It’s DONE: Direct ONE-shot learning without training optimization

Kazufumi Hosoda, Keigo Nishida, Shigeto Seno, Tomohiro Mashita
Hideki Kashioka, Izumi Ohzawa

Abstract

Learning a new concept from one example is a superior function of human brain and it is drawing attention in the field of machine learning as one-shot learning task. In this paper, we propose the simplest method for this task, named Direct ONE-shot learning (DONE). DONE adds a new class to a pretrained deep neural network (DNN) classifier with neither training optimization nor other-classes modification. DONE is inspired by Hebbian theory and directly uses the neural activity input of the final dense layer obtained from a data that belongs to the new additional class as the connectivity weight (synaptic strength) with a newly-provided-output neuron for the new class. DONE requires just one inference for obtaining the output of the final dense layer and its procedure is simple, deterministic, not requiring parameter tuning and hyperparameters. The performance of DONE depends entirely on the pretrained DNN model used as a backbone model, and we confirmed that DONE with a well-trained backbone model performs a practical-level accuracy. DONE has some advantages including a DNN’s practical use that is difficult to spend high cost for a training, an evaluation of existing DNN models, and the understanding of the brain. DONE might be telling us one-shot learning is an easy task that can be achieved by a simple principle not only for humans but also for current well-trained DNN models.

1 Introduction

As is well known, Artificial Neural Networks are originally inspired by the biological neural network in the animal brain. Subsequently, Deep Neural Networks (DNNs) achieved great success in computer vision [9, 14, 19] and other machine learning fields. However, there are lots of tasks that are easy for humans but difficult for current DNNs. One-shot learning is considered as one of those kinds of tasks [3, 18, 21, 24, 16]. Humans can add a new class to his/her large knowledge from only one input image but it is difficult for DNNs unless another specific optimization is added. In practical uses, additional optimizations require extra user skills and calculation costs for tuning parameters and hyperparameters. Thus, for example, if an ImageNet model [6, 15] that learned 1000 classes can learn a new class “baby” from one image of a baby without any additional training optimization, it will be useful in actual machine learning uses including out-of-distribution (OOD) detection tasks [39, 20, 33].

*Center for Information and Neural Networks (CiNet), Advanced ICT Research Institute, National Institute of Information and Communications Technology (NICT), 1-4 Yamadaoka, Suita, Osaka 565-0871, Japan
†Laboratory for Computational Molecular Design, RIKEN Center for Biosystems Dynamics Research (BDR), 6-2-3 Furuedai, Suita, Osaka, 565-0874, Japan
‡Graduate School of Information Science and Technology, Osaka University
§Cybermedia Center, Osaka University
Figure 1: Scheme of DONE. The neural activity input of final dense layer (orange $x$ vector in original model) obtained from a new-class data (an image of a cat) is directly used as a new-class vector (orange $w_{\text{cat}}$) in the new weight matrix ($W$) to obtain the new output vector ($y$). See text for explanation.

The human brain does not necessarily have more complex processes than DNNs, and rather one of known facts of the human brain function is that a series of simple processes such as linear filtering followed by a nonlinearity can describe the function of lower visual cortex \cite{4}. Also, another known fact is that the neural response can be predicted by a DNN not only in the lower layer but also in the higher visual cortex \cite{11}. These knowledge of the brain inspired the idea that one-shot learning is possible without adding extra training to existing high-performance DNNs. Indeed, some studies \cite{20,35} show that the capabilities of DNN itself have the potential to enable OOD detection.

In this paper, we introduce a very simple method, Direct ONE-shot learning (DONE). As shown in Figure 1, DONE directly uses the input of the final dense layer ($x$ in Figure 1), obtained by one input image belonging to a new class, as the weight vector for the new additional class $w_{\text{new}}$ (a row vector of the weight matrix $W$). Then it is done. The idea of the conversion from neural activity input to the weights was inspired by Hebbian theory \cite{2}.

Specifically, Hebbian theory states that a synaptic connection is strengthened when both its presynaptic and postsynaptic neurons are active simultaneously. When a single image of a new class, e.g., a cat, is presented as a visual input, the activity pattern in $x$ will be different from cases where other classes of images (e.g., rats and dogs) are presented. Although in actual brains of animals, Hebbian learning may not take place instantly, it is achieved instantly in an artificial DNN model by simply using $x$ as the weight vector $w_{\text{cat}}$ for the newly added “cat” output neuron in the output vector ($y$ in Figure 1).

In addition to Hebbian theory, another insight was that, for a DNN model trained with sufficiently rich set of images, a reasonable representation of unknown images must exist somewhere in the multi-dimensional space spanned by vector $x$. The location for any new image may be obtained by a simple one-shot interpolation calculation in $x$. Thus, unlike other methods for one-shot learning, DONE does not use information of other classes, but indirectly uses the information of previous training (knowledge) through how $x$ (the final-dense-layer input) have been provided.

Our method’s basis and procedure are very simple but it achieved a good performance and accuracy. In an actual implementation, DONE achieved over 50% accuracy in a one-shot image classification task that adds one or eight new classes to a model pretrained for the ImageNet 1000 classes (ViT (Vision Transformer) \cite{33} or EfficientNet \cite{28}) as a backbone (note that the chance level is less than 0.1%). In a typical five-way one-shot classification task, DONE with ViT achieved over 80% accuracy.

The advantages of DONE are (i) brain-inspired simple basis and procedure, (ii) no optimization thus little calculation cost, and (iii) no parameters or hyperparameters thus easily reproducible for anyone. DONE is so simple that we thought it might have already been reported, but to the best of our knowledge it was not. Thus in any case, this report will be useful for widespread application even if there are similar reports, because a similar method has not been widely used yet despite its excellent performance. It will also help us understand DNNs, as the performance of DONE just depends on the DNNs. In addition, DONE may provide a useful insight when exploring the learning principles of the brain because DONE is inspired by the Hebbian theory, and at least, DONE might be simply telling us “one-shot learning as a brain function can be easily reproduced by current DNNs in an image classification task.”
2 Related work

Typical approaches for one- or few-shot classification are metric learning, data augmentation, and meta learning. Each of these approaches has its own advantages and purposes, and they are not contradictory and can be used in a mixed manner.

Metric learning uses a classification at a feature space as a metric space [8, 21, 29]. Roughly speaking, metric learning aim to decrease the distances between training data belonging to the same class and increase the distance between the data belonging to different classes. Metric learning such as using Siamese network [13] is useful for tasks that require one-shot learning, e.g., face recognition. A Data-augmentation approach generatively increases the number of training inputs [41, 18, 24]. This approach includes various types such as semi-supervised approaches and example generation using Generative Adversarial Networks [10]. Meta learning approaches train abilities of learning systems to learn [17, 22, 27, 31, 36, 40]. The purpose of meta learning is to aim to increase the learning efficiency itself, and this is a powerful approach for learning from a small amount of training data, typically one-shot learning task [38].

It is meaningless to compare these approaches with DONE, because DONE does not contain any optimization algorithm. DONE just adds a new class to a DNN model directly from the inference result, and of course DONE can add a class to a model that is trained by another method of those approaches. Therefore, in principle, there is no reason for DONE to outperform other methods by itself in accuracy. The performance of DONE is uniquely determined by the backbone DNN, and thus the performance is suitable as a reference baseline for other methods. DONE does not aim for the highest accuracy, but contributes to practical convenience and understanding of DNNs and brain, as well as works as the baseline method.

3 Methodology

3.1 Basis and procedure

In most of DNN models for classification tasks, the output vector \( y \) denotes the degree to which an input belongs to each class and is calculated from the output vector of the final dense layer \( x \), weight matrix \( W \), and bias vector \( b \). Here, for \( i \)-th class in \( N \) classes (\( i = 1, 2, \ldots, N \)), a scalar \( y_i \) is calculated from the corresponding weight vector \( w_i \) (\( i \)-th row vector of \( W \) matrix) and bias scalar \( b_i \) as following equation.

\[
y_i = x \cdot w_i + b_i = ||x|| \cdot ||w_i|| \cos \theta + b_i,
\]

where only the cosine similarity \( \cos \theta \) depends on both \( x \) and \( w_i \). The cosine similarity will have the maximum value 1 if \( x \) and \( w_i \) are directly proportional. Thus, if a certain \( x \) is directly used for \( w_i \), the cosine similarity for \( i \)-th class becomes large when another \( x \) with a similar value comes.

DONE uses this basis. As shown in Figure 1, we directly apply \( x \) obtained from a new-class input image to the weight vector for the new class \( x_j \) (\( j = N + 1, \ldots \)). To convert \( x \) to \( x_j \), we apply quantile normalization [1, 3] so that statistical properties of the elements of new vector \( w_j \) are the same as those of the elements of original \( W \) matrix. As for \( b_j \), we employed the median of \( b \).

Quantile normalization is an easy and standard technique in Bioinformatics, and here we briefly explain why we applied it. Statistical properties of the elements of \( x \) (neural activity) and \( W \) (synaptic strength) can be different (e.g., difference in histograms between Figure 4(b) and (c)). The conversion from \( x \) to \( w_j \), is a transformation of a vector to a different dimension, referring to the original \( W \) matrix (denoted as \( W_{ori} \)), so that statistical properties of the elements of \( w_j \) vector become similar to those of the elements of \( W_{ori} \) matrix. For example, we can apply linear transformation from \( x \) to \( w_j \), so that the mean and variance of the elements (i.e., 1st and 2nd central moments) of \( w_j \) vector is the same as those of \( W_{ori} \) matrix. However, it is not clear if such adjustment for only 1st and 2nd central moments is enough in this situation where the 3rd or higher central moments is different (Figure 4). One of the simplest solution for every situation is to make all the statistical properties of the elements of \( w_j \) and \( W_{ori} \) identical. At the conversion from \( x \) to \( w_j \), quantile normalization makes all quantiles become identical between \( w_j \) and \( W_{ori} \), using rank information of \( x \) and value information of \( W_{ori} \). Namely, all statistical properties of the elements of \( w_j \) vector are identical to those of the original \( W_{ori} \) matrix, i.e., those probability distributions are the same.
Figure 2: One-class addition by one-shot learning. (a) Representative images of the newly-added CIFAR-100 classes. Each image was chosen as a representative because the model that learned the image showed the highest, median, and lowest accuracy in each class in (b)-(i). (b) Accuracy of the one-class-added models obtained by one-shot learning with DONE.

3.2 Implementation and Dataset

As backbone models for our method, we employed Vision Transformers for images (ViT) [33] and EfficientNet [28] as two representative DNNs with different characteristics. We used “vit-keras” [43] for ViT and “EfficientNet Keras (and TensorFlow Keras)” [42] for EfficientNet. Also, for demonstrating evaluation of DNNs in DONE, we used ResNet-50 [19], MobileNetV2 [25], and VGG16 [14], using Tensorflow [12]. All models used in this study are pretrained with ImageNet.

We used CIFAR-100 and CIFAR-10 [7] for the additional class, using Tensorflow [12]. Also, for demonstrating evaluation of DNNs in DONE, we used CIFAR-FS [26] by Torchmeta [30]. We used ImageNet (ILSVRC2012) images [6, 15] for testing the performance of models. We used a coarse 10 categorization of the original 1000 classes (Figure 4(a)), using information of previous 67 categorization [32]. All images were resized to (224 × 224) by OpenCV [44] or Tensorflow [12].

4 Results and Discussion

4.1 One-class addition by one-shot learning

First, according to our motivation, we investigated the performance of DONE when a new class from one image was added to a DNN model pretrained with ImageNet (1000 classes). As new additional classes, we chose eight classes, “baby”, “woman”, “man”, “caterpillar”, “cloud”, “forest”, “maple_tree”, and “sunflower” from CIFAR-100, which were not in ImageNet (shown in Figure 2(a)).

The weight parameters for the additional one class $w_j$ is generated from one image, thus the model had 1001 classes (not 1008 classes; see next subsection for the case of 1008 classes). To conduct stochastic evaluations, 100 different models for each class are built by using 100 different training images for one additional class.

Figure 2(b) shows Letter-value plots of the accuracy for each additional class. This result shows that the accuracy of additional classes are greater than the chance level ($= 1/1001$) in most of the models. The median accuracy of each class (black circles) were not much different from the accuracy of ImageNet validation (orange line), which suggests that the accuracy of the one-shot learning by DONE achieves a practical level. The mean of the median accuracy of 8 classes were 56.6% and 92.1% for ViT and EfficientNet, respectively (black line). Those values were not significantly different from the mean of the median accuracy of the all 100 classes of CIFAR-100, 56.5% and
Figure 3: Multi-class addition and K-shot learning. (a), (b), and (c) show the results of the 1008-class model constructed by 1, 10, and 100-shot learning, respectively. (d) shows the results of the 1100-class model constructed by 1-shot learning. Each graph shows the results about one model. The horizontal and vertical axes show the class of the input images, and the output class, respectively. The color and size of the circles indicate the percentage of the output class in the input-class images. The class numbers in (a), (b), and (c) are those shown in Figure 2(a). The class ID of CIFAR-100 was used in (d). The class [im] contains 1000 classes of ImageNet.

86.5% for ViT and EfficientNet, respectively (in Welch’s t-test with α=0.05), although ImageNet 1000 classification contains classes that are similar to the rest of 92 classes in CIFAR-100.

An obvious fact in one-shot learning is that a bad image for one-shot training produces a bad performance, and vice versa. In this evaluation some models show low performances due to a bad training image, e.g., the accuracy was 6% in ViT when the training image was a baby image shown at the bottom left in Figure 2(a). But in practical usage of one-shot learning, a user supposed to use a representative image for the training. We therefore think that the low performances due to the training by a bad image is not a significant issue.

We investigated the interference from the additional class to the classification performance of the original 1000 classes. We evaluated the original 1000-class model and eight 1001-class models that showed the median accuracy, by using all 50,000 ImageNet validation images (Figure 2(b)). The accuracy of the 1000 classes of ImageNet Validation of the added models (orange cross) was not much different from the original model (orange line), as all accuracy was 65% and 69% in ViT and EfficientNet, respectively. However, actually there is small interference, and Figure 2(b) also shows the numbers of images in which the original model and added model gave different answers in the 50,000 images (orange numbers), regardless of whether it is correct or not. When we checked the all two ImageNet-validation images that was classified as “baby” by ViT, each of both images indeed contained a baby though its class in ImageNet was “Bathtub” (image ID: 00013344 and 00020254). Therefore, observed interference was not mistake but just the result of another classification. EfficientNet obviously shows greater numbers of interference (Figure 2(b)), but we also confirmed that similar thing happened, e.g., 198 of the 204 ImageNet-validation images classified as “baby” in EfficientNet contained human or doll.

As above, we found that DONE can add a new class from a single image with good accuracy for practical uses. We also found differences between ViT and EfficientNet. EfficientNet outperformed ViT in the accuracy of newly-added classification, but the number of interference of original ImageNet classification in EfficientNet was greater than that in ViT. We below show that this greater accuracy of EfficientNet does not mean EfficientNet is better suited for DONE than ViT.

### 4.2 Multi-class addition and K-shot learning

DONE was able to add a new class as above, but it might just be because the models recognized the new-class images as OOD, i.e., something else. Therefore, it is necessary to add multiple new similar
classes and check the classification among them. In addition, it is necessary to confirm whether the accuracy increase by increasing the number of training images just like the human brain, because in practical uses, users will prepare not just one training data but multiple training data for each class.

Specifically, we used one image from each of the eight classes and added new eight classes to the original 1000 classes, using DONE as one-shot learning. We evaluated this 1008-class model by 100 CIFAR test images for each of 8 classes and 10,000 ImageNet validation images. Figure 3(a) shows the results of the output of the representative model constructed by one-shot learning in which one image that showed median accuracy in Figure 2(b) was used as a standard training image of each class. In both backbone DNNs, the fraction of output of the correct class was the highest among the 1008 classes, and mean accuracy of the 8 classes was 51.8% and 61.1% in ViT and EfficientNet, respectively. That is, DONE was also able to classify newly added similar classes together with the original classes, in both DNNs.

Next, we increased the number of training images as K-shot learning. In the case of 10-shot learning (Figure 3(b)), each of the ten images was input to obtain each x, and the mean vector of the ten x vectors was converted into wj. For this representative 10-shot model, we used 10 images whose index in CIFAR-100 was from the front to the 10th in each class. We also tested 100-shot learning in the same way (Figure 3(c)). As a result, we found that such a simple averaging operation steadily improved the accuracy (mean accuracy of the 8 classes was 77.5% and 74.5% for 10 shot, and 86.1% and 82.1% for 100 shot, in ViT and EfficientNet, respectively). Averaging x would extract generalized features.

Moreover, we tested 100-class addition to the 1000-class models by one-shot learning with DONE, for investigating a scalability (Figure 3(d)), although there is overlap between ImageNet and CIFAR-100 classifications as above. Again we used images that showed median accuracy at one-class addition task such as shown in Figure 2(b) as a standard training image of each class. The accuracy of the 1100-class models for the classification of all 10,000 test images of CIFAR-100 was 37.2% and 26.3% (note that the chance level was not 1/100 but 1/1100), with decreasing the accuracy for the classification of all 50,000 ImageNet validation images by 0.1% and 2.9% from the original 1000-class model (65% and 69% as above), in ViT and EfficientNet, respectively. It is not clear whether these performance are at a practical level, but in any case, this task (1-shot 100-class addition to 1000 class) is expected to be difficult for humans as well.

We can see the difference in the backbone DNNs again. In EfficientNet (iii in Figure 3), compared to ViT (i in the Figure), ImageNet images were more often categorized to the new classes, i.e., interference occurred (0.03%, 0.02%, and 0.04% for ViT, and 1.27%, 2.55%, and 2.92% for EfficientNet, in 1, 10, 100-shot learning of 1008-class models, respectively). Therefore, greater accuracy of EfficientNet shown in Figure 2(b) does not mean EfficientNet is better suited for DONE than ViT, and it is considered that EfficientNet tends to recognize the new-class images as OOD when compared to ViT. We should understand why such interference happens (see next section). At any rate, with either backbone DNN, in practice, DONE was able to classify images into an integrated classification that includes both multiple new classes and multiple original classes, and also able to increase the accuracy by increasing the training data.

4.3 Difference between ViT and EfficientNet in DONE

EfficientNet more often showed interference in classification of the original-class images than ViT, even though DONE did not change the weights for the original classes. In previous studies, ViT is considered to be better at predictive uncertainty estimation [37, 23], more robust to input perturbations [34], and more suitable at classifying OODs [35] than convolutional neural networks. Such difference between ViT and EfficientNet (a convolutional neural network) may also appear in DONE. To compare ViT and EfficientNet in our case, we analyzed W matrix (wj, and wj vectors) of the one-shot 1008-class models shown in Figure 3(a) by Principal component analysis (PCA; Figure 4(a)).

In ViT, newly added wj vectors (black circles) were comparable to those of the original classes wj (colored circles), e.g., wj vector of a new class “Caterpillar (3 in Figure 4(a))” was near wj of original “Invertebrates” classes. On the other hand, in EfficientNet, newly added wj were all far from wj of original 1000 classes. Also, even when we got wj by inputting ImageNet images (red crosses), some of them were outside of the cluster of wj of original classes in EfficientNet, unlike
ViT. Therefore, the difference between $w_i$ and $w_j$ in EfficientNet is not just due to the difference between ImageNet and CIFAR.

These differences between $w_j$ (directly constructed from $x$) and original $w_i$ (optimized by ImageNet training) suggest that some properties of $x$ (neural activity) and $W$ (synaptic strength) are different in EfficientNet. Indeed, statistical properties of the elements of $x$ (Figure 4(b)) and $W$ (Figure 4(c)) are more different in EfficientNet than in ViT (the $L^2$ distance between the standardized distribution of $x$ from 10,000 ImageNet images and that of $W$ from 1000 classes was 28-fold greater in EfficientNet than ViT). These statistical properties themselves are normalized in DONE at the conversion of $x$ to $w_j$, nevertheless, such larger difference in $x$ and $W$ would be the cause of the observed results in EfficientNet because their differences complicate the cosine similarity at the original optimization.

In the case of 100-shot learning, $w_j$ went away from the cluster of original $w_i$ in both backbone DNNs (Figure 4(a), gray arrows), although their performance was better than one-shot learning. Therefore, 100-shot $w_j$ were considered to work somehow in a different way from the original $w_i$. Note that, nevertheless, DONE still works for the classification with both ViT and EfficientNet as shown above, which indicates the robustness of the application of DONE.

4.4 Finetuning tasks for understanding DONE

Finetuning is not a recommend task for DONE in practical uses, but it is convenient for the understandings. As above, accuracy of DONE should be the lowest and baseline in one-shot learning method. Thus, in finetuning instead of class addition, accuracy of one-shot learning by DONE is expected to be comparable to that by using basic optimization methods. To demonstrate it, we compared the accuracy between DONE and stochastic gradient descent (SGD), one of the simplest optimizer, in multi-class classification of CIFAR-10. Specifically, we eliminated the final dense layer of DNNs, froze the other parameters, and created the weight $W$ for the 10-way classification of CIFAR-10.

In one-shot learning by DONE, one image was input from each of the 10 classes, and then the $x$ was converted to a $w_j$ vector to construct $W$ matrix, which contains ten $w_j$ vectors. In this finetuning situation, since there was no existing original $W$, quantile normalization was performed so that every $w_j$ followed the standard normal distribution. We constructed 100 models with using 100 sets of
Figure 5: Finetuning tasks for understanding DONE. (a) Results of one-shot learning with DONE or SGD. (b) The results of K-shot learning. The small circle is the accuracy for each class, and the large circle is the average value. The inset graph is the relationship between the accuracy of each class obtained by DONE and SGD. (c) PCA of \(w_j\) vectors. Circles and crosses show the results obtained by one-shot learning with DONE and SGD, respectively. The color indicates the class. The initial and terminal points of the grey arrows indicate the results of one-shot and 1000-shot learning, respectively.

10 images (1 image/class). We evaluated the accuracy by using all 10,000 test images of CIFAR-10 (Figure 5(a), boxplot at epoch 1). The median accuracy of the 100 models was 68.2% and 47.6% in ViT and EfficientNet, respectively (gray line).

In one-shot learning by SGD, we used the set of 10 images that provided median accuracy in DONE, as a standard-images set. As a standard setting, the cross entropy error was minimized using the learning rate 0.01 (Tensorflow default) by increasing the number of epochs (Figure 5(a), blue circles). We found the obtained accuracy by enough epochs was similar to that obtained by DONE with both backbone DNNs. We also found that the \(w_j\) vectors became similar in terms of Pearson correlation coefficient between DONE and SGD optimization as the epoch is increased (gray circles for each class, orange circles for the mean). Note that we here used a basic setting for the optimization just for comparison, but of course learning efficiency can increase by tuning hyperparameters such as increasing the learning rate. In actual use of finetuning optimization, it may work well if the initial value of \(W\) is created by DONE from one or several training data before the optimization.

We also tested the case of 10, 100, and 1000-shot learning (Figure 5(b)), adjusting the number of epoch in SGD so that the number of inferences was 10,000 (i.e., 10 epochs for 1000-shot learning) with learning rate 0.1. Both methods show similar increase in accuracy (circle with line) and a difficult class for DONE was also difficult for SGD optimization (inset). SGD optimization outperformed DONE when the number of training data was large. This is probably because SGD optimization can make fine adjustments between multiple \(w_j\) vectors while each \(w_j\) vector in DONE is determined independently. Also, PCA shows \(w_j\) of DONE (circles) and SGD (crosses) were somehow different in PC1 and similar in PC2 (Figure 5(c)), in both backbone DNNs. Increasing the number of training data (gray arrows) made move further apart on PC1. As above, some similarity and differences between DONE and SGD optimization were observed in the finetuning task.

4.5 Evaluation of DNNs in DONE

DONE learning does not use optimization, does not have any randomness, and is reproducible for anyone when the same DNN and input data are used. Therefore, DONE is convenient for the evaluation of a feature of DNN, especially as a baseline accuracy in one-shot learning. Here, for example, we examined the 5-way (5 classes) 1-shot task of CIFAR-FS, which is a kind of standard task in one-shot classification. Specifically, we used each single image in 5 classes out of 100 classes
Table 1: 5-way classification accuracy on CIFAR-FS by 1-shot and 5-shot learning using DONE with various backbone DNNs

|         | ViT-B/32 | EfficientNet-B0 | ResNet-50 | MobileNetV2 | VGG16 |
|---------|----------|-----------------|-----------|-------------|-------|
| 1-shot  | 80.1 ± 9.3% | 69.4 ± 11.5% | 70.4 ± 10.6% | 63.1 ± 10.0% | 61.7 ± 10.7% |
| 5-shot  | 92.1 ± 5.2% | 85.5 ± 8.4%   | 85.7 ± 8.0% | 82.5 ± 7.5%  | 82.1 ± 8.4%  |

of CIFAR-100 for constructing a model, and evaluate the model by 15 images in each class. The combination of the 5 classes (and corresponding training images) was randomly changed in 100 times, and the mean accuracy and SD were obtained (Table 1). Also 5-way 5-shot task was tested in a similar way.

We found ViT performed the best accuracy for both 1-shot and 5-shot learning tasks. Note that other one-shot learning methods in previous study can obtain the same degree of accuracy without using such an excellent backbone DNN like ViT (e.g., 82% [40] and 87% [41] with ResNet-12), and thus DONE is not the best method for accuracy. However, these results suggest that DONE with ViT is already at a level of practical uses. Also, since DONE is completely dependent on the ability of backbone DNNs and further development of DNN is certain, the situation to obtain sufficient accuracy with DONE will soon come.

5 Conclusion

We proposed the simplest one-shot learning method DONE that allows us to add new classes with simple basis and operations without training optimization. DONE is very convenient in various ways as above, thus we expect it to be widely applied. DONE completely depends on well-trained DNNs, which would be similar to the case of brain as the product of training through learning and evolution. Like this study, searching for tasks that are easy for brain but difficult for machine learning is useful not only for understanding the brain but also for developing technology. For example, DONE can be used to increase the new classes and to detect abnormal values as ODD, such as in situation of autonomous driving, video surveillance, cancer diagnosis, and bacteria identification. Moreover, since the basis is simple and universal, it would be used for various hi-dimensional input such as voice data and state of communication-network routing. This study has just proposed DONE, and its scalability and expected applications are yet to be elucidated, therefore, we look forward to future developments of DONE.

6 Acknowledgments

We thank Drs. Toshio Yanagida, Toshiyuki Kanoh, Kunihiko Kaneko, Tsutomu Murata, and Tetsuya Shimokawa for helpful advice and discussions. This work was supported in part by MIC under a grant entitled “R&D of ICT Priority Technology (JPMI00316)” and JSPS KAKENHI Grant Number JP20H05533.

References

[1] D. Amaratunga and J. Cabrera, “Analysis of Data From Viral DNA Microchips,” Journal of the American Statistical Association, vol. 96, no. 456, pp. 1161–1170, Dec. 2001, doi: 10.1198/016214501753381814.

[2] D. O. Hebb, The Organization of Behavior: A Neuropsychological Theory. New York: Psychology Press, 2002. doi: 10.4324/9781410612403.

[3] B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed, “A comparison of normalization methods for high density oligonucleotide array data based on variance and bias,” Bioinformatics, vol. 19, no. 2, pp. 185–193, Jan. 2003, doi: 10.1093/bioinformatics/19.2.185.

[4] M. Carandini et al., “Do We Know What the Early Visual System Does?,” J. Neurosci., vol. 25, no. 46, pp. 10577–10597, Nov. 2005, doi: 10.1523/JNEUROSCI.3726-05.2005.

[5] L. Fei-Fei, R. Fergus, and P. Perona, “One-shot learning of object categories,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pp. 594–611, Apr. 2006, doi: 10.1109/TPAMI.2006.79.
[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in 2009 IEEE conference on computer vision and pattern recognition, 2009, pp. 248–255.

[7] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” 2009.

[8] K. Q. Weinberger and L. K. Saul, “Distance Metric Learning for Large Margin Nearest Neighbor Classification,” J. Mach. Learn. Res., vol. 10, pp. 207–244, Jun. 2009.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in Advances in Neural Information Processing Systems, 2012, vol. 25. Available: https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b7e8c436e924a6c45b-Abstract.html

[10] I. Goodfellow et al., “Generative Adversarial Nets,” in Advances in Neural Information Processing Systems, 2014, vol. 27. Available: https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494e97b1afccf3-Abstract.html

[11] D. L. K. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo, “Performance-optimized hierarchical models predict neural responses in higher visual cortex,” Proceedings of the National Academy of Sciences, vol. 111, no. 23, pp. 8619–8624, Jun. 2014. doi: 10.1073/pnas.1403112111.

[12] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems.” 2015.

[13] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in ICML deep learning workshop, 2015, vol. 2, p. 0.

[14] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. Available: http://arxiv.org/abs/1409.1556

[15] O. Russakovsky et al., “ImageNet Large Scale Visual Recognition Challenge,” arXiv:1409.0575, Jan. 2015. Available: http://arxiv.org/abs/1409.0575

[16] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, “Human-level concept learning through probabilistic program induction,” Science, vol. 350, no. 6266, pp. 1332–1338, Dec. 2015. doi: 10.1126/science.aab3050.

[17] M. Andrychowicz et al., “Learning to learn by gradient descent by gradient descent,” in Advances in Neural Information Processing Systems, 2016, vol. 29. Available: https://proceedings.neurips.cc/paper/2016/hash/9ef2ed4b7fd2c8108477df5a5f85bce38-Abstract.html

[18] O. Vinyals, C. Blundell, T. Lillicrap, koray kavukcuoglu, and D. Wierstra, “Matching Networks for One Shot Learning,” in Advances in Neural Information Processing Systems, 2016, vol. 29. Available: https://papers.nips.cc/paper/2016/hash/90e1357833654983612fb05e3ec9148c-Abstract.html

[19] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[20] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles,” in Advances in Neural Information Processing Systems, 2017, vol. 30. Available: https://proceedings.neurips.cc/paper/2017/hash/9ef2ed4b7fd2c8108477df5a5f85bce38-Abstract.html

[21] J. Snell, K. Swersky, and R. Zemel, “Prototypical Networks for Few-Shot Learning,” in Advances in Neural Information Processing Systems, 2017, vol. 30. Available: https://proceedings.neurips.cc/paper/2017/hash/ceb8da6767461f2812ae4290eac7cbe42-Abstract.html

[22] C. Finn, P. Abbeel, and S. Levine, “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks,” in Proceedings of the 34th International Conference on Machine Learning, Jul. 2017, pp. 1126–1135. Available: https://proceedings.mlr.press/v70/finn17a.html

[23] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On Calibration of Modern Neural Networks,” in Proceedings of the 34th International Conference on Machine Learning, Jul. 2017, pp. 1321–1330. Available: https://proceedings.mlr.press/v70/guo17a.html

[24] E. Schwartz et al., “Delta-encoder: an effective sample synthesis method for few-shot object recognition,” in Advances in Neural Information Processing Systems, 2018, vol. 31. Available: https://papers.neurips.cc/paper/2018/hash/1f714726c817afa50457d810ae9d27a2e-Abstract.html

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, Jun. 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.
[26] L. Bertinetto, J. F. Henriques, P. Torr, and A. Vedaldi, “Meta-learning with differentiable closed-form solvers,” presented at the International Conference on Learning Representations, Sep. 2018. Available: https://openreview.net/forum?id=HyxnZh0ct7

[27] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, “Meta-Transfer Learning for Few-Shot Learning,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 403–412. Available: https://openaccess.thecvf.com/content_CVPR_2019/html/Sun_Meta-Transfer_Learning_for_Few-Shot_Learning_CVPR_2019_paper.html

[28] M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in Proceedings of the 36th International Conference on Machine Learning, May 2019, pp. 6105–6114. Available: https://proceedings.mlr.press/v97/tan19a.html

[29] M. Kaya and H. S. Bilge, “Deep Metric Learning: A Survey,” Symmetry, vol. 11, no. 9, Art. no. 9, Sep. 2019, doi: 10.3390/sym11091066.

[30] T. Deleu, T. Wurfl, M. Samiei, J. P. Cohen, and Y. Bengio, “Torchmeta: A Meta-Learning library for PyTorch,” arXiv:1909.06576, Sep. 2019, Available: http://arxiv.org/abs/1909.06576

[31] W. Xue and W. Wang, “One-Shot Image Classification by Learning to Restore Prototypes,” AAAI, vol. 34, no. 04, pp. 6558–6565, Apr. 2020, doi: 10.1609/aaaiv34i04.6130.

[32] N. Eshed, “Novelty Detection and Analysis in Convolutional Neural Networks,” May 2020, doi: 10.7298/7273-6v18.

[33] A. Dosovitskiy et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” presented at the International Conference on Learning Representations, Sep. 2020. Available: https://openreview.net/forum?id=YicbFdNTTy

[34] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit, “Understanding Robustness of Transformers for Image Classification,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10231–10241. Available: https://openaccess.thecvf.com/content/ICCV2021/html/Bhojanapalli_Understanding_Robustness_of_Transformers_for_Image_Classification_ICCV_2021_paper.html

[35] S. Fort, J. Ren, and B. Lakshminarayanan, “Exploring the Limits of Out-of-Distribution Detection,” in Advances in Neural Information Processing Systems, 2021, vol. 34, pp. 7068–7081. Available: https://proceedings.neurips.cc/paper/2021/hash/3941c4358616274ac2436aacf67fae05-Abstract.html

[36] Y. Hu, V. Grigon, and S. Pateux, “Leveraging the Feature Distribution in Transfer-Based Few-Shot Learning,” in Artificial Neural Networks and Machine Learning – ICANN 2021, Cham, 2021, pp. 487–499. doi: 10.1007/978-3-030-86340-1_39.

[37] M. Minderer et al., “Revisiting the Calibration of Modern Neural Networks,” in Advances in Neural Information Processing Systems, 2021, vol. 34, pp. 15682–15694. Available: https://papers.nips.cc/paper/2021/hash/8420d359404024567ba5e6c1231af24-Absract.html

[38] M. Huisman, J. N. van Rijn, and A. Plaat, “A survey of deep meta-learning,” Artif Intell Rev, vol. 54, no. 6, pp. 4483–4541, Aug. 2021, doi: 10.1007/s10462-021-10004-4.

[39] J. Yang, K. Zhou, Y. Li, and Z. Liu, “Generalized Out-of-Distribution Detection: A Survey,” arXiv:2110.11334, Oct. 2021, Available: http://arxiv.org/abs/2110.11334

[40] B. Zhang et al., “Sample-Centric Feature Generation for Semi-Supervised Few-Shot Learning,” IEEE Transactions on Image Processing, vol. 31, pp. 2309–2320, 2022, doi: 10.1109/TIP.2022.3154938.

[41] Y. Bendou et al., “EASY: Ensemble Augmented-Shot Y-shaped Learning: State-Of-The-Art Few-Shot Classification with Simple Ingredients,” arXiv:2201.09699, Feb. 2022, Available: http://arxiv.org/abs/2201.09699

[42] P. Iakubovskii, EfficientNet Keras (and TensorFlow Keras), Available: https://github.com/qubvel/efficientnet

[43] F. Morales, vit-keras. Available: https://github.com/faustomorales/vit-keras

[44] opencv-python: Wrapper package for OpenCV python bindings. Available: https://github.com/skvark/opencv-python