IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 22, 2022, accepted February 27, 2022, date of publication March 8, 2022, date of current version March 18, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3157706

Generative Data Augmentation for Automatic Meter Reading Using CNNs

NUNTIDA SRIPANUSKUL[1], PRAWIT BUAYAI[1], AND XIAOYANG MAO[1,2], (Member, IEEE)
[1]Faculty of Engineering, University of Yamanashi, Kofu 400-0015, Japan
[2]School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310013, China

Corresponding author: Xiaoyang Mao (mao@yamanashi.ac.jp)

ABSTRACT While smart meters are still not widely installed in many countries, automatic reading of traditional-type meters is useful from the perspective of both cost and safety. Although convolutional neural network (CNN) showed a high potential for automatic meter reading under unconstrained environment, it is facing various challenges. One is the difficulty of collecting a sufficient amount of training dataset since some digits of a meter may take a long time to update. Another challenging issue is how to recognize the transitional state between two consecutive numbers. To solve these problems, we propose a new data augmentation technique that can automatically generate annotated images of numbers, including the transitional states. By taking advantage of the state-of-the-art generative neural network model, the generated numbers resemble the local appearance of those in the original meter images. Evaluation experiments confirm that our proposed generative data augmentation techniques improve the robustness of the recognition model and achieve outstanding results when compared to the previous work.

INDEX TERMS Automatic meter reading, image data augmentation, generative neural network, object detection and recognition.

I. INTRODUCTION

Despite of the huge advantages, smart meters are still not widely installed in many countries, especially in the developing one [1]–[3]. Image-based Automatic Meter Reading (AMR), which uses computer vision technology to automatically read the traditional type meters from their captured images, is useful for reducing the labors and errors caused by manual readings [3]. AMR can be used for retrofitting traditional type meters with image processing and communication add on, to avoid replacing the already installed meters [1], [2]. It can also be adapted to other scenarios such as using with the inspection robot in a plant that has dangerous locations [4].

AMR has been studied by many researchers in the past decade. Early works use handcrafted features and need specific algorithm design and parameter adjustment for a particular kind of image degradation for recognizing meter images captured under unconstrained environment [5]–[11]. Recently, researchers started exploring the potential of deep learning-based approaches to AMR [12]–[14]. However, deep learning needs a large quantity of training data to generalize the model and yield a high classification accuracy for unseen data, while the meter datasets are usually not publicly available because the images belong to service companies [15]. Therefore, in real applications, it is usually necessary to collect and annotate sufficient training data for the meters installed in a particular environment, which is expensive in terms of both human effort and time. Especially in the case of rotating meters, as shown in Fig. 1(a), it may take a long time, such as several years, to update the significant digits, hence making it impossible to collect sufficient training data for those digits in a practically acceptable period. Laroca *et al.* [15] proposed a data augmentation technique for increasing the variation of numbers at all digits using character permutation, that is, by swapping among digits. However, because different digits of a meter may have different appearances depending on the lighting and shooting conditions for capturing the meter images, such a method may fail to generate training samples that realistically resemble the real images. Fig. 1(b) shows the examples generated with Laroca *et al.*'s method from the meter image in Fig. 1(a). These generated numbers do not have the appearance of the original number "0". Moreover, due to the limited number of digits on each meter, the variation in the generated

The associate editor coordinating the review of this manuscript and approving it for publication was Yongjie Li.

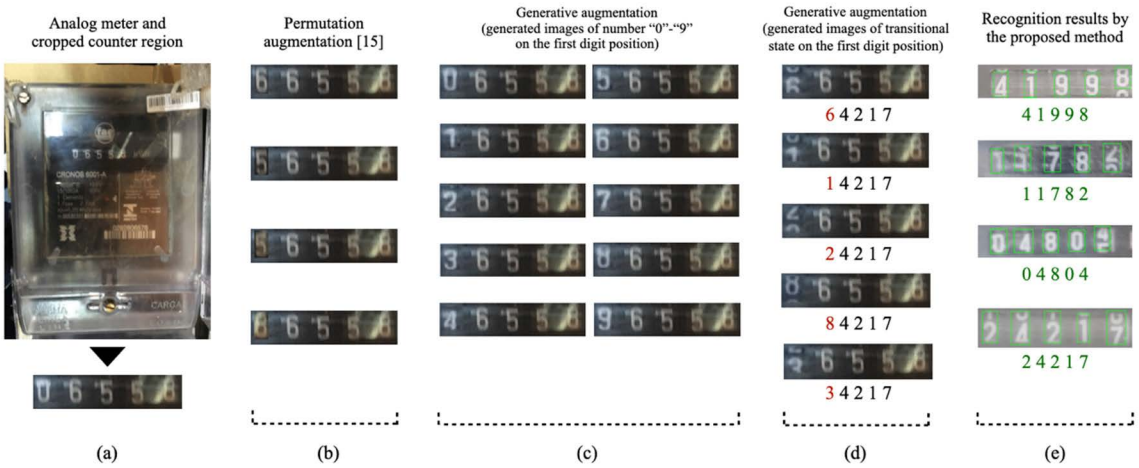

FIGURE 1. Example of augmentation for the first (most significant) digit. (a) The input meter images. (b) All possible results produced by the existing character permutation augmentation. (c) Number "0"–"9" generated by the proposed generative augmentation. (d) The images generated with the proposed method for simulating the transitional state with their label (red font). (e) The recognition results by the proposed method (green font).

number for the significant digits is also limited. Another challenging issue with the automatic reading of a rotating meter is the recognition of a state between two consecutive numbers. As the same state can look different depending on the camera's shooting angle, it is necessary to collect training data for a particular installation to achieve a high recognition accuracy. However, the manual labeling of continuing varying states between every two consecutive numbers can be highly expensive.

To challenge the above-mentioned issues, this paper proposes a novel data augmentation technique. We employ a generative deep neural network called Y-Autoencoder (Y-AE) [16] to generate the images of new numbers, along with transferring the local appearance of a meter. As shown in Fig. 1(c), the proposed method can generate a number from "1"–"9" for the first digit, resembling the appearance of the original meter image (Fig. 1(a)), from one single image (Fig. 1(a)) of number "0". To challenge the second issue, a novel method for automatically generating the annotated states between two consecutive numbers is also presented. Some annotated transitional states automatically generated with the proposed method for the first (most significant) digit are shown in Fig. 1(d). Fig. 1(e) shows some final recognition results by our automatic meter reading framework trained with the dataset including the augmented images generated by the proposed data augmentation technique.

The major contributions of this paper can be summarized as follows:

1) A novel data augmentation method called appearance preserving number generator for automatically generating annotated images of numbers resembling the local appearance of the target digits using a generative neural network.
2) A novel data augmentation method called rotating digit generator for automatically generating annotated images of transitional states between two consecutive numbers.
3) A new algorithm to create a training dataset consisting of a well-balanced amount of all numbers for each digit that can be adapted to any rotating meter image only by collecting a few images from the assumed environment.
4) Evaluation experiments using an originally designed deep learning-based AMR framework to demonstrate the effectiveness of the proposed generative data augmentation techniques by comparing with previous works.

The remainder of the paper is organized as follows: Section II reviews the related works. The generative augmentation technique is introduced in Section III. The automatic meter reading framework is presented in Section IV. The evaluation experiments and the results are described in Section V. Finally, Section VI offers our conclusions and future research directions.

II. RELATED WORKS

Pioneering works on AMR mainly rely on traditional image processing techniques combined with handcrafted features. Recently, methods that leverage the advancements of convolutional neural networks (CNNs) have been developed. In this section, we review the previous works considering three aspects related to ours: automatic meter reading, meter image data augmentation, and generative neural networks.

A. AUTOMATIC METER READING (AMR)

With early AMR implementation, many traditional image processing techniques such as color-based [5], [6] and the projection-based methods [7], [8], have been employed for locating meter area and segmentation. Some works proposed for pointer-based meter used the region growing method [9], [10] and matrix border grayscale detection

technique [11]. However, these methods can be easily affected by various factors such as blur, noise, distortions, meter font, light/shadow.

More recently, researchers started exploring the potential of deep learning-based approaches to AMR, with their impressive results on real-world scenarios and advantage to perform digit recognition in a segmentation-free way (the trained model can predict all digits simultaneously).

For counter detection, object detectors were trained to detect the counter region in meter images.S. Ren *et al.* conducted the experiments to compare Faster R-CNN [17] with other models, such as YOLO [18], RetinaNet [19], Fast R-CNN and SSD in [20]. Even though Faster R-CNN achieved the best accuracy in detecting counters against other models, its computational complexity is higher than others. Therefore, most of recent studies adopted YOLO family models for their outstanding speed and being light weighted [3], [15], [18], [21]. Our AMR framework uses YOLOv5 [22], the newest YOLO model for the counter detection.

For recognition stage, early works used a small network (3-layer artificial neural network) [12]–[14]. Then Tesseract [23], an open source OCR engine, have been widely used [6], [24], [25], though the recognition results are not quite satisfied. On the other hand, BLSTM [26], [27] and FCSRN [28] employed CNN architecture for sequence recognitions and achieved impressive recognition rates. However, their methods were evaluated with manually cropped counter images and hence its effectiveness for the real-world scenario is unclear.

Laroca *et al.* [15] designed a two-stage approach AMR that employs YOLOv4 object detector for fast counter detection. For number recognition, three different CNN-based approaches, CR-NET, Multi-Task Learning, and Convolutional Recurrent Neural Network (CRNN), were evaluated with their public dataset consisting of 2000 meter images. Recently, they have upgraded their work by introducing bigger meter dataset with 12,500 fully-annotated images [3]. Their research also introduced a preprocessing step in the AMR pipeline which rectified the corner of meter to improve the reading. Although their AMR framework trained with the large dataset captured under various environments achieved plausible recognition rate, the dataset is released to academic use only and it is usually difficult to collect dataset with well-balanced amount of images for all numbers in real applications since it may take very long time to update some digits of the meters. To solve such problem, we propose a novel technique to automatically generate any desired amount of high-quality training data just from a very small number of images collected in the assumed capturing environment. The proposed data augmentation method makes it possible to apply the state-of-the-art deep learning-based AMR technologies to real applications with ease.

One challenging issue of deep learning-based approaches for meter recognition is how to improve the accuracy when reading the transitional state between two consecutive numbers. Several experiments [5], [15], [26] reported that errors mainly occur with such situations. Although, some researchers tried to address this problem by considering the transitional states as separate classes [28], no particular solution has been given to this problem since it is difficult to collect and annotate such images.

We solve the transitional state digit problem by proposing a novel technique that can generate annotated transitional state images in a fully automatic way. To validate the effectiveness of the proposed data augmentation techniques, we also implemented a deep learning-based AMR framework by making an extension to YOLOv5 [22], which is the most recently version of YOLO detection model, for counter detection and using CR-NET [29], which achieved an impressive recognition rate in [15], for the recognition stage.

B. CHARACTER AUGMENTATION

In the image-based AMR literature, even though impressive accuracy was achieved in several CNN approaches, many models were trained with private datasets [6], [26], [30]. A few meter datasets are available to the public, but either limited to academic purpose [3] or with a small size [25]. In a real application, it is usually necessary to collect and annotate sufficient training data for the meters installed in a particular environment, which is expensive in terms of both human effort and time.

Data augmentation is the technology of deriving new data from existing data. Many traditional data augmentation approaches, such as translation, rotation, and zoom in/out, have been implemented for AMR [31]. These methods, however, cannot increase the variety of characters or numbers, which is required to address the problem of the lack of a sufficient variation in numbers for particular digits, such as the most significant digit of rotating meters. To the best of our knowledge, such a problem was only addressed in [15]. They proposed a technique called character permutation, which was originally proposed for license plate augmentation [32], to obtain a balanced amount for all letters. The method controls the frequency of each character by replacing a letter with a high probability of occurrence with a letter with a low probability of occurrence in the same meter images. Although promising results have been reported in [15], this method cannot be applied to meter images in which the appearance varies across different digits, as with that shown in Fig. 1(a). Moreover, due to the limited number of digits on each meter, the variation in the generated number is also limited.

Beside the meter recognition, data augmentation has been studied for other digit recognition scenarios. V. Kukreja *et al.* [33] proposed the license plate augmentation method to solve the noise issue by applying Generative Adversarial network (GAN) to create high-resolution images from a low-resolution image. However, their method did not address the limited character and number variation problem. Another license plate recognition research also relies on the permutation method [34]. The authors proposed to eliminate the bias of permutation method by creating synthetic license plate images to train the recognition network in the fully

connected layers. However, the details about the way to create synthesis image as well as the image amount and training process were not given.

In this paper, we propose a novel augmentation method that automatically generates annotated images of numbers resembling the local appearance of the target digits. By taking advantage of the generative deep neural network, we are able to produce a larger variation in numbers and control the frequency of numbers at each digit in the training data in a fully automatic way.

C. GENERATIVE NETWORK

For generating new numbers with the appearance of a given digit, we need a generic interpretable representation and a bidirectional network to reconstruct images of these new numbers. Over the last decade, the most outstanding generative models are the Variational Autoencoder (VAE) [35] and the GAN [36]. However, both GAN and VAE originally perform unconditional generation which is not suitable to our approach. Therefore, for better control of the generated results, other extension versions were reviewed.

Kingma *et al.* [37] introduced CVAE, an extension of their VAE, by constraining a condition to control the output. In addition, [38] introduced a framework using partially-specified graphical model structures and semi-supervised learning in the domain of VAEs to perform a conditional generation. However, CVAE performs poorly for the style transfer of number images compared with other methods in [16].

Similarly, the extension version of GAN, called InfoGAN, was proposed in [39]. InfoGAN can learn interpretable and disentangled representations and gain control over the content and style of the generated images. CVAE-GAN [40], which combines VAE with GAN to build a conditional generative model, can take the fine-grained category label as the input and generate images in a specific category. However, both VAEs and GANs have difficulty in exploiting a prior structure and are difficult to train.

Patacchiola *et al.* [16], succeeded in achieving impressive results with a simple architecture called Y-AE. Without changing the structure of conditional Autoencoder (cAE), the authors presented a new training procedure that allows the disentanglement of representation information in the latent space without using variational methods or adversarial losses. In their experiments, they compared Y-AE with other generative networks (cAE, cVAE, adversarial-AE [41] and beta-VAE [42]), in many aspects to verify the effectiveness and show the possibility of use in a large variety of domains with minimal adjustments.Considering their simplicity of training and practicability for our problem, we choose Y-AE for generating new numbers, along with transferring the local appearance of target digit.

III. GENERATIVE AUGMENTATION

In this section, we describe the proposed generative augmentation that consists of the three main stages, (i) appearance preserving number generator, (ii) rotating digit generator, and (iii) counter image composition, step-by-step in detail. Lastly, we explain the balanced dataset generating algorithm.

A. APPEARANCE PRESERVING NUMBER GENERATOR

To solve the insufficient training data problem, we need to generate annotated images of all possible numbers (from "0" to "9") for a digit without altering the local appearance of the digit. For this task, we employ Y-AE [16] to transfer the local appearance of a digit to an arbitrary number. Although other generative models, such as GAN [36], are also gaining attention for style transferring, Y-AE can adapt to a large variety of image domains with minimal adjustments.

The Y-AE architecture is based on cAEs that contain an encoder and a decoder. Fig. 2 depicts the training procedure of Y-AE. The encoding phase of a Y-AE is identical to a standard cAE, which is called the input branch in this paper, but the reconstruction is unique, consisting of two branches called the sample branch and target branch, respectively.

In the input branch, the input image is encoded by applying two types of activation functions. Considering that content information can be represented as discrete latent units and style information needs continuous presentation, the SoftMax function is used to define the content and the Sigmoid function is used for style. Then, the output of the encoder is split into two paths by giving the style information as the input to the two branches, whereas the sample content information (c_S) is used to calculate the loss (L_1) with the sample label (l_S) using cross-entropy loss, by calculating a separate loss for each class per observation (j) and summing the results to identify the generated content from the input image:

$$L_1 = CE\,(c_S, l_S) = -\sum_j c_{S,j}\, log\, l_{S,j} \tag{1}$$

The sample branch does the reconstruction by taking the style information produced by the input branch together with the sample label to generate the sample label image before re-encoding it. The sample label image (s_S) is used to calculate the standard least-squared error reconstruction loss (L_2) with the input sample image (s_I) to ensure appropriate reconstructions:

$$L_2 = |s_I - s_S|^2 \tag{2}$$

On the other hand, the target branch takes the same style information and the target label as the input. At the training stage, a random number generated with a uniform random function is used as an input of the decoder to obtain the random label image for re-encoding. Then, cross-entropy loss (L_3) is calculated to verify the target branch reconstruction content (c_T) with the label (l_T):

$$L_3 = CE\,(c_T, l_T) = -\sum_j c_{T,j}\, log\, l_{T,j} \tag{3}$$

After finishing the above-mentioned two branch reconstruction phases, a loss based on the Euclidean distance is computed to confirm the style information has not been

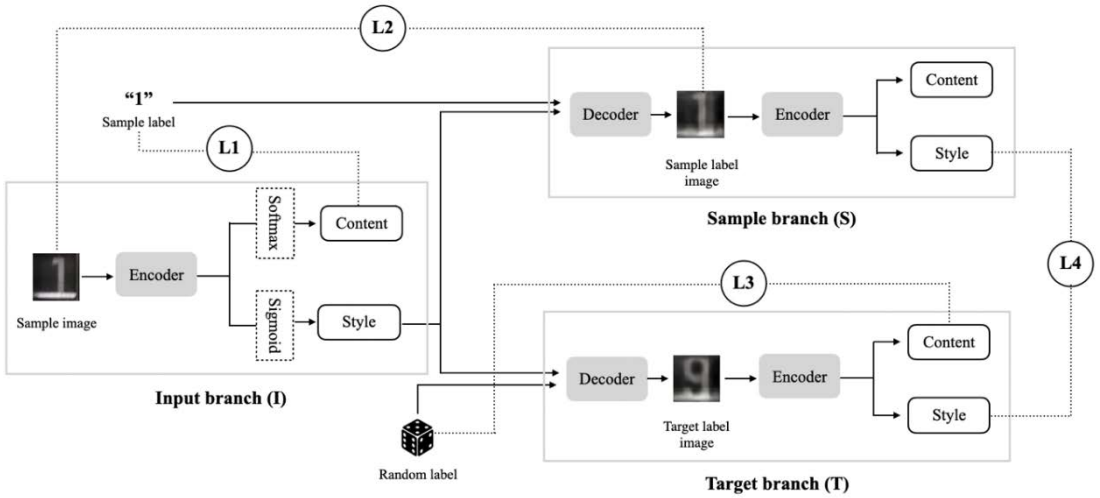

FIGURE 2. Y-AE model. The reconstruction phase is divided into two branches: sample branch and target branch, which uses weight sharing to enforce latent disentanglement.

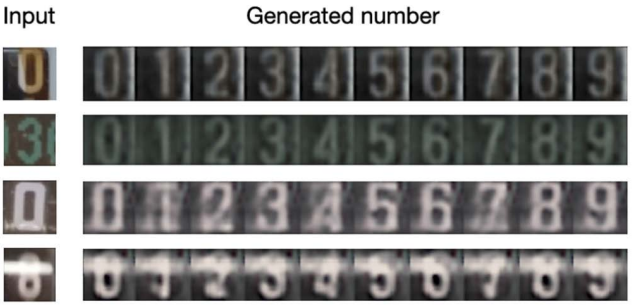

FIGURE 3. Samples generated by a Y-AE trained on number image datasets. The first column is the input, and other columns are the reconstructions given all possible numbers ("0"–"9") as the content. Y-AE is able to keep the style and change the content.

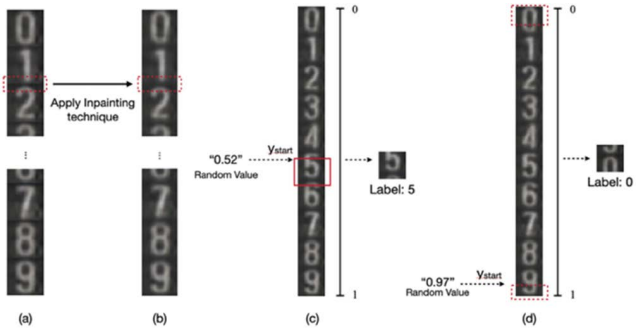

FIGURE 4. The rotating digit generator procedure. (a) The master image made by arranging the images of numbers "0" to "9" vertically. (b) The master image after applying the image inpainting technique. (c) A sample of a rotating digit generated with a random number 0.52. (d) A sample of a rotating digit generated with a random number 0.97.

changed, remaining consistent in the two branches. That is, the final loss (L_4) of the style information is computed from the sample branch (s_S) and the target branch (s_T) as follows:

$$L_4 = d\,(s_S, s_T) = \|s_S - s_T\| \tag{4}$$

The losses defined above are then integrated into the global loss function (L_G), where the content loss (L_3) and style loss (L_4) can be controlled by altering explicit weight (λ_e) and implicit weight (λ_i) respectively.

$$L_G = L_1 + L_2 + \lambda_e L_3 + \lambda_i L_4 \tag{5}$$

It was shown in [16] that the structure and training procedure of Y-AE could successfully separate the content and style in the latent space for difference tasks.

After training, the trained model can be used to generate the image for any number while keeping the local appearance of a particular input image. Fig. 3 shows some of the generated images in our experiment.

B. ROTATING DIGIT GENERATOR

By taking advantage of digit style transfer, we are able to generate images of all numbers for each digit of the meter image. To generate the annotated images simulating a rotating meter, we first generate a master image by arranging the images of numbers "0" to "9" vertically, as shown in Fig. 4(a).

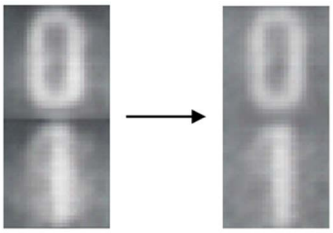

FIGURE 5. The result of apply inpainting technique to blend the border between two numbers.

To stitch the images of consecutive numbers seamlessly, the biharmonic function-based image inpainting technique [43], which aims to smooth images in a specific area by using information from thesurrounding mask, was adopted to blend borders between numbers as shown in Fig. 5. Fig. 4(b) shows an example of a master image after applying the image inpainting technique to Fig. 4(a).

Once a master image has been produced, we apply our rotating digit generator to generate images mimicking a real rotating meter. First, we generate a random floating number r ($r \in [0.0, 1.0]$). Then r is used to define the y position, denoted as y_{start}, for cropping the master image, as shown in Fig. 4(c), to obtain an image mimicking a particular state of a rotating digit. Then, the label for the cropped image is set as:

$$Label = Round(r \times 10)$$

Note that we make the master image satisfying periodical condition, that is, if the end position of the cropping position exceeds the bottom of the master image, then the exceeded part will be taken from the top of the master image as shown in Fig. 4(d). Such periodic operation also mimics a real rotating meter digit.The detailed procedure is summarized in *Algorithm 1*. This algorithm can be adapted for generating digit with specific labels simply by limiting the range of random number.

Algorithm 1 Rotating Digit Generator

```
function rotating_digit (master_img, h_master, w_crop, h_crop):
Input: master_img: a master image with number "0"-"9" arranged vertically,
       h_master: height of master image,
       w_crop: width of cropped digit image,
       h_crop: height of cropped digit image
Output: crop_img: The cropped digit image and and its label
r = random.randrange(0,1)
y1 = r * h_master
y2 = y1 + h_crop
x1 = 0
x2 = x_start + w_crop
if y2 > h_master then
   y2 = h_master
   crop_img1 = master_img[y1 : y2, x1: x2]
   y1 = 0
   y2 = h_crop - (h_master – y1)
   crop_img2 = master_img[y1: y2, x1: x2]
   crop_img = Concat(crop_img1, crop_img2)
else
   crop_img = master_img[y1: y2, x1: x2]
end
label = Round(r * 10)
if label > 9 then
   label = 0
return (crop_img, label)
```

C. COUNTER IMAGE COMPOSITION

The final step of generative augmentation is to set the generated images of each digit to the counter images. We resize the cropped digit image to that of the original counter image. Then, we apply seamless cloning [44], which is a technique for copying an image region from a foreground image onto a background image naturally without visible seams. In this way,we are able to create a new training dataset of sufficient and well-balanced annotated meter images from few sample images. Some generated meter images are shown in Fig. 6.

FIGURE 6. Counter images produced with the proposed generative augmentation. Above the line are the original images. Under the line are the generated images and their automatically annotated labels.

D. GENERATING A BALANCED DATASET

We also present an algorithm to create a training dataset consisting of a well-balanced amount of all numbers for each digit, using the proposed generative augmentation. The basic idea is to use weighted random selection to select a number from the 10 numbers "0"–"9" based on a given probability distribution. We define the relative frequency F_i^k of a number $i(i = 0, 1, \ldots, 9)$ at a digit k as

$$F_i^k = \frac{Frequency\ of\ number\ i\ at\ digit\ k}{Sum\ of\ frequencies\ of\ all\ numbers\ at\ digit\ k}$$

denoting FT_i^k the target relative frequency and FC_i^k the current relative frequency of number $i(i = 0, 1, \ldots, 9)$ for digit k, respectively. Then, number $i(i = 0, 1, \ldots, 9)$ should be selected with the following probability:

$$p_i^k = FT_i^k - FC_i^k$$

After number i is selected, a random floating-point number r is generated and fed to the rotating digit generator to generate a digit image with label i.

$$r = \begin{cases} rand\left[\dfrac{i-0.5}{10}, \dfrac{i+0.5}{10}\right), & i \in [1, 9] \\ rand\left[0\ .95, \dfrac{i+0.5}{10}\right), & i = 0 \end{cases}$$

With the images of all digits obtained, we composite them into a new counter image and update the current relative frequency $FC_i^k(i = 0, 1, \ldots, 9)$ for all digits. To achieve balanced amounts among all numbers, the target frequency distribution $FT_i^k(i = 0, 1, \ldots, 9)$ should be box shaped for all digits. The generation process is repeated until the desired number of counter images has been generated. The detailed procedure is summarized in *Algorithm 2*.

In the algorithm, *get_relative_frequency* is the function for computing the relative frequency (F_i^k) of numbers "0"–"9" in all digits from the input data distributions, while *calculate_probability* is a function for calculating the distribution of probabilities (p_i^k) of numbers at digit k for weighted random selection. Then, we use the number probabilities to select number (I) randomly from "0"–"9" following their

Algorithm 2 Generating Dataset With Balanced Distribution of Numbers

```
function generate_balanced_dataset (unbalanced_dataset, N, K):
Input: unbalanced_dataset: original training dataset with
       unbalanced distribution
       N: counter image amount to be generated
       K: number of digits in a counter image
Output: balanced_dataset: generated dataset with balanced dis-
tribution of numbers
Declaration: Relative_frequency_list[K][10]: list of all 10
             number's relative frequency for all K digits
             probability_list[10]: list of 10 number's
             probabilities for weighted random selection
             NUMBER[10]: list of number 0∼9
             digit_img_list[K]: list of generated digit images
             label_list[K]: list of generated labels
FT = 0.1
Relative_frequency_list =
get_relative_frequeny(unbalanced_dataset)
for i = 1 to N do{
    for j = 1 to K do {
      probability_list = calculate_probability(FT,
      Relative _frequency_list[j])
      I = weigted_random_choice(NUMBER, probability_list)
      if I = 0 then
        r = random_float_number (0.95, (I+0.5)/10)
      else
        r = random_float_number (I-0.5)/10, (I+0.5)/10)
      (digit_img_list[j], label_list[j]) =
      rotating_digit_generator(r, master_img, h_master,
      w_crop, h_crop)
    }
    generated_counter_img =
    counter_image_composition(digit_img_list,
    label_list)
    Add generated_counter_img to balanced_dataset
    update_frequency(Relative_frequency_list)
  }
```

probability using *weighted_random_choice* function. The *rotating_digit_generator* and *counter_image_composition* are functions of generative augmentation for generating rotating digits and compositing the digit images into a counter image, respectively, to get a new counter image together with their labels. Finally, after we add a generated counter image to the dataset, we update the current frequency distribution for all digits using *update_frequency* function.

IV. AUTOMATIC METER READING FRAMEWORK

Fig. 7 depicts the framework of our end-to-end automatic meter reading. We adopt a two-stage approach: the counter detection stage takes a whole meter image as the input and locates the counter region and the digit recognition stage performs digit segmentation and recognition simultaneously on the cropped counter image. CNN-based approaches are employed for both stages. For counter detection, we employed the YOLOv5 [22] model, a state-of-the-art deep learning based object detection model that is suitable for detecting objects occupying a small portion on an image. To read values from the meter image, we made an extension to CR-NET [29], a recognition model based on YOLO object detectors [45]. The details of each part of the framework are given in the remainder of this section.

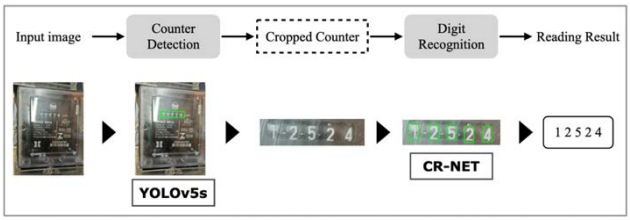

FIGURE 7. The framework of the proposed end-to-end deep learning-based automatic meter reading.

A. COUNTER DETECTION

YOLOv5 [22] was released in June 2020 with a significant improvement made over its predecessors in both accuracy and speed. Moreover, the size of the weights trained from YOLOv5 network is small, which is nearly 90% smaller than YOLOv4 [22], and hence is suitable for deployment to the embedded devices to implement real-time detection [46]. YOLOv5 structure consists of three components: backbone network, neck network and detect network. By performing detection at three detection layers for detecting the objects of different sizes, YOLO5 has yielded a remarkable performance on small object detection. On the other hand, to detect only one class, Laroca *et al.* [15] stated that very deep models are not necessary. Therefore, we decided to use a smaller model, YOLOv5s (YOLOv5-small), at the detection stage.

B. DIGIT RECOGNITION

We employ CR-NET, proposed by S. M. Silva *et al.* [29], for the recognition stage. CR-NET is a YOLO-based model proposed for license plate character detection and recognition, and it achieves remarkable results, not only in license plate recognition [29], [32], but also in AMR recognition [15]. Laroca *et al.* [15] compared it with two other CNN-based approaches and showed it could achieve a promising recognition result. The network architecture is shown in Table1, and we have adjusted the number of filters to 75 at the last convolutional layer of the CR-NET architecture, as we want to predict only 10 classes (numbers "0"–"9").

We apply a non-maximal suppression (NMS) algorithm to eliminate some digits that might be detected more than once by the network. Furthermore, we consider only the digits recognized with the highest confidence on each overlap. Some cases show that over five characters can be recognized.

V. EXPERIMENT

In this section, we describe the experiments for evaluating the effectiveness of the proposed generative augmentation technique together with the performance of the automatic meter-reading framework. First, we investigate the performance of counter detection, as the regions used in the following stages are extracted from the detection results. Then, we present the

TABLE 1. CR-NET with some modifications for counter recognition.

No.	Layer	Filters	Size	Input	Output
0	Conv	32	3×3/1	192 x 96 x 3	192 x 96 x 32
1	Max		2×2/2	192 x 96 x 32	96 x 48 x 32
2	Conv	64	3×3/1	96 x 48 x 32	96 x 48 x 64
3	Max		2×2/2	96 x 48 x 64	48 x 24 x 64
4	Conv	128	3×3/1	48 x 24 x 64	48 x 24 x 128
5	Max	64	1×1/1	48 x 24 x 128	48 x 24 x 64
6	Conv	128	3×3/1	48 x 24 x 64	48 x 24 x 128
7	Max		2×2/2	48 x 24 x 128	24 x 12 x 128
8	Conv	256	3×3/1	24 x 12 x 128	24 x 12 x 256
9	Max	128	1×1/1	24 x 12 x 256	24 x 12 x 128
10	Conv	256	3×3/1	24 x 12 x 128	24 x 12 x 256
11	Max	512	3×3/1	24 x 12 x 256	24 x 12 x 512
12	Conv	256	1×1/1	24 x 12 x 512	24 x 12 x 256
13	Conv	512	3×3/1	24 x 12 x 256	24 x 12 x 512
14	Conv	75	1×1/1	24 x 12 x 512	30 x 10 x 75
15	Detection				

TABLE 2. The proportion of transitional state image and normal image in each dataset.

Dataset	Total	Normal	Transitional state
Training	800	644	156
Validation	400	309	91
Testing	800	550	250

results of digit recognition based on the proposed generative data augmentation techniques.

A. IMPLEMENTATION

1) DATASET

As mentioned in the previous sections, a few meter datasets are available to the public. The relative large dataset from [28], which contains 6,000 water meter images with cropped counters, is not suitable for our framework, as it requires the whole meter image to be processed beforehand. Therefore, we decided to use the UFPR-AMR dataset [15], which consists of 2,000 fully annotated images of various meters. The images were captured in a warehouse of a service company with different conditions with a resolution between $2,340 \times 4,160$ and $3,120 \times 4,160$ pixels, and they contain a well-defined evaluation protocol to assist in the development and evaluation of AMR methods: 800 images for training, 400 images for validation, and 800 images for testing.

To evaluate the performance of our new augmentation technique in dealing with the transitional states of rotating digits, we manually identified the images with transitional states of rotating digits in the testing image set. As shown in Table 2, the images with transitional states of rotating digits account for a big proportion.

2) EXPERIMENTAL SETUP

We performed our experiments on a CPU with a 12-Core AMD Ryzen Threadripper 1920x 3.5GHz processor, 64GB of RAM, and an NVIDIA Titan RTX GPU. Both the detection model and recognition model were trained using the Darknet framework [47]. The parameters for training the models for counter detection, digit recognition, andgenerative augmentation are shown in Table 3.

TABLE 3. Training parameters.

Model	Iteration (k)	Learning rate	Training scheduler (k)
Detection	50	10^{-3}	40, 45
Recognition	10	10^{-3}	2, 6, 9
Generative augmentation	5	10^{-4}	Fixed to 10^{-4}

For generative augmentation, the model was trained using Tensorflow [48] with $epochs = 5,000$, $learning\ rate = 10^{-4}$, $implicit\ units = 32$, $lambda\ explisit\ (\lambda_e) = 1.0$, and $lambda\ implisit\ (\lambda_i) = 0.1$. We cropped the number images from the training and validation data and resized them to 32×32 pixels to get 6,000 digit images in total for training Y-AE.

3) EVALUATION METRICS

For evaluating counter detection, we measure the accuracy by computing the intersection over union (IoU) between the detected bounding box and the ground truth of the counter area. This approach is often used for evaluation in object detection challenges, such as the PASCAL VOC challenge [49] and MS COCO challenge [50]. It is also used in other AMR works [15], [24], [25].In the experiment, we design to evaluate the detection accuracy following the MS COCO challenge [50] that measure mAP over different IoU thresholds, from 0.5 to 0.95, to indicate how precise that the predicted bounding boxes aligned with the ground truth.

To evaluate the accuracy of digit recognition, we use the following *Digit Recognition Accuracy*(DRA) metric:

$$DRA = \frac{Number\ of\ correctly\ recognized\ digit}{Number\ of\ digits\ in\ the\ test\ set}$$

As well as the Average Precision (AP) which is the integration of Precision-Recall (PR) curve. Mean Average Precision (mAP) is calculated as

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i$$

Here, N is the number of classes.

We also introduce a metric called Average Maximum Structural Similarity Index Measure (AMSSIM) which is computed as the average of the maximum Structural Similarity Index Measure (SSIM) [51]:

$$AMSSIM(X, Y) = \frac{1}{N}\sum_{i=1}^{N} MAX\ (SSIM(x_i, y_i))$$

TABLE 4. Counter detection results on the UFPR-AMR dataset measured by mAP over different IoU thresholds.

Method	IoU Threshold						
	0.5	0.6	0.7	0.8	0.9	0.95	Average (0.5:0.95)
YOLOv5s [22] (480x480)	99.9	99.4	99.0	97.6	83.2	38.8	86.3

Denoting N is number of image pairs (x, y) that SSIM were computed. SSIM is a well-known metric for measuring the perceptual similarity between two images. The SSIM result is a decimal value between 0 and 1, value 1 indicates perfect structural similarity of two images, while a value of 0 indicates no structural similarity.

B. EVALUATION

We first assess the counter detection to verify the performance of the modified YOLOv5 in detecting counter regions. Afterward, the evaluation of the recognition stage, based on the generative data augmentation techniques, is conducted through three tests. The first test focuses on a particular digit and validates the effectiveness of the proposed appearance-preserving number generator to solve the issue of the lack of a sufficient variation in numbers for particular digits. The second test focuses on the transitional state problem to validate the effectiveness of the proposed rotating digit generators to address this issue. The last aims to validate comprehensively whether our generative augmentation technique combined with the AMR framework can be used to create large, balanced quantities of training data and to achieve a high recognition accuracy.

In all three tests, we compare our method with the existing character permutation method [15] and also with a standard augmentation technique in the third test.

1) COUNTER DETECTION

With the YOLOv5s model, an impressive detection result was achieved, even though the quality of some images is quite low and counters occupy a small portion in the image.

Table 4 shows the results of mAP over different IoU thresholds(from 0.5 to 0.95), following COCO's standard metric [50]. The last column is the average.

YOLOv5s model performs well even with high IoU threshold. The results show that the predicted bounding boxes aligned precisely with the ground truth. Moreover, when set IoU > 0.5 following the approach from the Pascal VOC Challenge [49], the model correctly detected all counter regions in the UFPR-AMR dataset.

2) EVALUATION OF APPEARANCE PRESERVING NUMBER GENERATOR

We conducted the experiment to validate the effectiveness of our method in the case of lacking a sufficient variation in numbers for a particular digit. Because the last (least significant) digit is likely to have more variation in numbers in the test dataset, we therefore create the scenario by extracting the meter images whose last digit contains one single number in the training dataset and use these images only to create the training dataset consisting of other numbers at the last digit. We generate 400 training images with two augmentation methods: our method and the character permutation method [15]. The amount of images extracted for all 10 numbers "0"∼"9" and the DRA of the models trained with the 400 images generated from the extracted images with the two augmentation methods are shown in Fig. 8.

TABLE 5. Recognition Results using the training dataset generating from the images whose last digit is "1" with 400 validation images and 800 testing images.

Method	Training	DRA (%)
Without augmentation	89	59.05
Permutation [15]	400	76.48
Proposed		85.47
Permutation [15]	1000	71.91
Proposed		87.06

We can see that number "8" and "3" have the smallest and largest amount of images, respectively. The proposed method outperforms the existing permutation method when any number is used for data augmentation. DRA could reach as high as 85.62 % when use less than 100 images consisting of a single number only for training.

Table 5 shows the results when using the image with number "1" at the last digit for generating the training dataset. The recognition results are greatly improved when taking advantage of data augmentation. Our method achieved a recognition accuracy of 85.47%, which outperforms the existing method by 8.99%. Furthermore, when the generated image increased to 1,000 images, the accuracy of permutation method dropped. Because the original image dataset only has 89 images, the permutation method has a limit of number variety for swapping which caused the unbalanced numbers on each digit position.

Fig. 9. shows the distribution of 10 numbers over the 5 digits in the training set (1,000 images) generated by each method, which proves that our proposed method can generate more balanced dataset than the permutation method especially when the original dataset is small and hence result in better recognition result.

The accuracy of the proposed method become saturated from 1,000 training images and starts to drop at 2,500 training images. This is because all the training images are actually generated from only 89 images with limited style variations, which causes the overfitting problem and make the accuracy dropped.

Table 6 shows the AP (area under PR curve) of each class by digit position when using the two methods for generating 1,000 training images. We can see that our method could

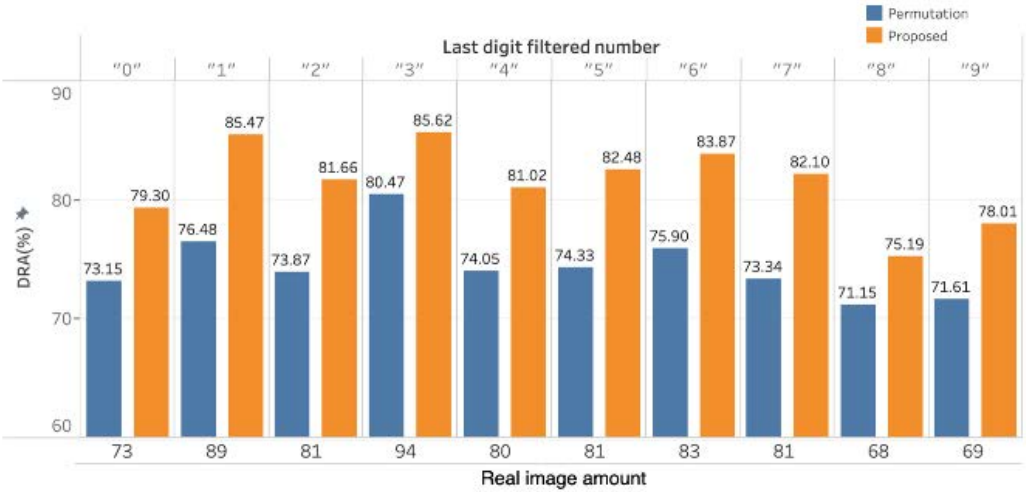

FIGURE 8. The amount of images extracted for all 10 numbers "0"∼"9" and the accuracy of the models trained with the 400 images generated from the extracted images with the two augmentation methods: the proposed method (orange) and the character permutation method [15] (blue).

TABLE 6. AP (Area under PR curve) and mAP by digit position.

Class	AP (Area under PR curve)									
	Permutation [15]					Proposed				
	Digit 1	Digit 2	Digit 3	Digit 4	Digit 5	Digit 1	Digit 2	Digit 3	Digit 4	Digit 5
0	0.979	0.981	0.993	0.979	0.969	0.976	0.979	0.980	0.975	0.966
1	0.616	0.757	0.647	0.729	0.600	0.702	0.697	0.726	0.700	0.618
2	0.747	0.773	0.793	0.804	0.721	0.826	0.741	0.750	0.789	0.769
3	0.788	0.732	0.802	0.793	0.795	0.859	0.800	0.827	0.796	0.785
4	0.874	0.904	0.841	0.902	0.871	0.912	0.906	0.914	0.945	0.879
5	0.865	0.829	0.833	0.845	0.850	0.870	0.821	0.860	0.857	0.825
6	0.610	0.643	0.652	0.613	0.694	0.682	0.692	0.783	0.736	0.654
7	0.857	0.862	0.902	0.881	0.866	0.897	0.847	0.844	0.858	0.909
8	0.958	0.943	0.944	0.939	0.964	0.952	0.947	0.959	0.934	0.945
9	0.804	0.809	0.756	0.804	0.761	0.754	0.821	0.803	0.778	0.832
mAP	**0.810**	**0.823**	**0.816**	**0.829**	**0.803**	**0.843**	**0.825**	**0.845**	**0.837**	**0.818**

particularly improve the AP of the classes with smaller number of training images, such as the class of number "6", when using permutation method. In overall, our method could improve mAP for all 5 digit positions. As expected, both methods got the lowest mAP at the last digit position, which is caused by the fact that only images consisting of number "1" is used for training or for generating the training dataset.

To further validate the effectiveness of the proposed appearance preserving number generator, we newly introduce a metric called AMSSIM. Structural Similarity Index Measure (SSIM) is a well-known metric for measuring the perceptual similarity between two images. The proposed appearance preserving number generator aims at generating the image of new numbers resembling the local appearance of a particular digit. Therefore, if the data augmentation went successfully, for each number in the test image, it should be possible to find an image with similar appearance, that is, having high SSIM value in the training set. Therefore, to evaluate the quality of the training set generated from a particular number, for each of the remaining 9 numbers in test dataset, we compute the SSIM with all images of the same number in the training set and find out the image with maximum SSIM. Then AMSSIM for each of the 9 number is computed as the average of this maximum SSIM of the images in the test dataset.

Table 7 compare AMSSIM of the proposed method and the character permutation method when using the 89 images with "1" at last digit to generate 400 images as the training set. We can see that the proposed method achieved much higher AMSSIM than the existing method on every generated number, which explain from another perspective on why the proposed method could achieve higher recognition accuracy than the existing method, in addition to the data balance issue.

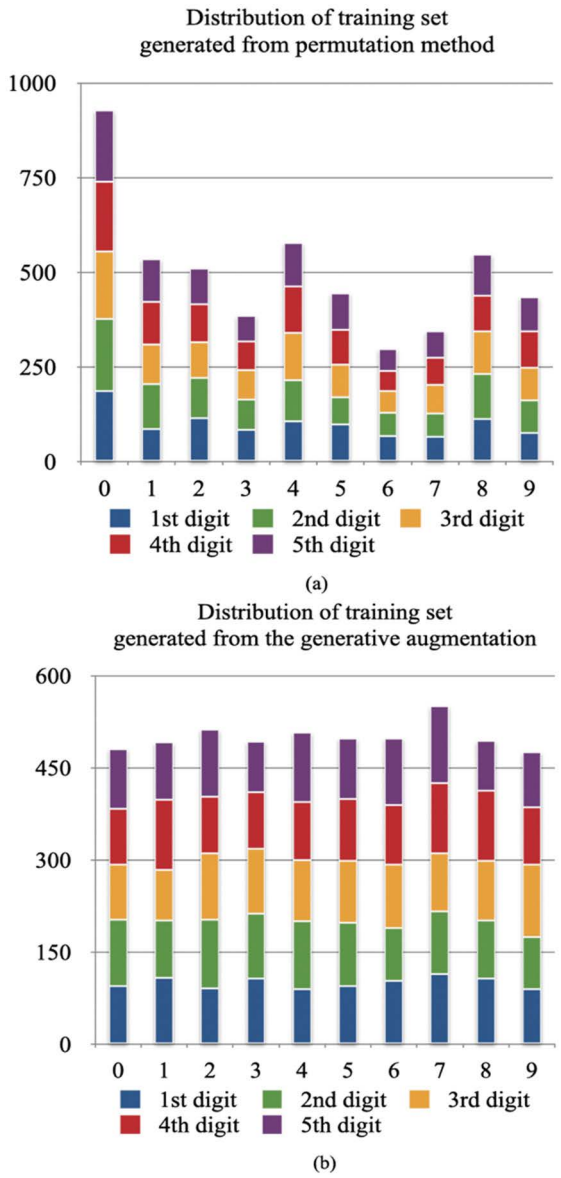

FIGURE 9. The number distribution of each digit position in training dataset generated from 2 methods (a) the permutation method (b) the proposed generative augmentation.

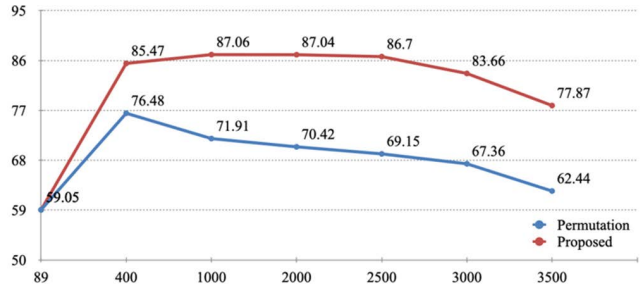

FIGURE 10. Recognition accuracy plot while increasing the training data generated from 2 methods 1) Permutation method (blue) 2) Proposed method (red).

3) EVALUATION OF ROTATING DIGIT GENERATOR

Another challenging issue of the reading stage is the recognition of transitional states between two consecutive numbers.

TABLE 7. The average maximum structural similarity index measure (AMSSIM) of each 9 numbers generated from number "1" in the last digit comparing with image on testing set.

Number	AMSSIM	
	Proposed	Permutation [15]
0	0.722	0.407
2	0.651	0.453
3	0.618	0.352
4	0.621	0.449
5	0.681	0.290
6	0.715	0.410
7	0.640	0.385
8	0.657	0.336
9	0.677	0.425

TABLE 8. Recognition results for digits at transitional states.

Method	Total Training Images	DRA (%)	
		Average	Transitional state digit
Without augmentation	800	93.96	69.37
Permutation [15]	10,000	94.52	66.58
Proposed	10,000	**96.28**	**83.26**

First, we selected 250 images with a transitional state from the test image set to evaluate the proposed augmentation method. For comparison, we generated images using two augmentation methods and applied them to train the recognition model.

The recognition result in Table 8 shows that the proposed generative augmentation yielded the best recognition results, especially for the transitional state digits. The average accuracy of our proposed method is higher than that of the existing method by 1.76%. For the transitional state digits, our proposed method outperforms the existing method by 16.68%.

This evaluation demonstrates the effectiveness of the proposed rotating digit generator for automatically generating annotated images of transitional states.

4) COMPREHENSIVE EVALUATION

We first conducted an experiment on a small dataset with the aim to mimic the problem of data insufficiency. Half of the UFPR-AMR training dataset was randomly selected, and we generated training images from these 400 images with three augmentation methods, our proposed method, the character permutation method and traditional augmentation method provided by [31], to obtain a training dataset with 800 images.

The traditional image data augmentation technology adopted in [35] is to synthesis various image degradation factors and have been widely used for enhancing the robustness of detection or recognition models against unconstrained

TABLE 9. Recognition results by applying image augmentation to the half of original training set.

Method	Total Training Images	DRA (%)
Without Augmentation	400	88.65
Traditional	800	88.71
Permutation [15]	800	89.10
Proposed	800	**91.63**

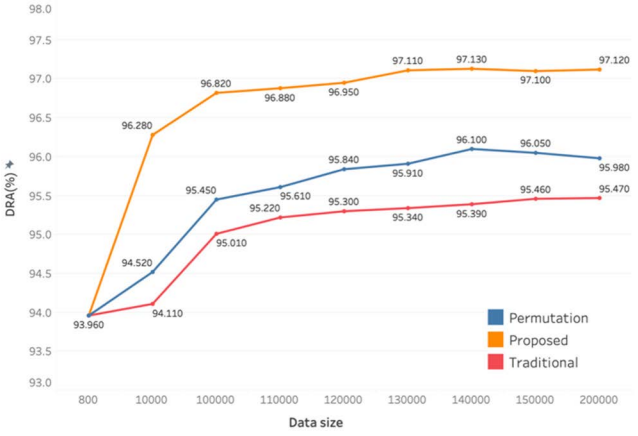

FIGURE 11. Recognition accuracy plot while increasing the training data generated from 3 methods 1) Permutation method (blue) 2) Proposed method (orange) and 3) Traditional method (red).

environment. We implemented the image synthesis method to mimic the following artifacts:

1) Crop and pad each digit tomimic the occlusion digit on the meter image that may occur by the camera's shooting angle
2) Blur and noise
3) Uneven lighting by randomly adjusting brightness and contrast
4) Geometric transform, such as stretching transformation, affine transformation, rotation, and down sampling.

The results obtained with the 3 methods are shown in Table 9. Our generative augmentation technique achieved the best result, outperforming the character permutation [15] by 2.53% and the traditional augmentation method by 2.92%.

In the last test, we compared our augmentation method with the existing character permutation method and traditional augmentation technique for generating a large quantity of training data. We have conducted the experiment by generating 100,000-200,000.

Fig. 11 shows the recognition accuracy when increasing the training data from 100,000 to 200,000. All methods' accuracies become saturated from 110,000 images. The traditional augmentation which is unable to generate new number has a lowest accuracy while a gap between permutation method and our proposed method remaining. The superiority of our method over the permutation method is considered mainly lying on two reasons. One is that a part of the remaining errors occurred with the intermediate state digits. When the augmentation proceeds, our method can generate more intermediate state training data and hence reduced the recognition errors. The other reason is the unbalance of the amounts of different numbers which may cause the accuracy dropping of permutation method in the latest period (when increase data size from 150,000 to 200,000 images). Our algorithm (Algorithm II given in Section III) can generate balanced number, hence contributed to the stability of the accuracy. Although the permutation method was also implemented in a way trying to generate meter images with a uniform distribution of all numbers as possible, it is inherently impossible to get a precise control over the amount of all numbers for each digit position.

Nevertheless, the dataset used for this test consists of the images which is taken from various meters under various environments. Even though the first digit position has an unbalanced data, other positions still have enough number to use for swapping among digits. For such data, permutation method usually can take advantage of increasing the data size. However, these public datasets are available for academic use only. On the other hand, although it is easy to collect a large number of images in industry application, the image amount for each number is usually severely unbalanced by digit positions. The significant digits usually take long time to be updated and it is difficult to get the images of different numbers for these digits. This is the problem drove us to develop the generative augmentation method which can generate different numbers for those digits from just very few numbers.

C. DISCUSSION AND LIMITATION

The proposed generative augmentation, which mainly consists of three stages, can successfully transfer the local appearance of a digit to an arbitrary number by taking advantage of generative network. It can also mimic a particular state of a rotating digit. The experiments show that the proposed method can generate up to 2000 training images from only 89 imbalanced sample images, which improves the recognition accuracy by about 26% from non-augmentation dataset. The accuracy starts to drop with 2,500 training images due to the overfitting problem caused by using the images with only "1" as the last digit. However, in the comprehensive experiment that increases the training dataset to more than 100,000 images from the original dataset consisting of 800 images, the recognition accuracy of the proposed method continued to improve and reached to 97.12% at 200,000 images, though it almost saturated from 110,000 images.

Fig. 12 shows some results by the existing method and our proposed method. The numbers shown on the top row represent the ground truth, and the middle and bottom rows show the recognition results from the existing method and the proposed method, respectively. The failed digits are shown

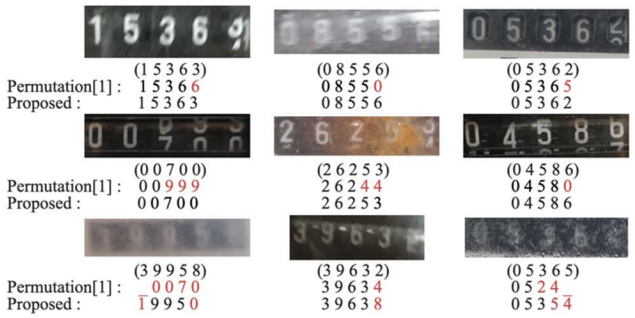

FIGURE 12. Results using the existing method and the proposed generative augmentation method.

FIGURE 13. Examples of generated images failed to transfer the local appearance realistically.

in red.As can be seen, the proposed method succeeds even in challenging cases, such as rotating states and low image quality, and it failed only in highly difficult cases that were even difficult to recognize by a human, as shown in the last row of Fig. 12.

Fig. 13 shows some cases in which the images generated by the proposed method do not resemble the original one perfectly. On the first and second examples, dirt and light on the input images were not correctly transferred onto the generated images. For the third and fourth examples, there are blur and noise on the generated images. However, these kinds of failure cases account for just a small proportion. In future work, we intend to improve the appearance preserving number generator by exploring new deep generative networks. For the fourth images, the font of meter is not successfully transferred. This is mainly due to the fact that the training dataset of generative network has few images (0.003% only) of the similar font as the fourth meter image. The generative network failed to learn the font and transfer it to the generated image precisely but instead used the font of some meter which has larger proportion in the dataset. Currently this is the main limitation of the proposed data augmentation method. However, compared to capturing conditions, a slight difference in font style usually does not affect the recognition accuracy. Therefore, for the purpose of AMR, the proposed technique, which can transfer the appearance of numbers captured under unconstrained environments, should have sufficient significance in real applications.

VI. CONCLUSION

In this paper, we presented a new automatic rotating meter-reading system featuring a novel data-augmentation technique called generative augmentation. This augmentation technique takes advantage of a generative deep neural network to generate new annotated digit images, along with

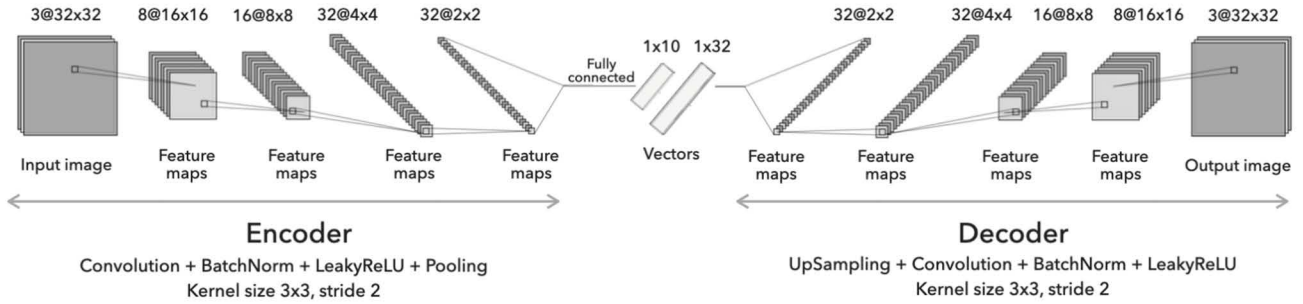

FIGURE 14. Neural network structure of generative network consists of an encoder network and a decoder network. The notation of each layer represents channels@vector size. After fully connected, 2 activation functions have applied (as explain in Section III: A. Appearance preserving number generator) and got 2 vectors, 1) 10 unit vector which refer to number "0"-"9", 2) 32 unit vector which contain the style values.

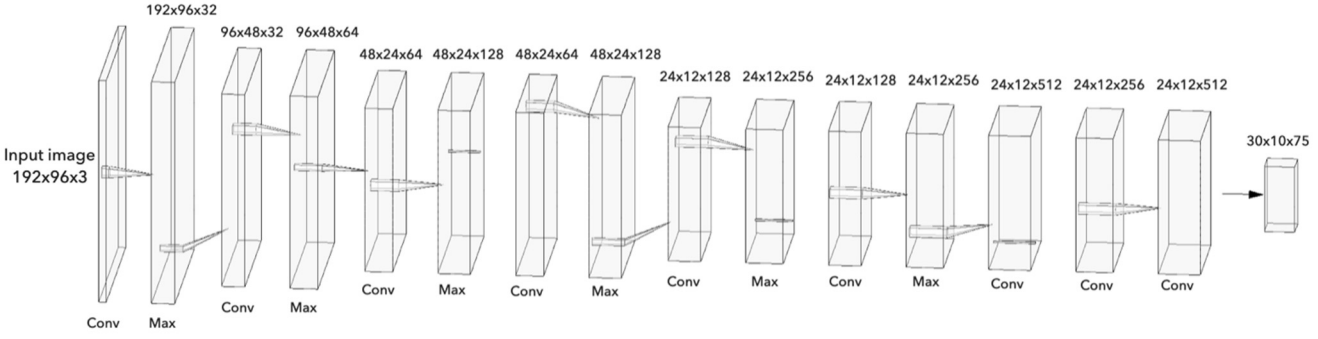

FIGURE 15. Neural network structure of recognition network which is consist of 15 layers. The notation of each layer represents the size given as height x width x channels, Conv: convolutional layer, and Max: Max pooling.

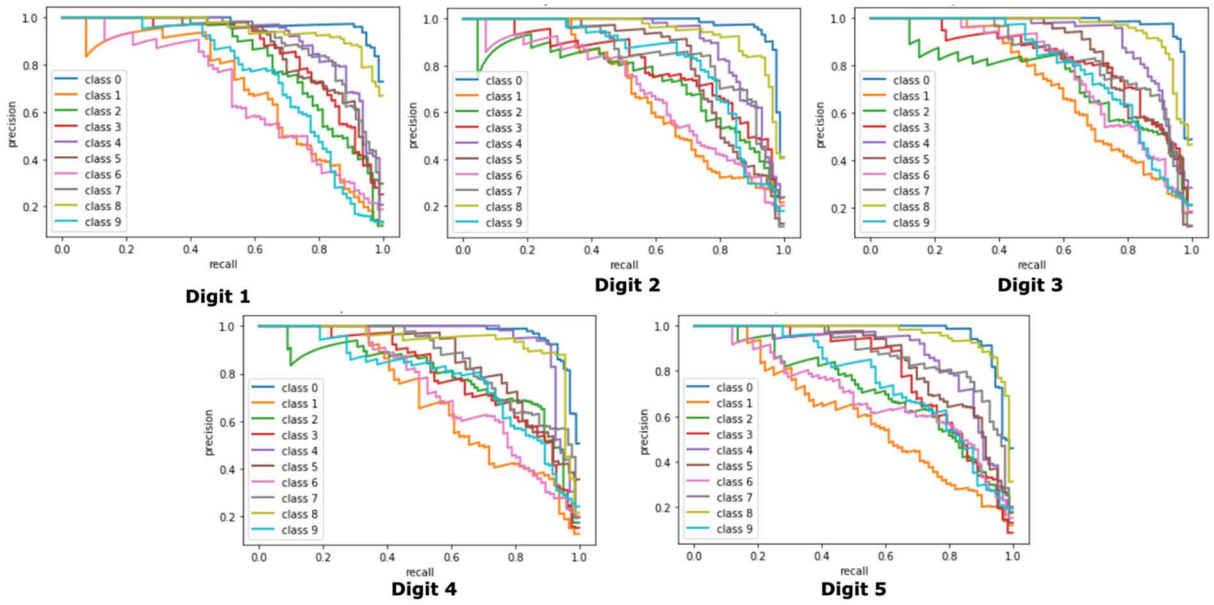

FIGURE 16. The PR curve of each class by digit position when generate 1,000 training images using the proposed augmentation method.

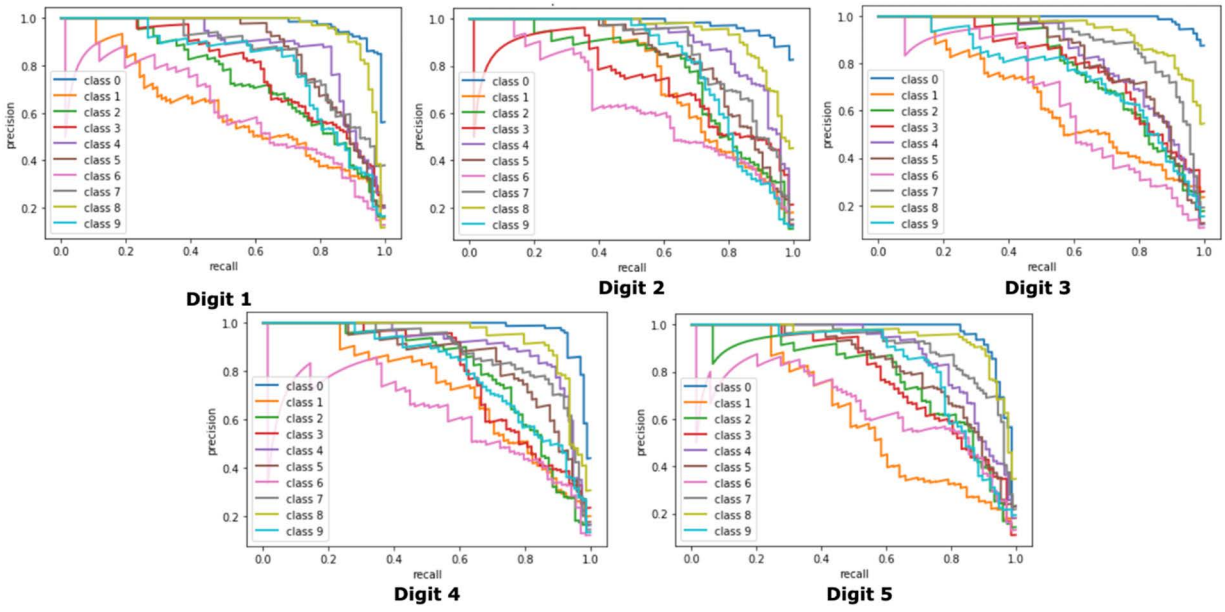

FIGURE 17. The PR curve of each class by digit position when generate 1,000 training images using the permutation method.

transferring the local appearance of a target digit to cope with the difficulty in collecting training data for digits requiring a long time to update. With this augmentation algorithm, we are able to control the frequency of each digit and construct a balanced training dataset from a few sample images. Moreover, our proposed data augmentation technique can automatically generate annotated images of transitional states between two consecutive numbers, which greatly improves the robustness of the recognition for analog rotating meters.

The experiments were conducted in various scenarios considering real-world AMR applications. All of the evaluation experiments confirm that our method improved the reading accuracy compared with using the original dataset, and the data expanded with the existing augmentation technique. The proposed augmentation technique can be combined with any state-of-the-art AMR framework for any real rotating meter image reading applications only by collecting a few images from the assumed environment.

The proposed method can be easily applied for reading the images sequence sent from a remotely installed camera. By using the coherence of numbers between consecutive frames, the reading accuracy can be improved even further.

APPENDICES

A. GENERATIVE NETWORK ARCHITECTURE

See Figure 14.

B. RECOGNITION NETWORK ARCHITECTURE

See Figure 15.

C. THE PR CURVE OF EACH CLASS BY DIGIT POSITION

See Figures 16 and 17.

REFERENCES

[1] J. Q. Azasoo and K. O. Boateng, "A retrofit design science methodology for smart metering design in developing countries," in *Proc. 15th Int. Conf. Comput. Sci. Appl.*, Jun. 2015, pp. 1–7.

[2] M. Spichkova and J. Van Zyl, "Application of computer vision technologies for automated utility meters reading," in *Proc. 15th Int. Conf. Softw. Technol.*, 2020, pp. 521–528.

[3] R. Laroca, A. B. Araujo, L. A. Zanlorensi, E. C. De Almeida, and D. Menotti, "Towards image-based automatic meter reading in unconstrained scenarios: A robust and efficient approach," *IEEE Access*, vol. 9, pp. 67569–67584, 2021.

[4] Y. Funayama, K. Nakamura, K. Tohashi, T. Matsumoto, A. Sato, S. Kobayashi, and Y. Watanobe, "Automatic analog meter reading for plant inspection using a deep neural network," *Artif. Life Robot.*, vol. 26, no. 2, pp. 176–186, May 2021.

[5] M. Rodr?guez, G. Berdugo, D. Jabba, M. Calle, and M. Jimeno, "HD-MR: A new algorithm for number recognition in electrical meters," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 22, no. 1, pp. 87–96, 2014.

[6] M. Cerman, G. Shalunts, and D. Albertini, "A mobile recognition system for analog energy meter scanning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Lecture Notes in Computer Science), vol. 10072. 2016, pp. 247–256, Accessed: Oct. 16, 2020, doi: 10.1007/978-3-319-50835-1_23.

[7] D. Shu, S. Ma, and C. Jing, "Study of the automatic reading of Watt meter based on image processing technology," in *Proc. 2nd IEEE Conf. Ind. Electron. Appl.*, May 2007, pp. 2214–2217.

[8] V. C. P. Edward, "Support vector machine based automatic electric meter reading system," Tech. Rep., 2013.

[9] J. Chi, L. Liu, J. Liu, Z. Jiang, and G. Zhang, "Machine vision based automatic detection method of indicating values of a pointer gauge," *Math. Problems Eng.*, vol. 2015, Oct. 2015, Art. no. 283629.

[10] Y. Fujita and Y. Hamamoto, "Automatic reading of an analogue meter using image processing techniques," *IEEJ Trans. Electron., Inf. Syst.*, vol. 129, no. 5, pp. 901–908, 2009.

[11] Y. Tang, C. W. Ten, C. Wang, and G. Parker, "Extraction of energy information from analog meters using image processing," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 2032–2040, Jul. 2015.

[12] L. Zhao, Y. Zhang, Q. Bai, Z. Qi, and X. Zhang, "Design and research of digital meter identifier based on image and wireless communication," in *Proc. Int. Conf. Ind. Mechatronics Autom.*, May 2009, pp. 101–104.

[13] Y. Zhang, S. Yang, X. Su, E. Shi, and H. Zhang, "Automatic reading of domestic electric meter: An intelligent device based on image processing and ZigBee/Ethernet communication," *J. Real-Time Image Process.*, vol. 12, no. 1, pp. 133–143, Jun. 2016.

[14] C. Li, Y. Su, R. Yuan, D. Chu, and J. Zhu, "Light-weight spliced convolution network-based automatic water meter reading in smart city," *IEEE Access*, vol. 7, pp. 174359–174367, 2019.

[15] R. Laroca, V. Barroso, M. A. Diniz, G. R. Gonçalves, W. Robson Schwartz, and D. Menotti, "Convolutional neural networks for automatic meter reading," 2019, *arXiv:1902.09600*.

[16] M. Patacchiola, P. Fox-Roberts, and E. Rosten, "Y-autoencoders: Disentangling latent representations via sequential encoding," *Pattern Recognit. Lett.*, vol. 140, pp. 59–65, Dec. 2020.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[18] G. Salomon, R. Laroca, and D. Menotti, "Deep learning for image-based automatic dial meter reading: Dataset and baselines," 2020, *arXiv:2005.03106*.

[19] R. C. da Silva Marques, A. Costa Serra, J. V. Ferreira Franca, J. O. Bandeira Diniz, G. Braz, J. D. Sousa de Almeida, M. I. Alves da Silva, and E. M. Garros Monteiro, "Image-based electric consumption recognition via multi-task learning," in *Proc. 8th Brazilian Conf. Intell. Syst. (BRACIS)*, Oct. 2019, pp. 419–424.

[20] Y. Liu, J. Liu, and Y. Ke, "A detection and recognition system of pointer meters in substations based on computer vision," *Measurement*, vol. 152, Feb. 2020, Art. no. 107333.

[21] S. Liao, P. Zhou, L. Wang, and S. Su, "Reading digital numbers of water meter with deep learning based object detector," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Lecture Notes in Computer Science), vol. 11857. 2019, pp. 38–49, Accessed: Nov. 17, 2020, doi: 10.1007/978-3-030-31654-9_4.

[22] Ultralytics. *YOLOv5*. Accessed: Jun. 4, 2021. [Online]. Available: https://github.com/ultralytics/yolov5

[23] R. Smith, "An overview of the tesseract OCR engine," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, vol 2, Sep. 2007, pp. 629–633.

[24] A. Nodari and I. Gallo, "A multi-neural network approach to image detection and segmentation of gas meter counter," *Proc. 12th IAPR Conf. Mach. Vis. Appl. (MVA)*, 2011, pp. 239–242.

[25] M. Vanetti, I. Gallo, and A. Nodari, "GAS meter reading from real world images using a multi-net system," *Pattern Recognit. Lett.*, vol. 34, no. 5, pp. 519–526, 2013.

[26] Y. Gao, C. Zhao, J. Wang, and H. Lu, "Automatic watermeter digit recognition on mobile devices," in *Proc. Int. Conf. Internet Multimedia Comput. Service* (Communications in Computer and Information Science), vol. 819. 2018, pp. 87–95.

[27] L. Li, Y. Li, K. Lian, X. Bian, K. Yang, and Y. Tian, "PGC-net: A light weight convolutional sequence network for digital pressure gauge calibration," *IEEE Access*, vol. 7, pp. 123280–123288, 2019.

[28] F. Yang, L. Jin, S. Lai, X. Gao, and Z. Li, "Fully convolutional sequence recognition network for water meter number reading," *IEEE Access*, vol. 7, pp. 11679–11687, 2019.

[29] S. Montazzolli and C. Jung, "Real-time Brazilian license plate detection and recognition using deep convolutional neural networks," in *Proc. 30th SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Oct. 2017, pp. 55–62.

[30] L. Gomez, M. Rusinol, and D. Karatzas, "Cutting Sayre's knot: Reading scene text without segmentation. Application to utility meters," in *Proc. 13th IAPR Int. Workshop Document Anal. Syst. (DAS)*, Apr. 2018, pp. 97–102.

[31] K. Koščevic and M. Subašic, "Automatic visual reading of meters using deep learning," in *Proc. Croatian Comput. Vis. Workshop*, Feb. 2019, pp. 1–6.

[32] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Goncalves, W. R. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the Yolo detector," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018.

[33] V. Kukreja, D. Kumar, A. Kaur, and Sakshi, "GAN-based synthetic data augmentation for increased CNN performance in vehicle number plate recognition," in *Proc. 4th Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Nov. 2020, pp. 1190–1195.

[34] G. Resende Goncalves, M. Alves Diniz, R. Laroca, D. Menotti, and W. Robson Schwartz, "Real-time automatic license plate recognition through deep multi-task networks," in *Proc. 31st SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Oct. 2018, pp. 110–117.

[35] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–14.

[36] C. Li, K. Xu, J. Zhu, and B. Zhang, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4089–4099.

[37] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 4, 2014, pp. 3581–3589.

[38] N. Siddharth, B. Paige, J.-W. van de Meent, A. Desmaison, N. Goodman, P. Kohli, F. Wood, and P. Torr, "Learning disentangled representations with semi-supervised deep generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5926–5936.

[39] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning," *Proc. 30th Conf. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 2172–2180, Accessed: Nov. 17, 2020. [Online]. Available: https://arxiv.org/abs/1606.03657
[40] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "CVAE-GAN: Fine-grained image generation through asymmetric training," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2745–2754.
[41] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, *arXiv:1511.05644*.
[42] A. L. Irina Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, and S. Mohamed. (2017). *B-Vae: Learning Basic Visual Concepts With a Constrained Variational Framework*. [Online]. Available: https://openreview.net/references/pdf?id=Sy2fzU9gl
[43] S. B. Damelin and N. S. Hoang, "On surface completion and image inpainting by biharmonic functions: Numerical aspects," *Int. J. Math. Math. Sci.*, vol. 2018, Feb. 2018, Art. no. 3950312.
[44] P. Patrick, M. Gangnet, and A. Blake, "Poisson image editing," in *Proc. ACM SIGGRAPH*, 2003, pp. 313–318.
[45] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.
[46] B. Yan, P. Fan, X. Lei, Z. Liu, and F. Yang, "A real-time apple targets detection method for picking robot based on improved YOLOv5," *Remote Sens.*, vol. 13, no. 9, p. 1619, 2021.
[47] J. Redmon. (2013). *Darknet: Open Source Neural Networks in C*. [Online]. Available: http://pjreddie.com/darknet/
[48] J. J. Weinman, A. Lidaka, and S. Aggarwal, "TensorFlow: Large-scale machine learning," in *GPU Computers*. G. Emerald, Ed. 2011, pp. 277–291.
[49] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.
[50] T. Y. Lin, M. Maire, S. Belongie, J. Hays, and P. Perona, "Microsoft COCO: Common objects in context," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Lecture Notes in Computer Science), vol. 8693. 2014, pp. 740–755.
[51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

NUNTIDA SRIPANUSKUL received the B.E. degree in computer engineering from Khon Kaen University (KKU), Thailand, in 2018. She is currently pursuing the M.E. degree in computer engineering with the University of Yamanashi, Japan. Her current research interests include deep learning, computer vision, and optical character recognition for the application to improve work efficiency and productivity in the industrial field.

PRAWIT BUAYAI received the B.E. and M.E. degrees in computer engineering from Khon Kaen University (KKU), Thailand, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree in computer engineering with the University of Yamanashi, Japan. His main research interest includes application of artificial intelligence technology to improve accessibility and productivity in agriculture.

XIAOYANG MAO (Member, IEEE) received the B.S. degree in computer science from Fudan University and the M.S. and Ph.D. degrees in computer science from The University of Tokyo. She is currently a Professor with the Department of Computer Science and Engineering, University of Yamanashi, Japan, and an Adjunct Professor with the College of Computer Science, Hangzhou Dianzi University, China. Her current research interests include image processing, visual perception, non-photorealistic rendering, and their applications to e-health. She received the Computer Graphics International Career Achievement Award in 2018.