SN Computer Science (2022) 3:178
https://doi.org/10.1007/s42979-022-01076-2

ORIGINAL RESEARCH

A Collaborative Training Using Crowdsourcing and Neural Networks on Small and Difficult Image Classification Datasets

Tomoumi Takase[1]

Received: 30 September 2021 / Accepted: 18 February 2022 / Published online: 2 March 2022

Abstract

Although machine learning has recently achieved performance that exceeds human capacity in prediction, humans still have an advantage in difficult tasks when the number of training samples is small or when human knowledge is required to identify features that are included in samples. However, if specialized knowledge is required, the number of humans that can perform those tasks is limited. In this study, we effectively use crowdsourcing to incorporate domain knowledge in neural network trainings; specifically, we decide feature values by asking crowdsourcing workers to answer easy questions prepared based on dictionary. We evaluated this method on a single type of task that is intuitive and relevant for non-specialists, which is binary classification of dog image datasets with similar breeds, and found that using crowdsourcing tended to improve the performance of machine-learning models.

Keywords Crowdsourcing · Neural network · Machine learning · Image classification · Feature extraction

Introduction

Machine-learning methods have been used for prediction tasks in many practical fields. Specifically, deep learning has been positively used because it has powerful approximating performance on the end-to-end process. Many types of deep neural networks have been developed and have shown outstanding performance for practical data [7, 14, 15, 34, 37, 38]. Surprisingly, deep learning presented the prediction performance that even exceeds human capacity [3, 12].

A large amount of data cannot be often obtained because of financial or availability difficulty. Using deep learning for a small amount of data tends to be avoided because it can cause overfitting whereas a study demonstrated that deep learning presented a high generalization performance for small datasets [9]. However, neural networks may be unsuitable if small datasets consist of difficult samples that require domain knowledge to be predicted. Since humans still have an advantage in those situations, a manual classification by an expert with domain knowledge should be more suitable than deep learning.

A problem is that we cannot always obtain the help of experts that are suitable for target fields. Therefore, we need to leverage the capabilities of non-experts. A powerful approach for achieving it is to use crowdsourcing, which is a service that can request work to unspecified persons through the Internet. Labeling and segmentation are representative examples of crowdsourcing tasks, and most such tasks address simple examples that non-experts can answer; however, our study addresses difficult classification tasks that require expert knowledge. We enable non-experts to answer difficult tasks by deciding feature values for each sample through crowdsourcing on the basis of feature texts selected from dictionaries beforehand. After that, the feature values obtained by crowdsourcing are used to train and test multilayer perceptron (MLP). This hybrid method of crowdsourcing and neural networks can be easily applied to multiclass classifications and to other types of data such as text and voice. In this study, the methodology has only been validated in a single type of task, and investigating the effectiveness of the method on less intuitive tasks is next step as this task is one for which the non-specialist status of the crowdsourced workers may not be as relevant as in more specific domains.

✉ Tomoumi Takase
takase-tomoumi@aist.go.jp

[1] Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

Deep learning has high approximating performance on the end-to-end process and is often used without manual preprocessing for image datasets. Although raw data can be also directly input to neural networks in our study, we employ inputting features extracted from images. Comparing using crowdsourcing for preprocessing with using only deep learning is an original attempt and helps to produce a novel method for classification tasks.

Although several studies have focused on image classification using crowdsourcing, most studies for crowdsourcing do not compare the results using crowdsourcing with those using deep learning. Automatic training and high performance that deep learning has cannot be ignored; thus, we compare the performance of the proposed method with that of deep learning. We evaluated our method on the binary classification of dog image datasets with confusing breeds and compared the prediction accuracy with several machine-learning models including ResNet [13], which is a typical model for deep learning.

A main purpose of this study is to demonstrate the effectiveness of incorporating non-expert's work into neural network training as well as proposing an effective training algorithm. Our contributions are summarized as follows:

(1) We designed a training method that combines human work with neural networks for difficult classification tasks.

(2) We proposed an effective way to use crowdsourcing for non-expert workers on small and difficult classification datasets.

(3) We experimentally showed that the proposed method can produce higher accuracy in most cases compared to the cases using several machine-learning methods, including deep learning and transfer learning.

Related Works

Crowdsourcing has been used in studies on various fields [4, 33, 41]. Image analysis is a representative example of crowdsourcing tasks, specifically, crowdsourcing is often used for the analysis of medical images [6, 23]. Crowdsourcing is also effectively used to classify samples with difficult classes. For example, Duan and Tajima [8] focused on the classification tasks for dogs and wild species. They reorganized a flat classification task into a hierarchical task and allocating workers to appropriate subtasks that are based on each worker's ability. In general, whether workers have domain knowledge that is required to engaged in a task affects the quality of crowdsourcing results. To reduce the effect of the difference in worker's ability, Tao et al. [35] proposed to decide the weights for weighted majority voting by modeling the domain knowledge of different workers in crowdsourcing. Zhang et al. [46] proposed a task assign algorithm to assign crowd assessment tasks about security and privacy in online social networks to most appropriate workers efficiently, effectively, and accurately. Saralioglu and Gungor [30] employed labeling using crowdsourcing to solve insufficient training data problem in deep learning-based classification.

Appropriately using manual work in crowdsourcing and automatic processes in machine learning is significant for improving the prediction accuracy. A machine-learning system that requires human interaction has recently attracted attention as a type of human-in-the-loop system [44]. A representative method is active learning [31, 32], which is a learning method that aims to improve the training performance by manually labeling data without labels on the basis of the prediction results by machine learning. A study developed a method to weight features on the basis of the level of confidence during active learning tasks [21]. Other studies applied active learning to deep learning [10, 37, 38]. They are similar to our crowdsourcing-based method but require manual labeling by experts with domain knowledge, whereas our method does not require it.

However, several studies have applied the approach of combining crowdsourcing with machine learning. For example, Albarqouni et al. [1] presented a new concept for learning from crowdsourcing that directly handles data aggregation as part of the learning process of the CNN via an additional crowdsourcing layer to deal with noisy annotations from crowdsourcing. Lu et al. [22] presented a novel approach using crowdsourced label distributions to improve the generalization performance of CNNs for facial expression recognition. A differential privacy-enabled DNN learning framework, which was developed by Wang et al. [40], protects the data privacy provided by crowdsourcing workers by intentionally injecting noise to the affine transformation of the input data features. Although our method combines crowdsourcing and neural networks, it focuses on image classification with similar classes.

Feature-Based Approach for Crowdsourcing

Overview

As described in "Introduction", neural networks exhibit high performance in prediction tasks, but manual prediction by humans still has an advantage in tasks that requires knowledge. Although crowdsourcing is a convenient tool, most workers do not have professional knowledge for the target tasks. Therefore, an overall scheme should be designed so that the capability of such workers can be effectively used.

In this study, we propose an effective way to use crowdsourcing to appropriately classify a small number of samples with similar classes. As seen in the flow of the proposed method shown in Fig. 1, this method consists of two steps:

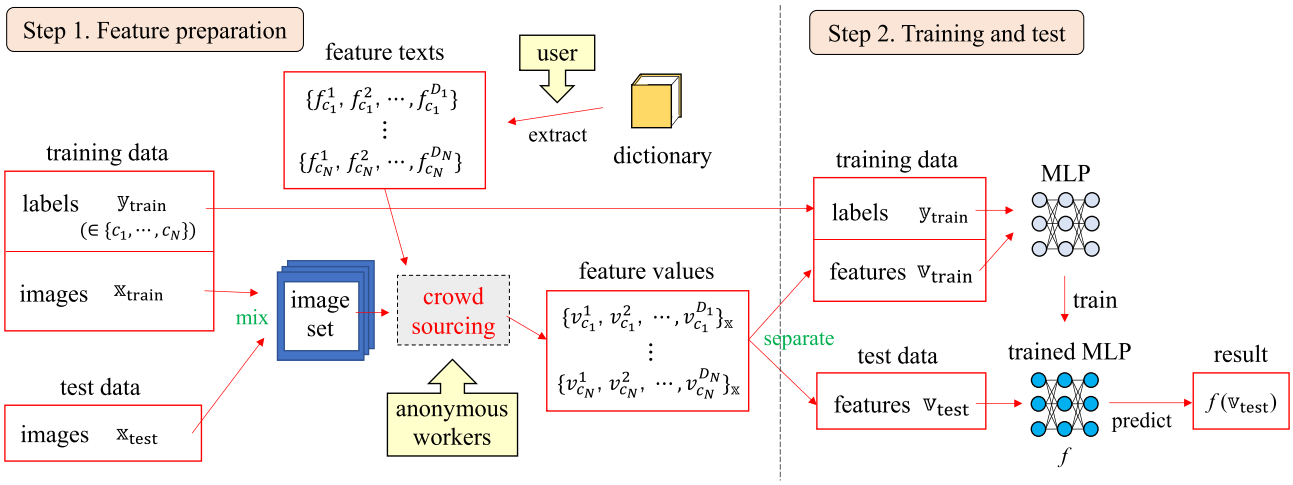

Figure 1 Flow of the proposed method. A human prepares the features of samples using dictionary and crowdsourcing in step 1, and a neural network is trained and predicts the class on the basis of the features in step 2

"feature preparation" for step 1 and "training and test" for step 2; step 1 is performed by humans, and step 2 is performed by a neural network. In step 1, the features for samples are selected from dictionaries, and feature values are decided through crowdsourcing. In step 2, a neural network is trained using the features, and the class is predicted. Being expressed using symbols, the purpose of the task that we address is to predict the class of the test data $\mathbb{x}_{\text{test}}$, given the training data ($\mathbb{x}_{\text{train}}$, $\mathbb{y}_{\text{train}}$). We assume supervised learning, i.e., when all training data have labels $\mathbb{y}_{\text{train}}$ ($\in \{c_1, c_2, \cdots, c_N\}$).

Generating Feature Texts

In step 1, a user, i.e., one who uses the proposed method, needs to create features for preparation. Since crowdsourcing workers do not usually have domain knowledge for target tasks (i.e., do not know objects that are contained in images), the user does not directly ask the workers to classify images but provide pre-selected features. The user selects features for the target classes from dictionaries or the database on the Internet (we refer to them as "dictionaries" for simplicity). A set of feature texts is $\{f_{C_n}^1, f_{C_n}^2, \cdots, f_{C_n}^{D_n}\}$, where D_n is the number of feature texts for class C_n. Each element is represented as a short sentence (e.g., "The eyes are blue.") that describes a feature of the class. Each sentence must contain only one feature so that workers can determine the strength of the feature for each sentence. It is desirable to select features that belong to one class but do not belong to other classes, especially when the number of classes is small. The generation of feature texts is performed for all classes $\{c_1, c_2, \cdots, c_N\}$, and the value of D_n can be different between classes. If many features that identify C_n can be obtained, the user should use many feature texts to improve the prediction accuracy. However, it leads to an increase in crowdsourcing costs because the number of questions that workers must answer increases. Thus, the user must decide the value considering the crowdsourcing costs.

Feature Weighting Using Crowdsourcing

Next, the values of the selected features must be chosen for each sample according to the strength of the features. To efficiently choose the values, we ask anonymous workers on crowdsourcing to weight feature values. All training and test images are randomly mixed and shown to the workers with a set of the feature texts represented by $\{f_{C_n}^1, f_{C_n}^2, \cdots, f_{C_n}^{D_n}\}$ ($n \in \{1, 2, \cdots, N\}$). The labels of samples are not given to workers. The workers rate on a scale of 0–10 how strong each feature is in each sample, and it is represented by $\{v_{C_n}^1, v_{C_n}^2, \cdots, v_{C_n}^{D_n}\}_{x_i}$, which denotes the weights added to feature texts $\{f_{C_n}^1, f_{C_n}^2, \cdots, f_{C_n}^{D_n}\}$ for the class C_n in sample x_i($\in \mathbb{x}$). When workers answer feature values of a sample, they tend to rate higher for a feature and lower for another feature because the set of feature texts contains the features of all classes.

To enhance the reliability of results, asking the same samples to several workers is typical in crowdsourcing. The mean of the values obtained from all workers that are asked to answer the same sample is calculated for each feature in each sample and is used for the next step. There are methods that integrate in an effective way the results obtained from several workers. A powerful approach is to adjust the results according to the worker's capability. For example, the results given by workers with low capability or spam workers, which can be judged from the accuracies in training data,

should not be included into the final result. This method is also effective because the mean can be calculated regarding the capabilities of the workers as weights [16]. However, the accuracy of training data could not be obtained in the current method because the method asks workers to answer feature values and not labels. Therefore, an improvement in designing the method is required to consider the capability of workers, and we simply average the given scores in our experiments.

Training and Testing Models

In step 2, a neural network is trained and tested using features prepared in step 1. To perform it, the set of feature values is separated into the training data $\mathbb{v}_{\text{train}}$ and the test data $\mathbb{v}_{\text{test}}$ on the basis of whether or not samples have labels. Then, the training data ($\mathbb{v}_{\text{train}}$, $\mathbb{y}_{\text{train}}$) are used for training a neural network. MLP should be used as a model because $\mathbb{v}_{\text{train}}$ are not the image forms but numerical data. The number of the input units of the MLP matches with the number of the feature texts represented by $\{f_{C_n}^1, f_{C_n}^2, \cdots, f_{C_n}^{D_n}\}$ ($n \in \{1, 2, \cdots, N\}$). After training the MLP, the test data $\mathbb{v}_{\text{test}}$ are input into it, and the class is predicted. The use of MLP with nonlinear transformation is expected to produce higher classification accuracy than the prediction using linear models.

Application to Multiclass Classification

Our crowdsourcing-based method can be applied to multiclass classification in two ways. First is to simply include samples from all classes in one crowdsourcing task. This enables that multiclass classification is dealt with in the same way as described in "Generating Feature Texts", "Feature Weighting Using Crowdsourcing" and "Training and Testing Models". Specifically, the example above was binary classification when n is 2, but it represents multiclass classification if n is larger than 2. A set of feature texts tends to increase when n is large, that is, the value of D_n for each class. Thus, the total number of features should be small to reduce the burden of human labeling and costs for such labeling. The use of a small number of feature texts is not a problem when a class often has only one or two strong unique features even if it is one of confusing classes. Therefore, this approach is useful if the number of classes is small.

However, this is not an appropriate approach if a task has a large number of classes. A solution to the problem in this case is to partially use the proposed method. CNN should be used for classes that are easy to classify, and this method should be used for classes that are difficult to classify. Specifically, in a classification task with N classes ($c_1, c_2, \cdots, c_N$), if c_1 and c_2 is a pair of confusing classes, mix c_1 with c_2 and generate a new class c_{1_2}. First, train and test a CNN with $N-1$ classes including c_{1_2}, and then train and test an MLP using features for c_1 and c_2 obtained by crowdsourcing to classify samples with their classes. This approach makes feature preparation by crowdsourcing simple and easy, and as a result, reduces cost. To investigate whether these ways are actually effective is future work, the experiments in this study focused on assessing the proposed method on binary classification that is the base of those applications.

Experiments

Dataset

Our experiments target the binary classification for dog breeds. Dog breed classification is a general task; for example, the ImageNet dataset [28], which is often used in competitions, includes several dog classes. The experiments using datasets with originally corrected dog image samples were also conducted in the study by Duan and Tajima [8], however, our results cannot be compared with their results because the purpose and condition of the experiments are completely different.

Although there are many types of dog breeds, we chose two pairs of dog breeds as confusing breeds: Alaskan Malamute and Siberian Husky, and Boston Terrier and French Bulldog. The appearances of these breeds are considerably similar as seen in examples shown in Fig. 2 and are difficult to classify for people who are not familiar with dog breeds. Moreover, the presence of several features that identify these breeds is also the reason why we chose these dog breeds because a large number of feature text is desirable to evaluate our proposed method, which has a novelty in feature preparation.

The automatic correction of images by web scrolling is convenient and often used for preparing the original dataset; however, it tends to correct samples with wrong labels in our experiments that address confusing breeds. Thus, we extracted the images for those breeds from the Stanford Dogs dataset [17]. We created two datasets for different combinations that were described above, and identified them by naming the combination of Alaskan Malamute and Siberian Husky as "Dog 1 dataset" and that of Boston Terrier and French Bulldog as "Dog 2 dataset." Each dataset consists of 240 samples, and each class was divided into 20 training samples and 100 test samples. The quality of images used in the experiments was high enough. The images containing two or more dogs were excluded because which dog is referred to by the feature text is unclear. We only used two datasets to increase the reliability of our proposed method, and one dataset is irrelevant to the others.

Figure 2 Examples of similar dog breed data used for the experiments

(a) Dog 1 dataset: Alaskan Malamute

(b) Dog 1 dataset: Siberian Husky

(c) Dog 2 dataset: Boston Terrier

(d) Dog 2 dataset: French Bulldog

We verified that the datasets we created are relatively difficult to classify. We selected six breeds in an ascending order of WordNet ID from the Stanford Dogs dataset [17] and prepared ten breeds in addition to the four breeds included in Dog 1 and Dog 2 datasets. Training and test data were prepared by dividing all samples into target classes. Trainings with ResNet18 for ten classes classification was performed, and the training condition are described in "Machine-Learning Models". The multiclass confusion matrix for dog breed datasets is shown in Fig. 3. Each row of the matrix represents instances in an actual class while each column represents the instances in a predicted class. In these classes, 6 and 7 belong to the Dog 1 dataset and 8 and 9 belong to the Dog 2 dataset. For Dog 1, many Malamute samples were mistakenly recognized as Siberian Husky and vice versa. For Dog 2, many French Bulldog samples were mistakenly recognized as Boston Bulldog. Considering these results, we can conclude that Dog 1 and Dog 2 datasets are difficult datasets with confusing breeds.

		0	1	2	3	4	5	6	7	8	9
	0: Chihuahua (79)	17	1	2	13	18	6	7	8	4	3
	1: Japanese_spaniel (93)	4	54	5	0	9	7	4	7	3	0
	2: Maltese_dog (131)	3	1	90	5	13	5	2	11	1	0
	3: Pekinese (76)	5	0	9	20	21	4	9	6	1	1
	4: Shih-Tzu (103)	5	4	12	12	33	2	10	20	4	1
	5: Blenheim_spaniel (95)	5	8	1	1	9	52	4	14	0	1
Dog 1	6: Malamute (90)	3	0	9	4	19	2	8	42	1	2
Dog 1	7: Siberian_husky (96)	4	0	6	5	6	6	21	40	7	1
Dog 2	8: Boston_bull (77)	4	2	5	3	12	3	3	16	25	4
Dog 2	9: French_bulldog (86)	2	3	9	7	16	3	9	13	21	3

Figure 3 Multiclass confusion matrix for dog breed datasets. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class. Values in parentheses in the leftmost column denote the number of test samples for each class. "Boston bull" is the same as Boston terrier in the Dog 2 dataset that we used to evaluate our proposed method

Method and Results of Feature Preparation

We chose five features for the Dog 1 dataset and six features for the Dog 2 dataset through the dog breed database on several websites. The feature texts that refer to those features are listed in Table 1. Each feature necessarily represents one of two classes, and the class names in column "Breed" denote which of two classes has the feature. Some features that are difficult to be judged from photos were not selected, such as the dog size and weight, which can be rarely obtained from images, and the eye color, which is difficult to distinguish because the eye is so small in most images.

Table 1 Features used for crowdsourcing tasks and feature values for each class that workers actually provided

Dataset	Text based on pre-selected features	Breed	Result (a)	Result (b)
Dog 1	1. The ears are set wide apart	(a) Malamute	5.53	4.59
	2. The ears point straight up	(b) Husky	6.33	6.76
	3. The muzzle is bulky	(a) Malamute	4.92	4.52
	4. The tail is carried over its back, not hanging down	(a) Malamute	3.54	3.11
	5. The coat is thick and wooly, not short and flat	(a) Malamute	5.30	4.72
Dog 2	1. The build is muscular	(b) Bulldog	3.98	4.58
	2. The legs are long	(a) Terrier	3.43	2.73
	3. The head is round, not square	(a) Terrier	4.70	4.82
	4. The ears stand erect	(b) Bulldog	6.22	7.73
	5. The ears are round, not pointed	(b) Bulldog	3.53	5.39
	6. The color is black, seal, or brindle, and the dog has white markings	(a) Terrier	7.32	4.39

Each feature necessarily represents one of two classes. The workers are required to answer using the score of 0–10 how strong each feature appears in each given image. Result (a) is the mean of the scores given to Malamute samples in the Dog 1 dataset and that to Terrier samples in the Dog 2 dataset; result (b) is the mean of the scores given to Husky samples in the Dog 1 dataset and that to Bulldog samples in the Dog 2 dataset. Each result is the mean of the scores given by ten workers

Figure 4 Example of tasks provided to workers through crowdsourcing. Each worker was assigned 40 images. Workers decide from the image the strength (0–10) of features described by texts

We used Amazon Mechanical Turk [2] (Amazon) as a crowdsourcing platform. For each dataset, we divided all the samples into six tasks, and each task contained 40 samples with 20 samples for each class. Ten workers were assigned the same task, and we averaged the feature values provided by them. Images and texts were shown to workers using the HTML format, as exemplified by Fig. 4. The workers were not taught any information other than images and feature texts, such as which class each sample belongs to and whether each sample is the training data or not. The workers were required to provide a score of 0–10 using the slide to answer how strongly each feature appears in each given image. 10 denotes "strongly agree," 5 denotes "neither agree nor disagree," and 0 means "strongly disagree." Moreover, when a feature that is referred to by the text is not clearly seen in the image, the workers provided a value close to 5, because such a feature is not significant to classify those images even if it is generally significant to distinguish those breeds.

The scores for each feature texts obtained by crowdsourcing are shown in Table 1. These were calculated on the basis of the scores for the training data. "Result (a)" is the mean of the scores given to Malamute samples in the Dog 1 dataset and that to Terrier samples in the Dog 2 dataset; "Result (b)" is the mean of the scores given to Husky samples in the Dog 1 dataset and that to Bulldog samples in the Dog 2 dataset. A value that exceeds 5 means that the breed strongly includes the feature, but the relative evaluation between two breeds in the same dataset is significant for neural network training. By comparing the results with the class names shown in the "Breed" column, we can determine that the scores provided by workers in crowdsourcing were mostly reasonable because the breed with the higher score of "Result (a)" and "Result (b)" corresponds to that in the column "Breed" in all feature texts other than "3. The head is round, not square." in Dog 2 dataset, where workers provided a higher score for Bulldog although the feature text refers to Terrier.

Moreover, a feature that has a large difference between the values for two breeds means that it is a significant feature for predicting the breed. For example, "1. The ears are set wide apart." in the Dog 1 dataset, which was selected as a feature of Malamute, is considered to be a significant feature because the score for (a) is higher and the difference between

the scores for (a) and (b) is considerably large. Similarly, "6. The color is black, seal, or brindle, and the dog has white markings." in the Dog 2 dataset is considered to be a significant feature for identifying Terrier.

Machine-Learning Models

In the proposed method, we trained an MLP containing one hidden layer with 50 units using the feature values obtained by the crowdsourcing tasks. In the MLP, a rectified linear unit [11] was inserted after each layer except for the final layer, which used the softmax function. Training was performed for 1000 epochs. The training can be finished in a moment because the input is only 5 or 6 dimensions and the model is so small.

We compared the proposed method with several machine-learning models. In those models, pixel values of images were directly input to the models and the class was predicted after training the model. For an MLP, we resized the samples to 64×64 and then deformed them to samples with 4096 dimensions. Using an MLP with 1000 hidden layers, training was performed for 1000 epochs. For a CNN, we mainly used ResNet18 [13], which is widely used. Although this model does not contain many parameters compared to other deep CNNs, it is suitable for our small datasets because too large model can easily cause overfitting, which deteriorates the generalization performance. We also used LeNet [20], AlexNet [19], and ResNet34 [13]. Since the input was high dimension and the model has several layers, the training takes longer than training an MLP, so the training was performed for 200 epochs. We used samples resized to 256×256.

Neural network trainings include several hyperparameters and standard implementation values were used in our experiments. As a common setting for MLP and CNN, we used Adam optimization [18] with an initial learning rate of 0.001. The batch size was always fixed to 10 because the number of training samples is small, and we confirmed that the training was sufficiently performed in both models. The test accuracy using the test data was calculated at each epoch, and the highest test accuracy achieved during training was evaluated. Six trials were performed using randomly divided training and test samples and different initial weights that were sampled from normal distribution, and the mean and standard error were determined.

Data augmentation is an approach that is generally considered to be effective when the amount of data is small; this is a method for increasing the number of examples by manipulating raw data. Data augmentation is widely used in applied machine learning [27, 29, 39], and its effect has been investigated [5, 25]. Data augmentation can be easily applied to image data, so we applied it to only the conventional method with CNN. We applied normalization and two augmentation methods of horizontal flip and random rotation to all training and test samples, each method generated random hyper-parameter values for each epoch. Moreover, we used the mix-up augmentation technique [45], which generated a new example by linearly interpolating the inputs and labels of two examples in the input space. Currently, this approach is attracting attention, and several studies have investigated the effectiveness of mix-up and improved the algorithm [36, 42]. Since the application of mix-up is not restricted to image data, we applied it to both CNN and MLP, but only to training data.

Moreover, we also used ResNet18 with transfer learning [24, 26], which is a powerful training method that applies a model trained using a dataset to a training using another dataset. Transfer learning is useful, especially when a small number of samples is used. We used parameters trained using the ImageNet dataset [28], and we trained again only the parameters that are included in the fully-connected layer.

As other machine-learning methods, we used a support vector machine (SVM) and k-nearest neighbor algorithm (k-NN). Both of them are supervised learning models and can be easily applied to our datasets. SVM classifies samples so that there is margin between samples for different classes and can efficiently perform a nonlinear classification using the kernel trick. We predicted classes using linear, RBF, and polynomial kernels with the default hyper-parameter values implemented in Scikit-learn, and evaluated the highest test accuracy among those results. k-NN classifies samples by a plurality vote of its neighbors, with the sample being assigned to the class most common among its k-nearest neighbors. We used $1, 2, \cdots, 10$ for k and evaluated the highest test accuracy among those results. In addition to using SVMs and KNNs with images as inputs, we used those with feature values obtained by crowdsourcing as inputs instead of training MLP.

Prediction Results

First, we investigated the test accuracy for several methods. As seen in the results shown in Table 2, the test accuracy when the proposed method was used was considerably higher than when only an MLP and CNNs were used, in particular for the Dog 2 dataset. Although ResNet34 has a deeper structure, it did not present a significantly higher accuracy than ResNet18 because a large number of parameters is not required for a small dataset. The accuracy for ResNet18 with data augmentation was almost the same as that without data augmentation in most cases, although data augmentation is generally effective for a small amount of data. However, it considerably improved the accuracy of the Dog 2 dataset when using the proposed method. Both k-NN and SVM produced lower accuracy than

Table 2 Comparison of test accuracies among the proposed and conventional methods

Training data	Dog 1	Dog 2
MLP	55.2 (±0.6)	59.6 (±0.9)
ResNet18 [13]	56.0 (±1.0)	62.5 (±1.4)
ResNet34 [13]	56.2 (±0.4)	61.8 (±1.4)
LeNet [20]	55.1 (±1.0)	60.6 (±1.2)
AlexNet [19]	54.2 (±1.9)	59.8 (±1.7)
ResNet18 + augmentation (flip, rotate)	56.2 (±0.6)	62.7 (±1.2)
ResNet18 + augmentation (mix-up [45])	56.8 (±0.8)	60.7 (±0.6)
ResNet18 + augmentation (flip, rotate, mix-up)	57.3 (±0.4)	59.4 (±0.9)
ResNet18 + transfer learning	**71.5** (±1.7)	84.7 (±1.5)
k-NN $K = 8$ (Dog 1), $K = 8$ (Dog 2)	49.3 (±0.8)	52.8 (±2.4)
SVM polynomial (Dog 1), linear (Dog 2)	51.9 (±2.0)	55.6 (±1.3)
Crowd + MLP	65.3 (±1.8)	90.8 (±0.5)
Crowd + MLP + augmentation (mix-up)	65.6 (±1.4)	**92.1** (±0.5)
Crowd + k-NN $K = 3$ (Dog 1), $K = 5$ (Dog 2)	60.1 (±1.9)	87.1 (±0.8)
Crowd + SVM polynomial (Dog 1), linear (Dog 2)	62.0 (±1.4)	88.0 (±1.1)

The highest accuracies achieved during training were compared. The values in parentheses are the standard error. Numbers in bold denote the highest accuracy

neural networks, regardless of whether using the conventional method or the proposed method.

The proposed method exhibited lower performance than that of ResNet18 with transfer learning in the Dog 1 dataset but exhibited higher performance in the Dog 2 dataset. Although transfer learning showed high performance in our experiments, it cannot perform well if source domain (ImageNet in this case) and target domain (Dog breed in this case) do not have common features. ImageNet was appropriate for transfer learning in our experiments because it includes dog breed classes. This was discussed in a study by Zamir et al. [43], which investigated the effectiveness of applying transfer learning from a task to another task. Thus, transfer learning does not always perform well, thus, the proposed method can be used in more situations.

Next, we investigated the effect of the number of training and test samples. In the abovementioned experiments, we fixed the number of training samples to 40 and that of test samples to 120, but conducting the experiments using various sample sizes strengthens the effectiveness of the proposed method. Thus, we compared the performance of ResNet18 and the proposed method when the number of training samples is 80, 120, 160, and 200, and the rest are used for test samples. As seen in Table 3, the proposed method produced higher test accuracy in all cases in both dog datasets.

Table 3 Effect of the number of training and test samples

Dataset	Training data	Test data	ResNet18	Crowd-based MLP
Dog 1	40	200	56.0 (±1.0)	**65.3** (±1.8)
	80	160	58.6 (± 1.2)	**67.9** (±1.2)
	120	120	56.4 (±1.0)	**66.7** (±0.9)
	160	80	62.5 (±2.1)	**67.7** (±1.0)
	200	40	65.0 (±1.8)	**71.3** (±2.5)
Dog 2	40	200	62.5 (±1.4)	**90.8** (±0.5)
	80	160	67.5 (±1.1)	**92.0** (±0.8)
	120	120	70.0 (±1.1)	**92.6** (±0.8)
	160	80	72.3 (±1.6)	**93.8** (±0.7)
	200	40	80.0 (±2.0)	**93.8** (±1.9)

Values in parentheses are the standard error. Numbers in bold denote higher accuracy

Conclusion

In this study, we investigated the effect of using crowdsourcing with neural network trainings and proposed a collaborative method of humans and neural networks for image classification with confusing classes. Although creating features by humans for neural network training has been commonly conducted, we presented a novel idea of generating feature texts on the basis of dictionaries and weighted features by workers on the crowdsourcing. This method has high versatility because it does not require the domain knowledge.

In the experiments with the difficult dog breed classification datasets, the proposed method produced higher accuracy than several machine-learning models including CNNs. Moreover, our method performed better than CNN with several data augmentation techniques; furthermore, it performed better when it was combined with mix-up. Although transfer learning performed well on our datasets,

our method produced higher accuracy than transfer learning in one dataset.

Although we focused on dog classification in this study, we are also interested in investigating how the proposed method also works in image classification tasks for other categories such as handwritten characters and illustration, less intuitive classification tasks such as medical images, and other classification tasks such as video and audio. Moreover, we aim for developing a method that reduces the effect of spam workers and workers with low capability to enhance the quality of crowdsourcing. By achieving these, we hope to combine human work and machine learning properly depending on the kind of tasks.

Acknowledgements This study is based on results obtained from a project, JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO), Japan.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Albarqouni S, Baur C, Achilles F, Belagiannis V, Demirci S, Navab N. AggNet: deep learning from crowds for mitosis detection in breast cancer histology images. IEEE Trans Med Imaging. 2016;35(5):1313–21.
2. Amazon Mechanical Turk. https://www.mturk.com/. Accessed 1 Oct 2020.
3. Bejnordi BE, Veta M, van Diest PJ, van Ginneken B, Karssemeijer N, Litjens G, van der Laak JA, Hermsen M, Manson Q, Balkenhol M, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. J Am Med Assoc (JAMA). 2017;318(22):2199–210.
4. Chatzimilioudis G, Konstantinidis A, Laoudias C, Zeinalipour-Yazti D. Crowdsourcing with smartphones. IEEE Internet Comput. 2012;16(5):36–44.
5. Cubuk ED, Zoph B, Mane D, Vasudevan V, Le QV. AutoAugment: learning augmentation policies from data. arXiv preprint: arXiv: 1805.09501. 2018.
6. de Herrera AGS, Foncubierta-Rodríguez A, Markonis D, Schaer R, Müller H. Crowdsourcing for medical image classification. Swiss Med Inform. 2014;30:13.
7. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N. An image is worth 16x16 words: transformers for image recognition at scale. In: Proceedings of the international conference on learning representations (ICLR). 2021.
8. Duan X, Tajima K. Improving multiclass classification in crowdsourcing using hierarchical schemes. In: Proceedings of the world wide web conference (WWW). 2019. p. 13–17.
9. Feng S, Zhou H, Dong H. Using deep neural network with small dataset to predict material defects. Mater Des. 2019;162(15):300–10.
10. Gal Y, Islam R, Ghahramani Z. Deep bayesian active learning with image data. Proc Int Conf Mach Learn (ICML). 2017;70:1183–92.
11. Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. In: Proceedings of the international conference on artificial intelligence and statistics (AISTATS). 2011. p. 315–323.
12. He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: Proceedings of the IEEE international conference on computer vision (ICCV). 2015. p. 1026–1034.
13. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). 2016. p. 770–778.
14. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. 2017.
15. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). 2018. p. 7132–7141.
16. Kazai G, Kamps J, Koolen M, Milic-Frayling N. Crowdsourcing for book search evaluation: impact of hit design on comparative system ranking. In: Proceedings of the international ACM SIGIR conference on research and development in information retrieval (SIGIR). 2011. p. 205–214.
17. Khosla A, Jayadevaprakash N, Yao B, Fei-Fei L. Novel dataset for fine-grained image categorization. In: The first workshop on fine-grained visual categorization (FGVC). 2011.
18. Kingma DP, Ba J. Adam: a method for stochastic optimization. In: Proceedings of the international conference on learning representations (ICLR). 2015.
19. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems (NIPS). 2012. p. 1097–1105.
20. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–324.
21. Legg P, Smith J, Downing A. Visual analytics for collaborative human-machine confidence in human-centric active learning tasks. In: Human-centric computing and information sciences. vol. 9. 2019.
22. Lu P, Li B, Shama S, King I, Chan JH. Regularizing the loss layer of CNNs for facial expression recognition using crowdsourced labels. In: Proceedings of the Asia Pacific symposium on intelligent and evolutionary systems (IES). 2017.
23. Ørting S, Doyle A, Hilten van MHA et al. A survey of crowdsourcing in medical image analysis. arXiv preprint: arXiv:1902.09159. 2019.
24. Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng. 2010;22(10):1345–59.
25. Perez L, Wang J. The effectiveness of data augmentation in image classification using deep learning. arXiv preprint: arXiv:1712.04621. 2017.
26. Pratt LY. Discriminability-based transfer between neural networks. Adv Neural Inf Process Syst (NIPS). 1993;5:204–11.
27. Rogez G, Schmid C. MoCap-guided data augmentation for 3D pose estimation in the wild. In: Advances in neural information processing systems (NIPS). 2016.
28. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L. ImageNet large scale visual recognition challenge. Int J Comput Vis. 2015;115:211–52.
29. Salamon J, Bello JP. Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Process Lett. 2017;24(3):279–83.
30. Saralioglu E, Gungor O. Crowdsourcing-based application to solve the problem of insufficient training data in deep learning-based classification of satellite images. Geocarto Int. 2021; 1–20.

31. Sener O, Savarese S. Active learning for convolutional neural networks: a core-set approach. In: Proceedings of the international conference on learning representations (ICLR). 2018.
32. Settles B. Active learning literature survey. In: Computer science technical report 1648, University of Wisconsin-Madison. 2009.
33. Sharma M, Saha O, Sriraman A, Hebbalaguppe R, Vig L, Karande S. Crowdsourcing for chromosome segmentation and deep classification. In: The IEEE conference on computer vision and pattern recognition (CVPR) workshops. 2017. p. 34–41.
34. Tan M, Le QV. EfficientNet: RethinkingModel scaling for convolutional neural networks. In: Proceedings of the international conference on machine learning (ICML). 2019.
35. Tao D, Cheng J, Yu Z, Yue K, Wang L. Domain-weighted majority voting for crowdsourcing. IEEE Trans Neural Netw Learn Syst. 2018;30(1):163–74.
36. Verma V, Lamb A, Beckham C, Couville A, Mitli-agkas I, Bengio Y. Manifold mixup: better representations by interpolating hidden states. In: Proceedings of the international conference on machine learning (ICML). 2019.
37. Wang J, Yang Y, Mao J, Huang Z, Huang C, Xu W. CNN-RNN: a unified framework for multi-label image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). 2016. p. 2285–2294.
38. Wang K, Zhang D, Li Y, Zhang R, Lin L. Cost-effective active learning for deep image classification. IEEE Trans Circuits Syst Video Technol. 2016;27(12):2591–600.
39. Wang X, Pham H, Dai Z, Neubig G. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP). 2018.
40. Wang Y, Gu M, Ma J, Jin Q. DNN-DP: differential privacy enabled deep neural network learning framework for sensitive crowdsourcing data. IEEE Trans Comput Soc Syst. 2020;7(1):215–24.
41. Xu Z, Liu Y, Yen N, Mei L, Luo X, Wei X, Hu C. Crowdsourcing based description of urban emergency events using social media big data. IEEE Trans Cloud Comput. 2016;8(2):387–97.
42. Yun S, Han D, Oh SJ, Chun S, Choe J, Yoo Y. CutMix: regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE international conference on computer vision (ICCV). 2019. p. 6023–6032.
43. Zamir AR, Sax A, Shen W, Guibas L, Malik J, Savarese S. Taskonomy: disentangling task transfer learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). 2018. p. 3712–3722.
44. Zanzotto FM. Human-in-the-loop artificial intelligence. J Artif Intell Res. 2019;64:243–52.
45. Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. mixup: beyond empirical risk minimization. In: Proceedings of international conference on learning representations (ICLR). 2018.
46. Zhang Z, Jing J, Wang X, Choo K-KR, Gupta BB. A crowdsourcing method for online social networks security assessment based on human-centric computing. Human-centric Comput Inf Sci. 2020;10.