Count Estimation With a Low-Accuracy Machine Learning Model

Yuichi Sei, *Member, IEEE*, and Akihiko Ohsuga, *Member, IEEE*

***Abstract*—Many Internet-of-Things (IoT) systems use machine learning techniques, such as deep neural networks. IoT systems can predict attributes, such as age, sex, car speed, human walking speed, and types of animals, using machine learning techniques. Although the functionality of machine learning is undeniable, the prediction accuracy is not always high. When a machine learning model is used to recognize several objects in an object counting system, the estimated count will have a significant error because of the accumulation of the recognition error of each object. In this study, a count estimation method that uses a confusion matrix generated in the training phase was proposed. The proposed method consists of an iterative Bayesian technique with the confusion matrix for count estimation and mitigating over-iterations technique for reducing estimated errors. The proposed method can be used even for a low-accuracy machine learning model. Experiments with synthetic and real data sets were conducted to demonstrate the functionality of the proposed method. The estimation errors of the proposed method were reduced by 64.3% in average compared to the baseline method in the experiments.**

***Index Terms*—Count estimation, deep neural network (DNN), Internet of Things (IoT), machine learning, prediction error.**

I. Introduction

STATISTICAL information of customers is important for marketing analysis [1]. For example, the information about how many males and females and how many young and adult people that visit a station can be useful for a store development plan. There are many machine learning approaches for people counting using Internet-of-Things (IoT) systems [2].

IoT systems can be used to recognize the number of people, animals, vehicles, and so on [3], [4]. The information about the number of dogs in a city is vital for rabies control [5]. Detecting and counting of wild animals is important for ecology and wildlife management [6]. Classifying vehicles into autobuses, automobiles, motorcycles, and trucks is useful for traffic noise analysis [7].

Manuscript received May 8, 2020; revised October 14, 2020; accepted November 12, 2020. Date of publication November 17, 2020; date of current version April 7, 2021. This work was supported in part by the Japan Society for the Promotion of Science KAKENHI under Grant JP17H04705, Grant JP18H03229, Grant JP18H03340, Grant JP18K19835, Grant JP19K12107, and Grant JP19H04113; and in part by JST, Precursory Research for Embryonic Science and Technology under Grant JPMJPR1934. *(Corresponding author: Yuichi Sei.)*

Yuichi Sei is with the Department of Informatics, Graduate School of Informatics and Engineering, University of Electro-Communications, Tokyo 1828585, Japan, and also with JST, PRESTO, Kawaguchi 3320012, Japan (e-mail: seiuny@uec.ac.jp).

Akihiko Ohsuga is with the Department of Informatics, Graduate School of Informatics and Engineering, University of Electro-Communications, Tokyo 1828585, Japan (e-mail: ohsuga@uec.ac.jp).

Digital Object Identifier 10.1109/JIOT.2020.3038273

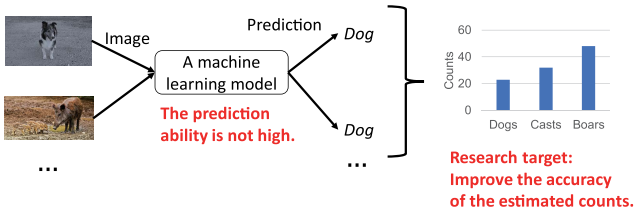

Fig. 1. Research objective.

IoT systems equipped with sensors and/or cameras can recognize objects by using machine learning techniques, such as deep neural networks (DNNs). With extremely careful tuning, a machine learning model is expected to recognize objects with high accuracy. However, in some cases, the accuracy is low. For example, in the case of age estimation systems that use facial images to determine the age of people, especially when the images are noisy or have the back view of people [8].

Although it is important to improve the recognition accuracy of each machine learning model, it is not always easy because of the noisy and blur data obtained by the sensors and the cameras.

In this study, it is assumed that the IoT systems count several types of objects using machine learning models that recognize each object with a certain accuracy. Using a confusion matrix, a method was proposed to improve the accuracy of object counting systems. A confusion matrix shows the classification errors in the training phase of a machine learning model. The proposed method can be used even for a low-accuracy machine learning model (see Fig. 1).

The remainder of this article is structured as follows. Section II, presents an application and make some assumptions. In Section III, some related methods are highlighted. Section IV, introduces the proposed design and its mechanisms. In Section V, simulation results on some synthetic and real data sets are presented. Section VI, discusses several design issues of the proposed method. Finally, the conclusions are presented in Section VII.

II. Background

A. Motivating Example

Example 1: The overabundance of deer has been a big problem in some parts of the world as it is possible cause of tree bark stripping and outbreaks of wildlife diseases and zoonoses [9]. To control this problem, it is imperative to study the information on deer population. The camera trap technology, where pictures of wild animals are automatically

captured using motion sensor cameras, has been used for animal population survey.

According to a survey conducted in Nakanoshima Island in Japan, on average, approximately 400 deer per month are photographed. In a year-long survey, researchers counted deer by considering sex and age class and clustered 4809 deer in total [10].

Although the camera trap technology is automated, images are generally classified manually. This manual classification is a limitation for the animal population survey. To tackle this limitation, several studies have started using machine learning technologies to identify species [11]–[13]. However, the accuracy of machine learning may be low in case of morphologically similar species or when the number of training samples is small [14]. Therefore, a method that can automatically count species from images with high accuracy is required for the case when machine learning models cannot identify each species with high accuracy.

Iijima *et al.* [9] showed that the mean percentage of the counted deer for each year increased from 9.0% in their annual experiment. Ikeda *et al.* [10] analyzed deer numbers, which fluctuate about 10% each month. To analyze a difference of a few percent to 10%, it is necessary for high-precision statistics.

On the other hand, because a lot of motion sensor cameras are necessary for a wide region, it is difficult to prepare expensive motion sensor cameras. When only low-quality motion sensor cameras are available, the accuracy of machine learning model becomes very low. Even in such case, the accuracy of the count estimation can be increased by the method in this work.

B. Assumptions

An IoT system can recognize target objects by using a machine learning model with possible low accuracy.

Many sample images can be used for training a machine learning model. For example, ImageNet [15] contains more than 14 000 000 images of various objects. Using these samples, the IoT system manager can train the machine learning model; it collects the image samples and conduct transfer learning. The IoT manager prepares a pretrained general-purpose machine learning model and then retrains it with the training samples. Even if the number of training samples is small, the accuracy of the retrained machine learning model is expected to be higher than that of a machine learning model without transfer learning.

In general, a confusion matrix is generated after training the machine learning model. A confusion matrix is one of the most important metrics used to evaluate classification performance [16]–[19]. It is a 2-D matrix and each cell represents the number of actual and predicted classifications performed by a machine learning model. Fig. 2 shows the structure of a confusion matrix. Let f be the number of classes and C_i represent the ith class, the value $m_{i,j}$ represents the number of times that the true class is C_i and the predicted class is C_j. It is assumed that the IoT manager can obtain the confusion matrix generated from several image samples.

		Predicted class				
		C_1	C_2	C_3	...	C_f
Actual class	C_1	$m_{1,1}$	$m_{1,2}$	$m_{1,3}$	...	$m_{1,f}$
	C_2	$m_{2,1}$	$m_{2,2}$	$m_{2,3}$	...	$m_{2,f}$
	C_3	$m_{3,1}$	$m_{3,2}$	$m_{3,3}$	...	$m_{3,f}$
	...	...	...	...	...	...
	C_f	$m_{f,1}$	$m_{f,2}$	$m_{f,3}$	...	$m_{f,f}$

Fig. 2. Confusion matrix.

III. Related Works

A. Object Identification Using Machine Learning

Machine learning is an artificial intelligence technique that can be used for various tasks, such as classification, regression, and generation of new objects.

In the recent decade, DNNs have been widely studied. Although DNNs require considerable computational power and a large quantity of samples for training, they can achieve higher accuracy in many applications compared to other machine learning models [20]. For example, DNNs can recognize animals [11], human poses [21], plants [22], insects [23], and so on.

However, few reliable DNN models exist for low-quality data [24]. For low quality or small number of training samples, the accuracy of DNNs decreases. Moreover, even when a large number of high-quality data are used in the training phase, the identification accuracy greatly decreases when only low-quality sensors or cameras are available.

B. Improving Classification Accuracy Based on Confusion Matrix

Visa *et al.* [25] proposed a feature selection method based on a confusion matrix. Feature selection is one of the main concerns of machine learning. If a training sample has many features, by selecting several important features, the accuracy of a machine learning model can be improved. Although a DNN can automatically extract important features, feature selection is still done manually. The method proposed by Visa *et al.* could improve the classification accuracy using its confusion matrix.

Deng *et al.* [26] proposed a construction method of basic probability assignment (BPA) using a confusion matrix. Construction of BPA is necessary for Dempter–Shafer evidence theory, which can be used for reasoning with uncertainty and large-complex systems. Using multiple classifiers and confusion matrixes, Deng *et al.* constructed BPA and showed that the classification accuracy of each object could be increased.

Ohsaki *et al.* [27] developed an imbalanced data classifier based on a confusion matrix for their target of classification of two classes. They showed its high performance by extensive experiments on benchmark imbalanced data sets.

All the above-mentioned studies aimed to improve the classification accuracy of *each sample*. By contrast, this work aims

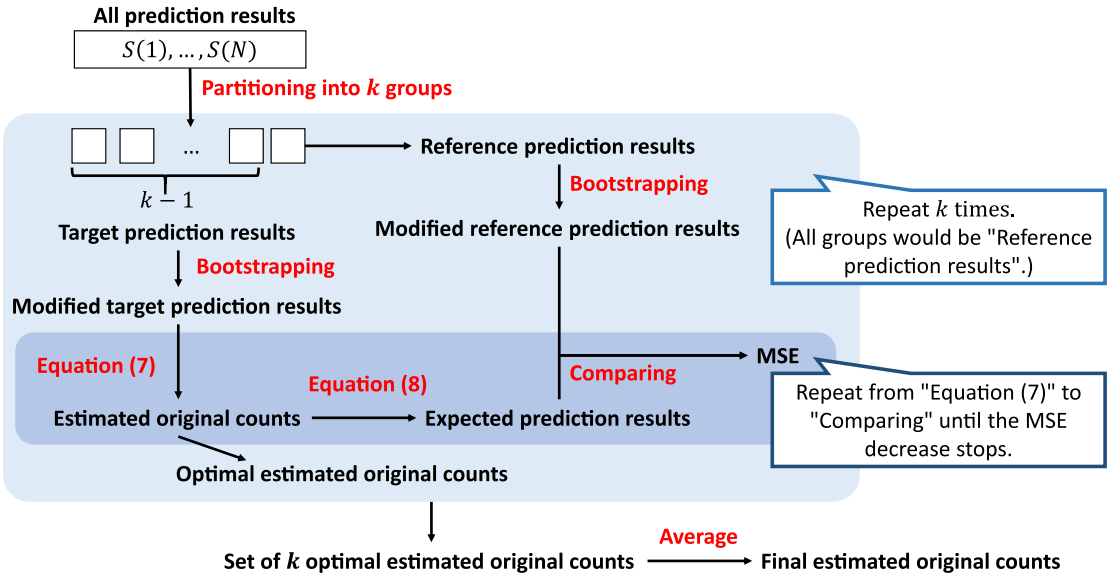

Fig. 3. Overall procedures of the proposed method.

to improve the accuracy of *total counts* by using a machine learning model that can classify each sample. We can use these existing methods as machine learning models that classify each object, and our proposed method can improve the accuracy of total counts using their outputs.

C. Estimation Using Manually Labeled Small Samples

In case machine learning techniques are not used for identifying objects, each object in each image is identified manually and how many times each object appears is estimated by labeling a small portion of the images manually.

Krejcie proposed a method of determining sample size for a representative of a given population [28]. If the population is sufficiently large, the following equation applies:

$$n_0 = \gamma^2 \frac{p(1-p)}{d^2} \tag{1}$$

where n_0, p, d, and γ represent the sample size, the population proportion, the desired degree of accuracy expressed as a proportion, and the abscissa of the standard normal distribution that cut off an area of the desired confidence level, respectively.

If the population is small, the following equation applies:

$$n_0' = \frac{n_0}{1 + \frac{n_0 - 1}{N}} \tag{2}$$

where N represents the population size [29].

For example, assuming that $N = 1000$, $p = 0.5$, $d = 0.05$, and $\gamma = 1.96$, which means the desired confidence level is 0.95. In this case, the necessary sample size n_0' is 278. That is, by manually labeling 278 images, we can estimate each number of labels of 1000 images. Although we can reduce labeling tasks, 278 labeling tasks should be conducted manually. If many images are obtained every day or every month, the accumulated costs cannot be ignored.

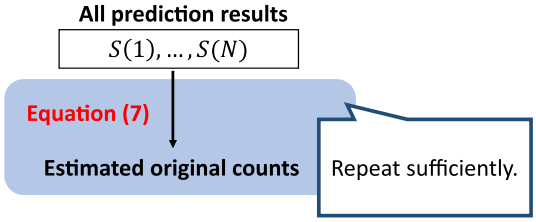

Fig. 4. Overall procedures of the proposed method without mitigating over iterations.

IV. Proposed Method

A. Overview

Our proposed method uses Bayesian technique [30], [31] for count estimation. In this technique, a probability matrix that shows how the machine learning models misclassify objects is generated. The process of generating the probability matrix is in Section IV-B. In Section IV-C, the Bayes-based estimation method is proposed. The Bayes-based method repeats an internal process many times. It is difficult to determine when the iteration should be stopped, so the Bayes-based method suffers from over iterations. Therefore, a method was proposed to mitigate over iterations (MOs) in Section IV-D.

The overall procedures of our proposed method and our proposed method without MO are shown in Figs. 3 and 4, respectively. In the figures, (7) uses the probability matrix described in Section IV-B and the iterative technique described in Section IV-C.

B. Generation of Probability Matrix

Let N and f represent the number of data samples and the number of classes, respectively. A confusion matrix can be generated as shown in Fig. 2.

First, $\mathcal{P}$ is created to represent each conditional probability $\mathcal{P}(i, j)$ of the predicted class i under the true class j. If there are many classes and the number of sample for creating the

confusion matrix is small, the values of $m_{i,j}$ for some i and all j might become zeros. In this case, $\mathcal{P}(i,j)$ cannot be obtained for those is. To avoid this issue, $\mathcal{P}(i,i)$ is set to be 1.0 if $m_{i,j}$ are zeros for $j = 1, \ldots, f$.

As a result, each $\mathcal{P}(i,j)$ is calculated as follows:

$$\mathcal{P}(i,j) = \begin{cases} \dfrac{m_{i,j}}{\sum_{k=1}^{f} m_{i,k}}, & \left(\sum_{k=1}^{f} m_{i,k} \neq 0\right) \\ 1, & \left(\sum_{k=1}^{f} m_{i,k} = 0 \quad \& \quad i = j\right) \\ 0, & (\text{otherwise}). \end{cases} \tag{3}$$

C. Bayes-Based Estimation

Let $\mathcal{C} = \{C_1, \ldots, C_f\}$ represent the set of classes of the target objects. The prediction results of a machine learning model is represented by $\mathcal{S} = [S(1), S(2), \ldots, S(N)]$, where $S(i) \in \mathcal{C}$.

Let X and Y represent the random variables of the true and the predicted classes, respectively. $P(X = C_i | Y = C_j)$ represents the conditional probability that the true class is C_i under the condition that the predicted class is C_j. Let $\mathcal{X}_i$ and $\mathcal{Y}_i$ represent the true number of objects of C_i and the times a machine learning model identifies target objects as C_i, respectively. Each value of $\mathcal{Y}_i$ is calculated as follows:

$$\mathcal{Y}_i = \left|\{S(j) \mid S(j) \in \mathcal{S} \wedge S(j) = C_i\}\right|. \tag{4}$$

Our aim is to obtain each value of $\mathcal{X}_i$ for $i = 1, \ldots, f$. According to Bayes' theorem

$$\begin{aligned} P(X = C_\zeta | Y = C_z) &= \frac{P(Y = C_z | X = C_\zeta) P(X = C_\zeta)}{P(Y = C_z)} \\ &= \frac{\mathcal{P}(\zeta, z)\mathcal{X}_\zeta}{\sum_k \mathcal{X}_k \mathcal{P}(k, z)}. \end{aligned} \tag{5}$$

By contrast, we have

$$\mathcal{X}_\zeta = N \times P(Y = C_z) \times \sum_z P(X = C_\zeta | Y = C_z). \tag{6}$$

Let $\hat{\mathcal{X}}_i$ represent the estimated value of $\mathcal{X}_i$. Therefore, by repeating the following insertion, the estimated value is obtained as follows:

$$\hat{\mathcal{X}}_\zeta^{T+1} \leftarrow \sum_z \mathcal{Y}_z \frac{\mathcal{P}(\zeta, z)\hat{\mathcal{X}}_\zeta^T}{\sum_k \hat{\mathcal{X}}_k^T \mathcal{P}(k, z)} \tag{7}$$

where T represents the Tth iteration and $\hat{\mathcal{X}}_\zeta^T$ represents the estimated value of $\hat{\mathcal{X}}_\zeta$ at the Tth iteration. $\mathcal{Y}_\zeta$ is used as the initial value of $\hat{\mathcal{X}}_\zeta^1$.

D. Mitigating Over Iterations

It is difficult to determine when the iteration of (7) should be terminated. In the existing studies based on the Bayesian method, such as [32], it is suggested that the iterations should be large enough to achieve high accuracy, but increased iterations implies a higher calculation cost. However, in our scenario, we found that excessive number of iterations led to poor accuracy.

In DNNs, an epoch is a complete pass of the training data to be learned by a learning model. Because too few epochs or too many epochs can lead to low accuracy, the number of epochs should be optimally adjusted to achieve high accuracy.

In DNNs, the data samples are generally divided into training data, validation data, and test data. The number of epochs is determined such that maximum accuracy of the validation data is achieved. Note that in DNNs for supervised learning, data samples are labeled. Thus, the estimation accuracy of the validation data set can be measured. The data samples are sometimes divided into only the training data and the test data. In this case, the number of epochs is determined such that the accuracy of the test data is maximum. However, in our scenario, as labeled data are not obtained, we cannot know the accuracy of any data samples.

In our proposed method, the elements of the set $\mathcal{S}$ of the prediction results of a machine learning model are divided into k groups. Based on the set of $k-1$ groups, the process of (7) is executed. The prediction results of the set of the $k-1$ groups and that of the remaining group are referred to as the target prediction results and the reference prediction results, respectively.

During the execution process of (7), the difference between the count distribution of the reference prediction results and the *expected prediction results* is measured. The expected prediction results are the expected count distribution if the estimated $\hat{\mathcal{X}}$ matches perfectly with $\mathcal{X}$. The expected prediction results are calculated by generating $\hat{\mathcal{X}}$ and $\mathcal{P}$.

More specifically, let α_z be the expected number of times a machine learning model predicts C_z when the true count of class C_j is $\hat{\mathcal{X}}_j$, the value of α_z is represented as

$$\alpha_z = \sum_j \hat{\mathcal{X}}_j \mathcal{P}(j, z). \tag{8}$$

Theorem 1: When the estimated $\hat{\mathcal{X}}$ is close to the true value $\mathcal{X}$, α_z is expected to be close to $\mathcal{Y}_z$ for all z.

Proof: $\mathcal{P}(j, z)$ represents the conditional probability that the output of the machine learning model is C_z under the condition that the true class is C_j. Therefore, the expected value of $\mathcal{Y}_z$ is calculated as follows:

$$E[\mathcal{Y}_z] = \sum_j \mathcal{X}_j \mathcal{P}(j, z). \tag{9}$$

Therefore, the difference between α_z in (8) and $E[\mathcal{Y}_z]$ in (9) becomes small if $\hat{\mathcal{X}}_j$ is close to $\mathcal{X}_j$. ■

Thus, a highly accurate estimation of $\hat{\mathcal{X}}_j$ can be obtained by reducing the difference between α_z and $E[\mathcal{Y}_z]$. The value of $E[\mathcal{Y}_z]$ cannot be obtained because $\mathcal{X}_j$ is an unknown value; however, the actual $\mathcal{Y}_z$ is obtained through the machine learning model's prediction. Hence, by reducing the difference between α_z and $\mathcal{Y}_z$, over iterations can be mitigated. In the proposed method, α_z and $\mathcal{Y}_z$ are calculated from the target prediction results and the reference prediction results, respectively. The ratio of the number of samples of the target prediction results and the number of samples of the reference prediction results is set to be $(k-1)$ versus 1 in the proposed method. Because (7) achieves high accuracy when the number of samples is large, the number of samples of the

Algorithm 1 Bootstrapping Method for Creating N Samples From n Samples

Input: A set of n samples $S = \{s_1, \ldots, s_n\}$, the target number of samples N.
Output: A set of bootstrapped N samples S'.

```
S' ← {}.
for i = 1, ..., N do
    r ← Rand(1, n).
    S' ← S' ∪ {s_r}.
end for
return S'
```

target prediction results is set to be a large value. Searching the optimal ratio remains an issue to be addressed in future work.

Therefore, in each iteration, the difference between the two data distributions is measured—$[\alpha_1, \ldots, \alpha_f]$ is calculated on the basis of the target prediction results and $[\mathcal{Y}_1, \ldots, \mathcal{Y}_f]$ from the reference prediction results. the iteration is terminated when the value of the difference is minimal.

As a metric of the difference, mean-squared error (MSE) is defined as follows:

$$\mathrm{MSE} = \sum_{i=1}^{f} (\alpha_i - \mathcal{Y}_i)^2. \tag{10}$$

Here, to match the sizes of the target prediction results and the reference prediction results, the bootstrapping method is used. The sizes of the target prediction results and the reference prediction results are $N(k-1)/k$ and N/k, respectively. The target prediction results and the reference prediction results are recreated. In details, N samples were randomly selected from the target prediction results with repetition; the selected N data samples were taken to be the new target prediction results. In the same way, new reference prediction results were created.

Algorithm 1 describes the bootstrapping method which creates N samples from n samples. The function $\mathrm{Rand}(1, n)$ returns a random integer value from 1 to n.

In early iterations, the difference kept decreasing. After several iterations, the difference began to increase and at this point the iterations were terminated.

The process was repeated k times with each of the k groups used exactly once as the reference prediction results.

Finally, the average of the k set of the estimated counts was calculated. The resulted values are the output of our proposed method.

V. Evaluation

A method that counts objects simply based on the predictions of a machine learning model is referred to as a baseline. In this section, our proposed method and without MO and the baseline method were all evaluated.

A. Synthetic Data Set

First, MSEs were evaluated using synthetic data sets. Each value of $\mathcal{X}_i$ was randomly determined to satisfy $\sum_i \mathcal{X}_i = N$.

Both f and N were set as follows: $f = 2, 5, 10, 100$, $N = 100, 1000, 10\,000$. The accuracy of the object identification machine learning model was set from 0.3 to 1.0. When f was 2, the accuracy of the learning model was set from 0.6 to 1.0 because the minimum value should be larger than 0.5. Every parameter setting was executed 10 times and its average MSEs are shown in Fig. 5.

When the accuracy of the machine learning model was approximately 1.0, the MSEs of all methods were close to 0. By contrast, when the accuracy of the learning model was low, the MSEs of the baseline method increased, especially when the number of samples was large. However, the MSEs of the proposed method without MO increased, especially when the number of samples was small.

When the number of samples was small and the number of classes was large, the MSEs of the proposed method were greater than those of the baseline method. However, the difference was relatively small. In almost all parameter settings, the MSEs of the proposed method were the smallest compared to the other methods.

B. Real Data Set

In this section, real data sets and real prediction results of the machine learning models were used.

The aim is not to improve the accuracy of each machine learning model itself, but to improve the accuracy of count estimation even if the accuracy of each machine learning model is relatively low. Although the accuracy of each machine learning model constructed for the evaluations could be improved, the parameters and the structures of the constructed machine learning models were not tuned.

Several famous data sets were used, including CIFAR-10, CIFAR-100 [33], ANIMAL-10,[1] Dogs&Cats,[2] WIKI, and IMDb [34] data sets.

CIFAR-10 data set consists of 60 000 images in ten classes of various objects (e.g., airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck), whereas CIFAR-100 data set consists of 60 000 images in 100 classes. ANIMAL-10 data set contains 18 000 images in ten classes of animals (e.g., dog, horse, elephant, butterfly, chicken, cat, cow, sheep, spider, and squirrel). Dogs&Cats data set originally contains images of 1000 dogs, 1000 cats, and 1000 pandas; however, only images of dogs and cats were used because ANIMAL-10 does not contain images of pandas. As described below, Dogs&Cats data set was used with ANIMAL-10 data set to identity dogs and cats. WIKI data set and IMDb data set were created by crawled profile images from Wikipedia and IMDb, with 62 328 and 460 723 face images, respectively.

As summarized in Table I, six experiments were conducted. Because images obtained in a real situation might be out of focus, blur filtering was conducted on several data sets.

In the first experiment, CIFAR-10 data set was used to train a DNN with 40 000 images and a confusion matrix with

[1] https://www.kaggle.com/alessiocorrado99/animals10

[2] https://www.kaggle.com/ashishsaxena2209/animal-image-datasetdog-cat-and-panda

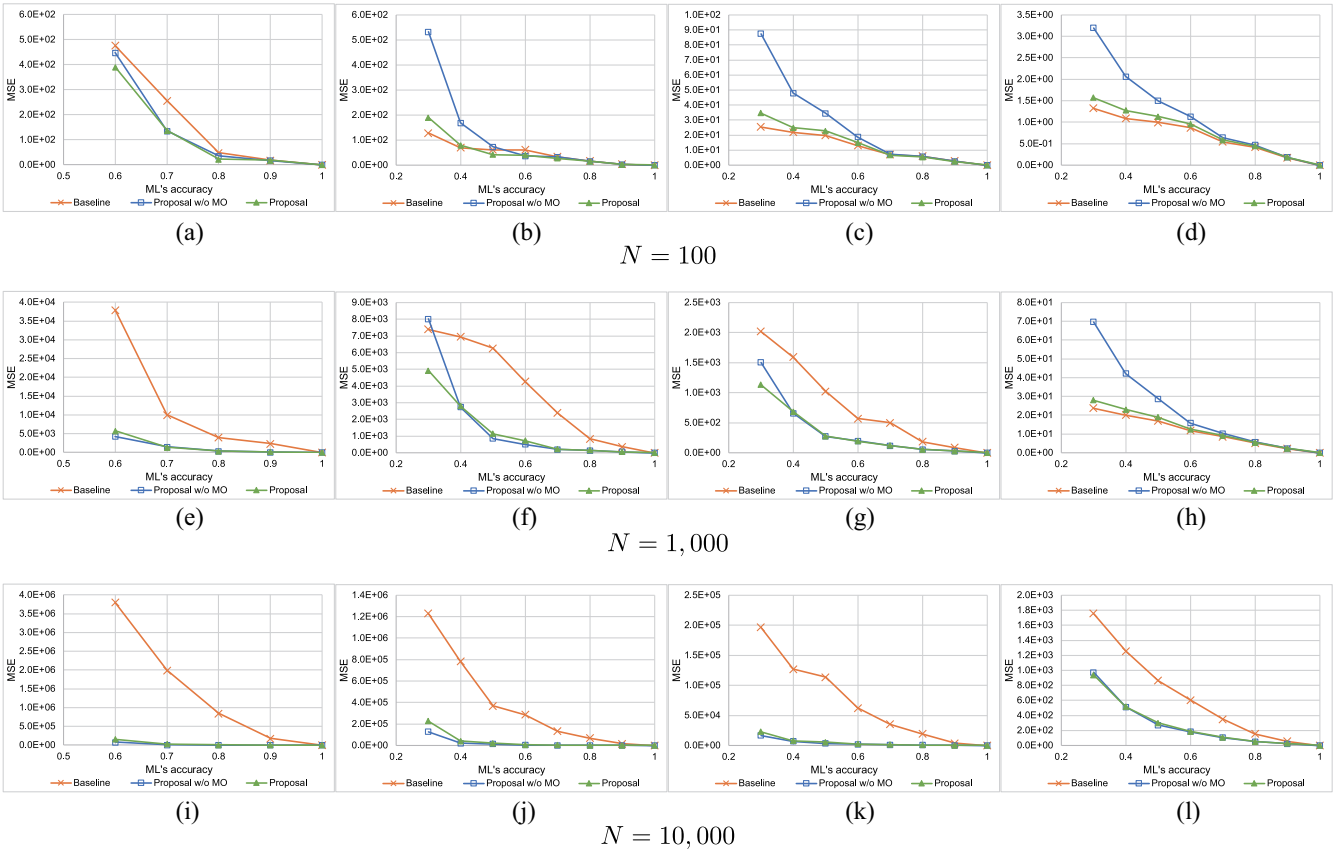

Fig. 5. Mean-square errors of synthetic data. (a), (e), and (i) $f = 2$. (b), (f), and (j) $f = 5$. (c), (g), and (k) $f = 10$. (d), (h), and (l) $f = 100$.

TABLE I
DESCRIPTION OF DATA SETS

Experiment ID	Train/validation data set	Test data set	Target objects	# test samples	# target classes	ML's accuracy
1	CIFAR-10	CIFAR-10	Various objects	10,000	10	0.74
2	CIFAR-100	CIFAR-100	Various objects	10,000	100	0.39
3	ANIMALS-10	ANIMALS-10	Animals	2,618	10	0.84
4	ANIMALS-10	Dogs&Cats with blur filtering	Animals	2,000	10	0.73
5	WIKI	IMDb with blur filtering	Person's gender	10,000	2	0.92
6	WIKI	IMDb with blur filtering	Person's age	10,000	21	0.31

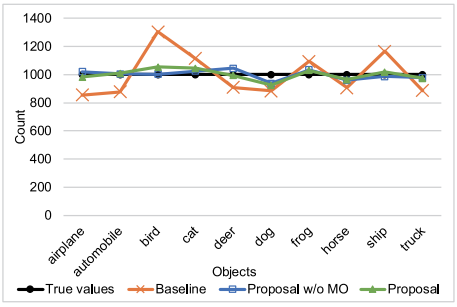

Fig. 6. Result of Experiment 1 (CIFAR-10).

10 000 images was obtained. Finally, the estimation accuracy was tested using 10 000 images.

The true and estimated counts are shown in Fig. 6. Although a bar graph is suitable for representing count values, a line graph was used for the sake of readability. The true values were all 1000 with each class. Because the accuracy of the DNN model is not perfect, the estimated count values of the baseline method did not match the true values. However, our proposed method could achieve a high accuracy because it considers the misclassifying information of the validation phase. In this experiment, the technique of MO described in Section IV-D does not affect the accuracy.

CIFAR-100 data set was used for the second experiment. Fig. 7 shows the estimation results. The true values were all 100 with each class. There was a large disparity in the estimated counts of the baseline method and our proposed method without MO. Because the number of images of each class was smaller than that of CIFAR-10, the over iterations of our proposed method without MO possibly occurred. Although the estimated values of our proposed method did not match the true values perfectly, the accuracy was the best compared with other methods.

With the SODA machine learning model where the input images are clear and fine and very similar to the training data, there is little need to use our proposed method. However, it is not always easy because of the noisy and blur data obtained by

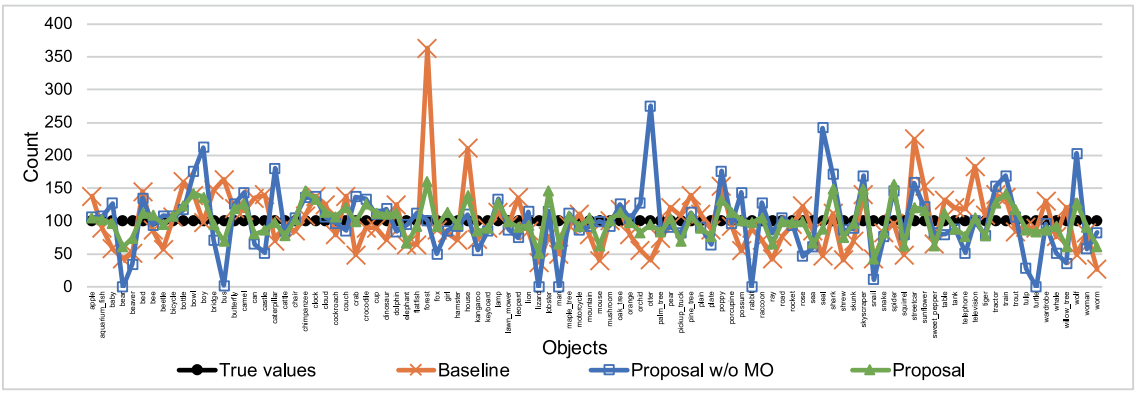

Fig. 7. Result of Experiment 2 (CIFAR-100).

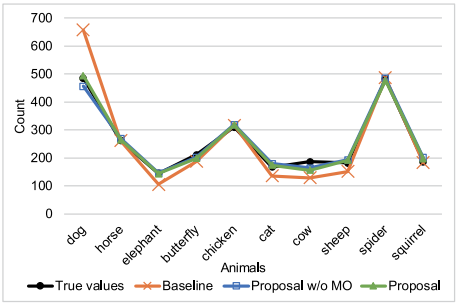

Fig. 8. Result of Experiment 3 (ANIMALS-10).

(a) (b)

Fig. 9. Sample image with the blurring effect. (a) Original image. (b) Blur image.

the sensors and the cameras. Moreover, if we need to identify objects that are not in the training data of the SODA machine learning model, a machine learning model should be created from the beginning or the transfer learning should be conducted based on the SODA machine learning model. In this case, the accuracy of the machine learning model might be very low. Our proposed method aims to improve the accuracy of the count estimation even if we have only a low-accurate machine learning model.

In the third experiment, ANIMAL-10 data set was used. The results are shown in Fig. 8. The accuracy of the DNN was relatively high (i.e., 0.84). Hence, the accuracies of all the methods were higher than those of the first experiment.

In the first three experiments, the training images, the validation images (i.e., images used for creating a confusion matrix), and the test images were obtained from the same data set in each experiment. However, in a real situation, a machine learning system is applied to unknown images. To confirm that our proposed method can be used in a real situation, different data sets were used for the training/validation phase and for the test phase.

Moreover, while applying a trained machine learning model to an IoT system, the input samples of the machine learning model might contain more noise than the samples used in the training phase. To confirm that our proposed method can be used in such situation, spatial filtering was applied to the images used in the last three experiments. In detail, the kernel was constructed as follows:

$$K = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{11}$$

Each original image is convolved with the kernel and each pixel (i, j) of the resulted image is calculated as follows:

$$\mathrm{dst}(i,j) = \sum_{-1 \le i', j' \le 3} K(i', j') \times \mathrm{src}(i + i', j + j') \tag{12}$$

where $dst(i, j)$ and $src(i, j)$ represent the ith column and the $j$the row of the resulted and the original images, respectively, and $K(i, j)$ represents the term on the ith column and the jth row of the kernel. This filtering has a blurring effect.

The sample image with the effect of the blur filtering is shown in Fig. 9.

In the fourth experiment, ANIMALS-10 was used for training and validation while Dogs&Cats was used for blur filtering for the test. Because the test images only contained dogs and cats, the true values were 1000 with the dog and cat classes and 0 with other classes. The results are shown in Fig. 10. The baseline method overestimated the number of dogs and underestimated the number of cats. This trend is the same as that of the result of the third experiment (that used only ANIMALS-10 for training, validation, and testing). However, the proposed method estimated the count values of dogs and cats with high accuracy.

Fig. 11 shows the probability matrix of the machine learning model used in Experiments 3 and 4. The machine learning model incorrectly predicted "dog" many times even when the true class was not dog. This is the reason why the baseline method over estimated the number of dogs in Figs. 8 and 10. By contrast, because our proposed method used the

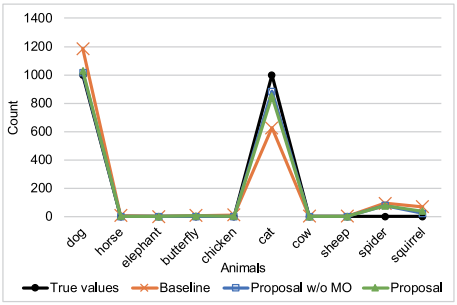

Fig. 10. Result of Experiment 4 (ANIMALS-10 and dogs&cats).

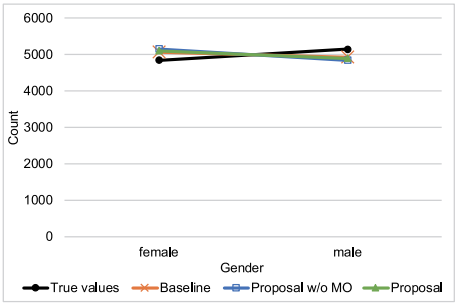

Fig. 12. Result of Experiment 5 (WIKI and IMDb for gender estimation).

True class \ Predicted class	dog	horse	elephant	butterfly	chicken	cat	cow	sheep	spider	squirrel
dog	0.883	0.031	0.002	0.001	0.024	0.022	0.008	0.009	0.007	0.013
horse	0.108	0.839	0.009	0.001	0.006	0.000	0.016	0.013	0.007	0.002
elephant	0.213	0.040	0.678	0.002	0.019	0.000	0.016	0.017	0.002	0.014
butterfly	0.044	0.001	0.000	0.884	0.009	0.001	0.000	0.001	0.054	0.005
chicken	0.084	0.002	0.000	0.004	0.887	0.001	0.001	0.002	0.007	0.011
cat	0.235	0.000	0.000	0.006	0.013	0.679	0.001	0.009	0.013	0.042
cow	0.188	0.059	0.007	0.000	0.021	0.000	0.666	0.048	0.004	0.007
sheep	0.213	0.023	0.014	0.001	0.032	0.000	0.045	0.666	0.003	0.003
spider	0.026	0.000	0.000	0.011	0.003	0.001	0.000	0.002	0.948	0.010
squirrel	0.157	0.000	0.000	0.003	0.015	0.016	0.000	0.008	0.028	0.773

Fig. 11. Probability matrix of the machine learning model used in Experiments 3 and 4.

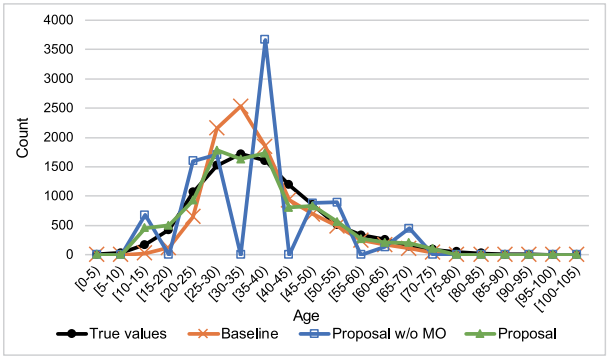

Fig. 13. Result of Experiment 6 (WIKI and IMDb for age estimation).

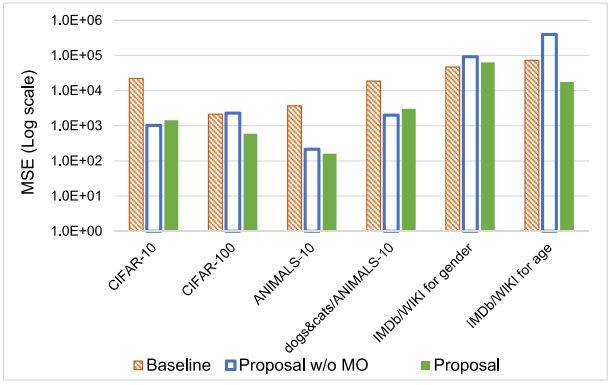

Fig. 14. Overall results of MSEs.

information of the probability matrix, the estimated counts were more accurate than the baseline method.

In the last two experiments (i.e., Experiments 5 and 6), we used WIKI data set for the training and validation phases and IMDb data set for the test phase. The experiments were conducted to estimate people's gender and age, respectively.

For the age estimation, the DNN was trained for 101 classification from 0 to 100 based on [34]. Because a general DNN for classification outputs each probability for each class, the predicted age can be obtained by calculating the expected value. Specifically, assuming p_i represent the probability that the person in the image is i-years old, the predicted value is calculated as follows:

$$\text{Predicted age} = \sum_{i=0}^{100} i \times p_i. \tag{13}$$

This calculation method is the same as [34]. The results of gender estimation and age estimation are shown in Figs. 12 and 13, respectively.

Based on Fig. 12, the accuracies of all the methods were considerably high for gender estimation. Because there were just two classes and the accuracy of the machine learning model was high (i.e., 0.92), it was probably an easy task for all the methods. Although our main target is an IoT system with a low-accuracy machine learning model, our proposed method does not underperform in the baseline method.

According to Fig. 13, the class (i.e., 30–35) is the largest class in the age distribution. The true distribution is a long-tailed normal distribution. The estimation result of our proposed method was very similar to the true distribution. The result of the baseline method overestimated the range from 20-years-old to 40-years-old. The estimation result of our proposed method without MO is extremely different from the true distribution as over iterations may have occurred in this experiment.

The overall results of the MSEs are shown in Fig. 14. In the fifth experiment, the MSE of the baseline is smaller than that of the proposed method. However, the difference is extremely small as shown in Fig. 12. In other experiments, the MSEs of the proposed method are smaller than those of the baseline method. In the first and fourth experiments, the proposed method without MO outperformed the proposed method with MO. However, the differences were small. In other experiments, the proposed method with MO achieved considerably higher accuracy than the proposed method without MO.

VI. Discussion

In our experiments, data augmentation was not conducted. By applying data augmentation techniques, such as flipping images and adding noises, the accuracy of a machine learning model can be improved [35]. By conducting data augmentation, the accuracy of not only the machine learning model but also the confusion matrix is expected to improve.

If the training samples are considerably different from the test samples, the MSEs of the proposed method would increase. However, in this case, the MSEs of a baseline method will also increase. Our experimental results demonstrated that our proposed method is robust to the difference between training and test samples.

In this study, machine learning models for classification were targeted. However, other important machine learning task is regression. A general DNN does not provide the probability of the predicted value for regression, although it provides the probability of each class for classification. However, several machine learning techniques can provide a probability distribution for regression tasks; for example, the Gaussian process regression (GPR) [36], [37]. By using the GPR or combining a DNN with the GPR, our proposed method is expected to be used for regression tasks. To confirm this consideration, further experiments will be conducted in the future.

VII. Conclusion

Many IoT systems use machine learning models to identity objects. Although the identification accuracy has been improving, it is not always perfect. Scenarios where the IoT system aims to count objects, such as specific animals were targeted. Even when the accuracy of the machine learning model that is used to identify objects is not high, our proposed method can greatly improve the accuracy of the estimated counts of the target objects. In the proposed method, a machine learning model predicts a true label of each sample in the usual way. When the prediction ability of the machine learning model is not high, iterative Bayesian technique with the confusion matrix is used for enhancing the accuracy of the count estimation. The iterative Bayesian technique suffers from over-iterations problem; therefore, techniques for mitigating over iterations was also discussed. The over-iterations-mitigating technique calculates and reduces the difference between the target prediction results and the reference prediction results. The results of the experiments conducted using synthetic and real data sets as well as real outputs of the prediction results of DNNs confirmed that our proposed method outperforms the baseline method. The estimation errors of the proposed method were reduced by 64.3% in average compared to the baseline method in the six experiments. Moreover, the proposed method was shown to be effective even when the training data differed significantly from the actual data.

References

[1] G. Guo, "Human age estimation and sex classification," in *Studies in Computational Intelligence*, vol. 409. Berlin, Germany: Springer, 2012, pp. 101–131.

[2] I. Sobron, J. D. Ser, I. Eizmendi, and M. Velez, "Device-free people counting in IoT environments: New insights, results, and open challenges," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4396–4408, Dec. 2018.

[3] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J. S. Oh, "Semisupervised deep reinforcement learning in support of IoT and smart city services," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 624–635, Apr. 2018.

[4] B. Sliwa, N. Piatkowski, and C. Wietfeld, "The channel as a traffic sensor: Vehicle detection and classification based on radio fingerprinting," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7392–7406, Aug. 2020.

[5] J. Zinsstag *et al.*, "Transmission dynamics and economics of rabies control in dogs and humans in an African city," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 35, pp. 14996–15001, Sep. 2009.

[6] Y. Shiu *et al.*, "Deep neural networks for automated detection of marine mammal species," *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, 2020.

[7] D. A. Jiménez-Uribe, D. Daniels, A. González-Álvarez, and A. M. Vélez-Pereira, "Influence of vehicular traffic on environmental noise spectrum in the tourist route of Santa Marta City," *Energy Rep.*, vol. 6, pp. 818–824, Feb. 2019.

[8] N. R. Baek, S. W. Cho, J. H. Koo, N. Q. Truong, and K. R. Park, "Multimodal camera-based gender recognition using human-body image with two-step reconstruction network," *IEEE Access*, vol. 7, pp. 104025–104044, 2019.

[9] H. Iijima, T. Nagaike, and T. Honda, "Estimation of deer population dynamics using a Bayesian state-space model with multiple abundance indices," *J. Wildlife Manag.*, vol. 77, no. 5, pp. 1038–1047, 2013.

[10] T. Ikeda, H. Takahashi, T. Yoshida, H. Igota, and K. Kaji, "Evaluation of camera trap surveys for estimation of sika deer herd composition," *Mammal Study*, vol. 38, no. 1, pp. 29–33, 2013.

[11] M. S. Norouzzadeh *et al.*, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 25, pp. E5716–E5725, Jun. 2018.

[12] J. Wäldchen and P. Mäder, "Machine learning for image based species identification," *Methods Ecol. Evol.*, vol. 9, no. 11, pp. 2216–2225, 2018.

[13] M. Willi *et al.*, "Identifying animal species in camera trap images using deep learning and citizen science," *Methods Ecol. Evol.*, vol. 10, no. 1, pp. 80–91, 2019.

[14] L. C. Potter, C. J. Brady, and B. P. Murphy, "Accuracy of identifications of mammal species from camera trap images: A Northern Australian case study," *Aust. Ecol.*, vol. 44, no. 3, pp. 473–483, 2019.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, 2010, pp. 248–255.

[16] J. Xu, Y. Zhang, and D. Miao, "Three-way confusion matrix for classification: A measure driven view," *Inf. Sci.*, vol. 507, pp. 772–794, Jan. 2020.

[17] T. Kiyohara, R. Orihara, Y. Sei, Y. Tahara, and A. Ohsuga, "Activity recognition for dogs based on time-series data analysis," in *Proc. ICAART*, 2015, pp. 163–184.

[18] R. Tanno, A. Saeedi, S. Sankaranarayanan, D. C. Alexander, and N. Silberman, "Learning from noisy labels by regularized estimation of annotator confusion," in *Proc. IEEE CVPR*, 2019, pp. 11244–11253.

[19] I. D. Apostolopoulos and T. A. Mpesiana, "COVID-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," *Phys. Eng. Sci. Med.*, vol. 43, pp. 635–640, Apr. 2020.

[20] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[21] A. Toshev and G. C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *Proc. CVPR*, 2014, pp. 1653–1660.

[22] M. Mehdipour Ghazi, B. Yanikoglu, and E. Aptoula, "Plant identification using deep neural networks via optimization of transfer learning parameters," *Neurocomputing*, vol. 235, pp. 228–235, Apr. 2017.

[23] Y. Shen, H. Zhou, J. Li, F. Jian, and D. S. Jayas, "Detection of stored-grain insects using deep learning," *Comput. Electron. Agricult.*, vol. 145, pp. 319–325, Feb. 2018.

[24] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Inf. Fusion*, vol. 42, pp. 146–157, Jul. 2018.

[25] S. Visa, B. Ramsay, A. Ralescu, and E. Van Der Knaap, "Confusion matrix-based feature selection," in *Proc. CEUR Workshop*, vol. 710, 2011, pp. 120–127.

[26] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Inf. Sci.*, vols. 340–341, pp. 250–261, May 2016.

[27] M. Ohsaki, P. Wang, K. Matsuda, S. Katagiri, H. Watanabe, and A. Ralescu, "Confusion-matrix-based kernel logistic regression for imbalanced data classification," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 9, pp. 1806–1819, Sep. 2017.
[28] R. V. Krejcie and D. W. Morgan, "Determining sample size for research activities," *Educ. Psychol. Meas.*, vol. 30, no. 3, pp. 607–610, 1970.
[29] G. D. Israel, "Determining sample size," Dept. Agricultural Educ. Commun., Extension Specialist, Program Evaluation Org. Develop., Inst. Food Agricultural Sci. (IFAS), Univ. Florida, Gainesville, FL, USA, Rep. PEOD-6, 1992.
[30] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proc. ACM SIGMOD*, 2000, pp. 439–450.
[31] R. Agrawal, R. Srikant, and D. Thomas, "Privacy preserving OLAP," in *Proc. ACM SIGMOD*, 2005, pp. 251–262.
[32] Y. Sei and A. Ohsuga, "Differential private data collection and analysis based on randomized multiple dummies for untrusted mobile crowdsensing," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 926–939, Apr. 2017.
[33] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.
[34] R. Rothe, R. Timofte, and L. Van Gool, "DEX: Deep expectation of apparent age from a single image," in *Proc. ICCV Workshops*, 2015, pp. 10–15.
[35] E. D. Cubuk, B. Zoph, V. Vasudevan, and Q. V. Le Google Brain, "AutoAugment: Learning augmentation strategies from data," in *Proc. CVPR*, 2019, pp. 113–123.
[36] B. Huang, Z. Xu, B. Jia, and G. Mao, "An online radio map update scheme for WiFi fingerprint-based localization," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6909–6918, Aug. 2019.
[37] M. Xue *et al.*, "Locate the mobile device by enhancing the WiFi-based indoor localization model," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8792–8803, Oct. 2019.

Yuichi Sei (Member, IEEE) received the Ph.D. degree in information science and technology from the University of Tokyo, Tokyo, Japan, in 2009.

From 2009 to 2012, he was with Mitsubishi Research Institute, Chiyoda, Japan. He joined the University of Electro-Communications, Tokyo, in 2013, where he is currently an Associate Professor with the Graduate School of Informatics and Engineering. He is also a Visiting Researcher with Mitsubishi Research Institute, Chiyoda, Japan, and an Adjunct Researcher with Waseda University, Tokyo. His current research interests include pervasive computing, privacy-preserving data mining, and software engineering.

Akihiko Ohsuga (Member, IEEE) received the Ph.D. degree in computer science from Waseda University, Tokyo, Japan, in 1995.

From 1981 to 2007, he was with Toshiba Corporation, Minato, Japan. He joined the University of Electro-Communications, Tokyo, in 2007. He is currently a Professor with the Graduate School of Informatics and Engineering, where he is also the Dean of the Graduate School of Information Systems. He is also a Visiting Professor with the National Institute of Informatics, Tokyo. His research interests include agent technologies, Web intelligence, and software engineering.

Prof. Ohsuga received the IPSJ Best Paper Awards in 1987 and 2017. He served as the Chair of IEEE CS Japan Chapter, and a member of JSAI Board of Directors and JSSST Board of Directors. He is a member of the IEEE Computer Society, the Information Processing Society of Japan, the Institute of Electronics, Information and Communication Engineers, the Japanese Society for Artificial Intelligence, the Japan Society for Software Science and Technology, and the Institute of Electrical Engineers of Japan. He has been a Fellow of IPSJ since 2017.