter.

The system does not require RGB color information to predict the seven-segment LCD value; therefore, the training data were generated as grayscale images. This method uses a grayscale image with 256-pixel values; higher

Table 4.1. Image generation parameters

$18 \le x_{\mathrm{lefttop\_0}} \le 35$
$0 \le y_{\mathrm{lefttop\_0}} \le 45$
$4 \le x_{\mathrm{far\_outside}} \le 7$
$4 \le y_{\mathrm{far\_outside}} \le 7$
$4 \le x_{\mathrm{far\_inside}} \le 7$
$4 \le y_{\mathrm{far\_inside}} \le 7$
$y_{\mathrm{lefttop\_0}} + 65 \le y_{\mathrm{center}_0} \le y_{\mathrm{lefttop}_0} + 70$
$1 \le x_{\mathrm{midspace}} \le 2$
$8 \le y_{\mathrm{midspace}} \le 10$
$4 \le x_{\mathrm{central\_outside}} \le 7$
$4 \le y_{\mathrm{central\_outside}} \le 7$
$4 \le x_{\mathrm{central\_inside}} \le 7$
$4 \le y_{\mathrm{central\_inside}} \le 7$
$45 \le x_{\mathrm{horizontal\_width}} \le 53$
$4 \le x_{\mathrm{farspace}} \le 7$
$4 \le y_{\mathrm{farspace}} \le 7$
$4 \le x_{\mathrm{farmid\_outside}} \le 7$
$4 \le y_{\mathrm{farmid\_outside}} \le 7$
$4 \le x_{\mathrm{center\_lower}} \le 7$
$4 \le y_{\mathrm{center\_lower}} \le 7$
$4 \le x_{\mathrm{center\_upper}} \le 7$
$4 \le y_{\mathrm{center\_upper}} \le 7$
$5 \le \mathrm{tilt} \le 15$

values indicate brighter colors. The background color was randomly set from 50 to 180. The character color is obtained by subtracting the value between 20 and 100 from the background color value. Note that pixel values less than zero (e.g., 50 for the background color and 100 for the text color) should be replaced with zero.

The size of the generated images was set such that the images were 180 pixels in height and 90 pixels in width. This aspect ratio is close to that of a digit image from an actual image. Figure 4.1 (a) indicates the digit image generated using the method described in this section.

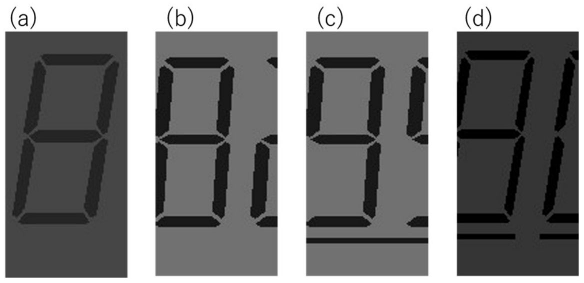


Figure 4.1. (a): the generated digit image. b: Training data for "8" with "2" included. c: Underline connected training data. d: Training data with disconnected underlines

4.2. Adding the adjacent digit and underlines

The adjacent digit image generated by adding or subtracting the $x_{\mathrm{lefttop\_0}}$ value specified in Table 4.1 is superimposed on the generated image to produce an image including an adjacent digit. Figure 4.1 (b) shows a digit image containing an adjacent digit.

Some seven-segment LCD meters use underlines added to the LCD to improve readability, including M8FM-S1R. Underlining sometimes results in low estimation accuracy when neglected in preliminary trials. The training images were superimposed on the underlines as noise to build noise-tolerant models.

The underlines are drawn with reference to the M8FM-S1R meter image with a thickness of three pixels, starting five pixels away from the y-coordinate of the bottom coordinates of the lower horizontal segment. In addition, the underlines in M8FM-S1R are separated at the decimal point but continuous at other locations. Therefore, in addition to images without underlines, training data with connected underlines (Figure 4.1 (c)) and training data with broken underlines (Figure 4.1 (d)) were generated. The images containing underline(s) were generated such that the ratio of connected underline images to images with broken underlines to those without was 1:1:8.

4.3. Superimposing light reflections, shadows, and blurring

In outdoor locations, sunlight reflections may be included in the image, and shadows may appear depending on lighting conditions. Therefore, we generated training images containing noise by superimposing them onto the original image. Because shadows and reflections often appear horizontally, these extensions were added to the current training data. Specifically, reflections and shadows are represented simultaneously by adding or subtracting pixel values in specific areas within the generated image. The pixel values were specified by uniform random numbers of 30, 50,

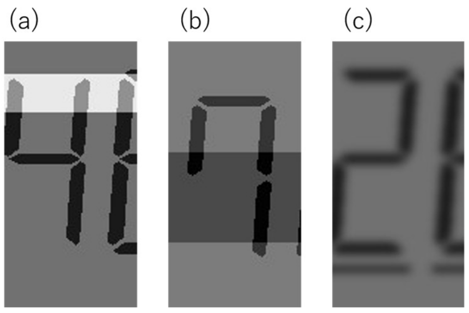


Figure 4.2. (a): A digit image with sunlight reflection. (b): A digit image with shadow. (c): A digit image with Gaussian blur.

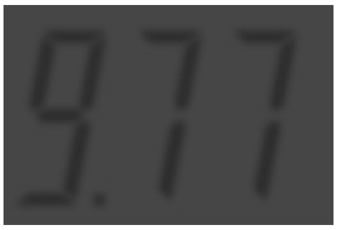


Figure 4.3. Multiple-digit image including decimal point with Gaussian blur.

90, and 120. The images containing these noises were generated such that the ratio of images with light reflection to the images with shadows to those without noise was 1:1:9.

Additionally, some images captured in the field were blurred due to hand shaking or low-light environments. Therefore, we generated training images with Gaussian blur using the Gaussian blur function in the OpenCV library. The kernel size of the Gaussian blurring was chosen by a uniform random number between 15, 25, and 45, and the standard deviation was specified as three. Superimposed blurred images were generated in a manner such that the ratio of the number of images with blur to the number of images without noise was 1:2 because some were taken in dark locations. Our preliminary study showed that the estimation rate was lower for darker images.

4.4. Generating multiple-digit images with a decimal point

Multiple-digit images were used to build the decimal point position estimation model, including decimal points for the training images. Our method for generating digit images can also generate multiple-digit images, including decimal points. In this study, the program generates 3-digits images because the target meters displayed values up to the second decimal place. Therefore, the generated images were divided into three categories: 1) values containing a point in the second decimal place, 2) values containing a point in the first decimal place, and 3) values without a decimal point. As shown in Figure 4.3, these training images include underlines, reflection, shadow, and blur, as mentioned in Sections 4.2 and 4.3.

5. Training the Estimation Model by Generated Images

We implemented the program in building the two models using Python 3.9 and ResNet50V2 in Keras, a neural network library. To lighten the estimation model, the size of the images input into the model was reduced to 64*64 pixels. As the target of this study was seven-segment LCD, it was believed that reducing the size of the input images did not significantly decrease accuracy.

5.1. Training the digit estimation model

The digit estimation model outputs are a predicted integer value from zero to nine and the confidence level when an image is used as the input. This model was trained with 225,337 images as training data and 32,139 images as validation data using ResNetV2. The reading system crops the input images from the captured photographs into images of each digit as a pre-processing step.

5.2. Training the decimal point position estimation model

The decimal point position estimation model outputs a number indicating the location of the decimal point and a confidence value when an image is an input. This model was trained with 416,864 images as training data and 257,599 images as validation data using ResNetV2. The reading system crops the input images from the photograph into a 3-digit image as a pre-processing step.

6. Evaluation

We evaluated the accuracy of the two models using video frame images captured at MEBS and in an actual field. The test images for the estimation models were selected from frame images that matched the template images. An Overview of these results is presented in Table 6.1.

6.1. Evaluation of each model

The digit estimation model inputs each digit image and outputs the predicted value and confidence, after which the frame images are cropped into test images of each digit. To evaluate the decimal point position model, the pre-processing program cropped the frame images into test 3-digit images, and the model input and output the number indicating the location. In this study, we decided that the images with a confidence value below 0.9 were non-predictable, and those with a confidence value above 0.9 were predictable. There were 415 test images; the target meters were M1PM-R, M2PM-R, and M8FM-S1R.

As a result of the digit estimation model, the percentage of predictable digit images was 86.1%, and the accuracy was 96.8%. As a result of the decimal point position estimation model, the percentage of predictable digit images was 93.0%, and the accuracy was 94.3%.

Table 6.1. An Overview of estimation result

	Digit	Decimal point position	Meter value reading	Same value twice in a row
Predictability percentage	86.1%	93.0%	77.8%	-
Accuracy	96.8%	94.3%	85.2%	95.0%

6.2. Evaluation of the whole meter value

By combining the two estimation models, we evaluated the accuracy of the entire meter value, including all digit and decimal point positions. The number of test images was 409, and the target meters were M1PM-R and M8FM-S1R. We decided that all confidence above 0.9 were predictable and the others were non-predictable.

The percentage of the predicted images was 77.8%, and the accuracy was 85.2%. The models consecutively output the same values because the test images were originally video frames. Therefore, we decided on a method in which the estimation was successful when the output values were the same twice in a row, and the accuracy rate was 95.0% by applying this method.

7. Discussion

An accuracy rate of 95.0% was achieved by operating the same output meter values twice in a row. However, in this evaluation, the accuracy rate was increased by excluding unpredictable images. The percentage of correct values output from the entire frame image matching the template data can be calculated using the following formula:

$$0.778 * 0.852 \approx 0.662 = 66.2\%$$

Although highly accurate results can be obtained from this system, it requires a large number of input images to produce reliable results. In the proposed method, the system must input at least two predictable images. In the evaluation, the percentage of predicted images was 77.8%; however, the percentage of predictable images depended on the environment, including light conditions. If the percentage is lower than the evaluation, the system waits for predictable images to be captured. Consequently, the responsiveness and usability of the system may deteriorate.

There are two ways to increase the system's responsiveness: the first is to improve the accuracy of each estimation model and increase confidence. By tuning the hyperparameters and reviewing the training data, we sought methods to train the models to output higher confidence levels. Second, the processing speed should be enhanced to increase the number of images processed per second. Our system's current processing time is 0.43 seconds per image; therefore, our system can output a result in as little as 0.86 seconds because it requires two images. To speed up the processing time, it is necessary to shorten the time required for the entire system, including the time required for pre-processing the captured images and communication.

8. Conclusion

In this paper, we propose a seven-segment LCD training data generation method and a meter reading method using two estimation models–the digit estimation model and the decimal point position estimation model–learned from the generated data. The generated training data contain underlined noise, flash reflection, shadow, and Gaussian blur to improve the accuracy of the models. In addition, the system outputs the estimation results when the output values are the same twice in a row, and the proposed method achieves an accuracy rate of 95.0%. In future work, we aim to build a more accurate model using the generated training images and improve the system's responsiveness. Additionally, a constraint in this is that evaluations are only being conducted on three types of meters. In the future, if we develop a new model trained on a wider variety of seven-segment LCD fonts, there remains a possibility that a lightweight model may not be able to handle it.

Reference

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778. doi: 10.1109/CVPR.2016.90

Wannachai, A., Boonyung, W., & Champrasert, P. (2020). Real-Time Seven Segment Display Detection and Recognition Online System Using CNN. In Y. Chen, T. Nakano, L. Lin, M. U. Mahfuz, & W. Guo (Eds.), Bio-inspired Information and Communication Technologies Cham: Springer International Publishing, pp. 52-67. doi: 978-3-030-57115-3

Salomon, G., Laroca, R., & Menotti, D. (2020). Deep Learning for Image-based Automatic Dial Meter Reading: Dataset and Baselines. Proceedings of 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1-8. doi: 10.1109/IJCNN48605.2020.9207318

Zheng, W., Yin, H., Wang, A., Fu, P., & Liu, B. (2017). Development of an automatic reading method and software for pointer instruments. Proceedings of 2017 First International Conference on Electronics Instrumentation & Information Systems (EIIS), pp. 1-6. doi: 10.1109/EIIS.2017.8298626

Liu, Y., Liu, J., & Ke, Y. (2020). A detection and recognition system of pointer meters in substations based on computer vision. Measurement, vol. 152, pp. 107333. doi: 10.1016/j.measurement.2019.107333

Vanetti, M., Gallo, I., & Nodari, A. (2013). GAS meter reading from real world images using a multi-net system. Pattern Recognition Letters, vol. 34, no. 5, pp. 519-526. doi: 10.1016/j.patrec.2012.11.014

Imran, M., Anwar, H., Tufail, M., Khan, A., Khan, M., & Ramli, D. A. (2023). Image-Based Automatic Energy Meter Reading Using Deep Learning. Computers, Materials & Continua, vol. 74, no. 1, pp. 203-216. doi: 10.32604/cmc.2023.029834