It's DONE: Direct ONE-shot learning with Hebbian weight imprinting

Anonymous Author(s)
Affiliation
Address
email

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 with a nonparametric weight imprinting, named Direct ONE-shot learning (DONE). DONE adds new classes to a pretrained deep neural network (DNN) classifier with neither training optimization nor pretrained-DNN 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, by transforming all statistical properties of the neural activity into those of synaptic strength. DONE requires just one inference for learning a new concept 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 initially inspired by the biological neural network in the animal brain. Subsequently, Deep Neural Networks (DNNs) achieved great success in computer vision [9, 14, 20] 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 [5, 17, 19, 22, 27]. Humans can add a new class to their 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, 16] that learned 1000 classes can learn a new class "baby" from one image of a baby with neither additional training optimization nor pretrained-DNN modification, it will be useful in actual machine learning uses.

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 [4]. Also, another known

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

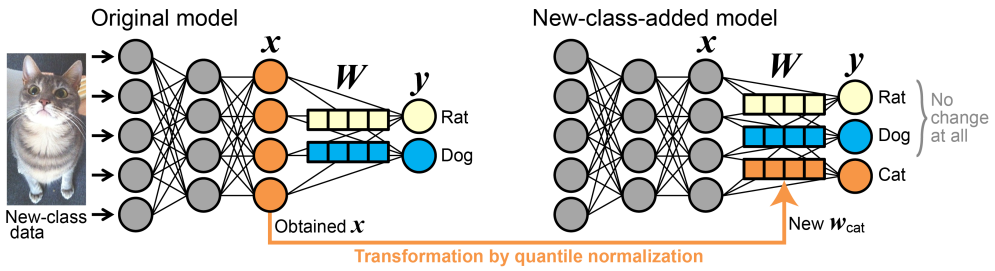

Figure 1: Scheme of DONE. The neural activity input of final dense layer (orange $\boldsymbol{x}$ vector in original model) obtained from a new-class data (an image of a cat) is directly used as a new-class vector (orange $\boldsymbol{w}_{\mathrm{cat}}$) in the new weight matrix ($\boldsymbol{W}$) to obtain the new output vector ($\boldsymbol{y}$) without any modification to the original backbone model. See text for explanation.

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 [11]. These knowledge of the brain inspired the idea that one-shot learning is possible for an existing high-performance DNN itself.

For a DNN model trained with sufficiently rich set of images, a reasonable representation of unknown images must exist somewhere in the hidden multi-dimensional space. Indeed, *weight imprinting*, proposed by Qi et al. [26], can add novel classes to Convolutional Neural Networks (CNNs) by using final-dense-layer input of a new-class image without extra training. Qi's weight imprinting method needs just a few CNN-architecture modifications and can provide decent accuracy in a one-shot image classification task (e.g., accuracy for novel-class images was 21% when novel 100 classes were added to original 100 classes in CUB-200-2011 dataset). Moreover, some studies [21, 36] show that the capabilities of DNN itself have the potential to enable Out-of-Distribution Detection (OOD).

In this paper, we introduce a very simple method, Direct ONE-shot learning (DONE). As shown in Figure 1, DONE directly transform the input of the final dense layer ($\boldsymbol{x}$ in Figure 1) obtained by one input image belonging to a new class, into the weight vector for the new additional class $\boldsymbol{w}_{\mathrm{new}}$ (a row vector of the weight matrix $\boldsymbol{W}$). Then, it is done. DONE uses weight imprinting but never modify backbone DNN unlike Qi's method. Qi's method was inspired by the context of metric learning, but DONE was inspired by Hebbian theory [2]. This difference in inspiration source makes a small but important difference in method procedures.

Specifically, Hebbian theory states that a synaptic connection (weight) is strengthened when both its presynaptic and postsynaptic neurons are active simultaneously. When a single image of a new class, e.g., a cat (in Figure 1), is presented as a visual input, some of the presynaptic neurons of final dense layer become active (i.e., $\boldsymbol{x}$). Simultaneously, a postsynaptic neuron corresponding to cat is active ($\boldsymbol{y}$ for cat) because the training image is known to be a cat. Thus, the synaptic connections between the active presynaptic neurons and the postsynaptic neuron will be strengthened according to Hebbian theory. 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 $\boldsymbol{x}$ as the weight vector $\boldsymbol{w}_{\mathrm{cat}}$ for the newly added "cat" output neuron in the output vector $\boldsymbol{y}$. This conversion of $\boldsymbol{x}$ to $\boldsymbol{w}_{\mathrm{cat}}$ would not be linear because neural activity and synaptic strength are different in dimension. DONE takes into account this nonlinear conversion by quantile normalization [1, 3] (see Methodology section). One-shot learning will work by such a simple procedure because the activity pattern in $\boldsymbol{x}$ by the cat image will be different from cases where other classes of images (e.g., rats and dogs) are presented, assuming that the whole neural network is trained to be able to classify various natural concepts.

Our method's basis and procedure are very simple but it achieved a good performance and accuracy. 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) [34] or EfficientNet [29]) as a backbone model (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 over other weight imprinting methods are (i) Hebbian-inspired simpler basis and procedure, (ii) no modification to backbone models, and (iii) nonparametric procedure for wide range of backbone models including future models. The advantages of DONE as a weight imprinting are (iv) no optimization thus little calculation cost and (v) no parameters or hyperparameters thus reproducible for anyone. In addition to proposing the new methods, this paper contains the following useful information for practical use: a generic task to add new classes to 1000-class ImageNet models, and the difference in backbone DNNs, specifically, between a Transfomer (ViT) and a CNN (EfficientNet). Given the above advantages and useful information, this paper will contribute to widespread application in the current situation where weight imprinting 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 backbone DNNs. Moreover, 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 at least in image classification tasks."

2 Related work

2.1 One-shot and few-shot image classification

Typical learning approaches for one- or few-shot image classification are metric learning, data augmentation, and meta learning. Weight imprinting has come out from metric learning, but actually it is essentially different because it does not require optimization as explained below. Each of these approaches has its own advantages and purposes, and they are not contradictory but can be used in a mixed manner.

Metric learning uses a classification at a feature space as a metric space [8, 22, 30]. 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 [19, 27, 43]. 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 [18, 23, 41]. 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 [40].

2.2 Weight imprinting

Weight imprinting is a learning method that initially arose from an innovative idea "learning without optimization" [26], and DONE is a type of weight imprinting. It is meaningless to compare the above three approaches with weight imprinting, because weight imprinting does not contain any optimization algorithm. Therefore, in principle, there is no reason for weight imprinting methods to outperform other methods by themselves in accuracy. The performance of weight imprinting methods is uniquely determined by the backbone DNN without any randomness, hence its performance is suitable as a reference baseline for other methods. Thus, weight imprinting does not aim for the highest accuracy but for practical convenience and reference role as a baseline method.

We here explain the basis of weight imprinting and then specific procedure of Qi's method. Let us consider the classification at the final dense layer of DNN models in general. In most cases, the output vector $\boldsymbol{y}$ of the final dense layer denotes the degree to which an image belongs to each class and is calculated from the input vector of the final dense layer $\boldsymbol{x}$, weight matrix $\boldsymbol{W}$, and bias vector $\boldsymbol{b}$. Here, for i-th class in N classes ($i = 1, 2, \cdots, N$), a scalar y_i is calculated from the corresponding weight vector $\boldsymbol{w}_i$ (i-th row vector of $\boldsymbol{W}$ matrix) and bias scalar b_i as following equation:

$$y_i = \boldsymbol{x} \cdot \boldsymbol{w}_i + b_i = ||\boldsymbol{x}||_2 \, ||\boldsymbol{w}_i||_2 \cos\theta + b_i, \tag{1}$$

where the cosine similarity is a metric that represents how similar the two vectors $\boldsymbol{x}$ and $\boldsymbol{w}_i$ are irrespective of their size. Thus, this type of model contains cosine similarity as a part of its objective function. It is also possible to use the cosine similarity alone instead of the dot product [25].

Weight imprinting uses this basis of the cosine similarity. The cosine similarity will have the maximum value 1 if $\boldsymbol{x}$ and $\boldsymbol{w}_i$ are directly proportional. Thus, if a certain $\boldsymbol{x}$ is directly used for the weight of a new j-th class $\boldsymbol{w}_j$ ($j = N+1, \cdots$), the cosine similarity for j-th class becomes large when another $\boldsymbol{x}$ with a similar value comes.

In Qi's method, to focus only on the cosine similarity as a metric for the objective function, the backbone DNN models are modified in the following three parts:

- **Modification 1** : Adding L_2 normalization layer before the final dense layer so that $\boldsymbol{x}$ becomes unit vector, i.e., $||\boldsymbol{x}||_2 = 1$
- **Modification 2** : Normalizing all $\boldsymbol{w}_i$ to become unit vectors, i.e., $||\boldsymbol{w}_i||_2 = 1$ for all i.
- **Modification 3** : Ignoring all bias values b_i, i.e., $\boldsymbol{b}$ vector.

Then, the final-dense-layer input obtained from a new-class image $\boldsymbol{x}_{\mathrm{new}}$ (L_2-normalized, in Qi's method) are used as the weight vector for the new class $\boldsymbol{w}_j$, i.e.,

$$\boldsymbol{w}_j = \boldsymbol{x}_{\mathrm{new}}. \tag{2}$$

Qi's method is already simple and elegant, but it still requires some modifications to the backbone DNN, which involves changes in the objective function. Modification of the backbone DNN can be a problem, for example, in the following case, which would often occur now and in the future: the backbone DNN model is a very good model as a heritage of mankind and should not be changed as much as possible (especially for many non-expert users). Even small modifications can make a significant difference to the original performance of the backbone DNN. Also, Qi's method uses linear transformation for conversion of the embedding vector ($\boldsymbol{x}$ before normalization) into $\boldsymbol{w}_j$ as a result of focusing on the cosine similarity, without considering difference in statistical properties between $\boldsymbol{x}$ and $\boldsymbol{w}_j$, which limits the range of backbone DNNs used. There have been various researches that make Qi's method more complex and applicable [32, 38, 39, 42, 44], but to the best of our knowledge, none that make it simpler.

3 Methodology

3.1 Procedure, basis, and limitations of DONE

DONE does not modify backbone DNN and just directly apply $\boldsymbol{x}_{\mathrm{new}}$ into $\boldsymbol{w}_j$ ($j = N+1, \cdots$), as shown in Figure 1, as

$$\boldsymbol{w}_j = F(\boldsymbol{x}_{\mathrm{new}}, \boldsymbol{W}_{\mathrm{ori}}), \tag{3}$$

$$b_j = \tilde{\boldsymbol{b}}_{\mathrm{ori}}, \tag{4}$$

where $F(\boldsymbol{x}_{\mathrm{new}}, \boldsymbol{W}_{\mathrm{ori}})$ is a quantile normalization of $\boldsymbol{x}_{\mathrm{new}}$, using the information of the original weight matrix ($\boldsymbol{W}_{\mathrm{ori}}$) as the reference distribution, and $\tilde{\boldsymbol{b}}_{\mathrm{ori}}$ is the median of the original bias vector $\boldsymbol{b}_{\mathrm{ori}}$. Then, it is done. We call the process in Eq. 3 *Hebbian weight imprinting* as explained below.

The quantile normalization changes the value of the elements of $\boldsymbol{x}_{\mathrm{new}}$, so that the elements of $\boldsymbol{W}_{\mathrm{ori}}$ and resultant $\boldsymbol{w}_j$ have identical quantiles, using rank information of $\boldsymbol{x}_{\mathrm{new}}$ (which neuron) and value information of $\boldsymbol{W}_{\mathrm{ori}}$ (how much). Namely, all statistical properties of the elements of $\boldsymbol{w}_j$ and $\boldsymbol{W}_{\mathrm{ori}}$ are identical except the number of elements, i.e., those probability distributions are the same. Quantile normalization is an easy and standard technique in Bioinformatics [1, 3], and it is suitable for implementing Hebbian theory. Statistical properties of the elements of $\boldsymbol{x}$ (neural activity) and $\boldsymbol{W}$ (synaptic strength) can be different (e.g., difference in histograms between Figure 4 (b) and (c)). The conversion from $\boldsymbol{x}_{\mathrm{new}}$ to $\boldsymbol{w}_j$ should be a transformation of a vector into a different dimension so that statistical properties of the elements of $\boldsymbol{w}_j$ and $\boldsymbol{W}_{\mathrm{ori}}$ become similar because both are synaptic

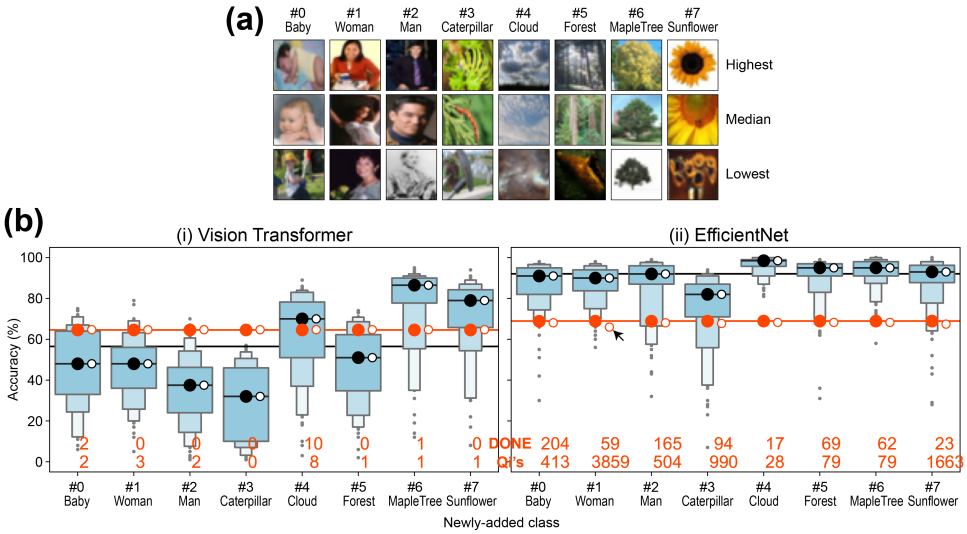

Figure 2: One-class addition by one-shot learning. (a) Representative images of the newly-added CIFAR-100 classes. Each image was chosen as a representative because the model that learned the image showed the highest, median, and lowest accuracy in each class in (b)-(i). (b) Accuracy of the one-class-added models obtained by one-shot learning with DONE or Qi's method.

strength. For example, we could apply linear transformation from $\boldsymbol{x}_{\mathrm{new}}$ into $\boldsymbol{w}_j$ so that the mean and variance (i.e., 1st and 2nd central moments) of the elements of $\boldsymbol{w}_j$ are the same as those of $\boldsymbol{W}_{\mathrm{ori}}$. 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 could be different depending on the backbone DNN. One of the simplest solution for every situation is to make all the statistical properties of the elements of $\boldsymbol{w}_j$ and $\boldsymbol{W}_{\mathrm{ori}}$ identical. Quantile normalization achieves it nonparametrically.

As limitations, DONE requires a neural network model that has dense layer for classification as above. The range of applicable models is yet unclear, but in principle it is wider than Qi's method.

3.2 Implementation and Dataset

As backbone models, we employed ViT [34] and EfficientNet [29] as two representative DNNs with different characteristics, using "vit-keras" [46] and "EfficientNet Keras (and TensorFlow Keras)" [45], respectively. Also, for transfer learning, we employed InceptionV1 [15] (employed in the paper by Qi et al. [26]) and ResNet-12 [20], using "Trained image classification models for Keras" [47] and Tensorflow [12], respectively. All models used in this study were pretrained with ImageNet.

We used CIFAR-100 and CIFAR-10 [7] for additional classes, using Tensorflow [12]. Also, for transfer learning, we used CIFAR-FS [28] by Torchmeta [31]. We used ImageNet (ILSVRC2012) images [6, 16] for testing the performance of models. We used information of 67 categorization [33] of ImageNet 1000 classes, for a coarse 10 categorization in Figure 4(a). All images were resized to (224×224) by OpenCV [48] or the preprocessing resizing layer of 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 $\boldsymbol{w}_j$ is generated from one image, thus the model

had 1001 classes (not 1008 classes here). To conduct stochastic evaluations, 100 different models were built by using 100 different training images for each additional class.

Figure 2(b) shows letter-value plots of the accuracy for each additional class (chance level $1/1001$). Black closed circles show the median accuracy of each class by DONE, and the mean of them were 56.5% and 92.1% for ViT and EfficinetNet, respectively (black line). When the mean accuracy was compared with the accuracy of ImageNet validation test by the original 1000-class model (orange line; 65% and 69%), the mean with ViT had no significant difference and the mean of EfficinetNet was significantly greater (one sample t-test; two-sided with α=0.05, in all statistical tests in this study), which suggests that the accuracy of the one-shot learning by DONE achieves a practical level.

An obvious fact in one-shot learning is that a bad training image produces a bad performance, 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, a user supposed to use a representative image for the training. We therefore think that the low performances due to a bad training image is not a significant issue.

We investigated the interference of the class addition with 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 difference between accuracy of the original 1000-class model (orange line) and the mean of accuracy of the eight 1001-class models (orange closed circles) were less than 1% (0.004% and 0.664% for ViT and EfficinetNet, respectively).

Figure 2(b) also shows the numbers of ImageNet validation images in which the output answer of the added model was the new class (thus incorrect) in the 50,000 images (orange numbers; upper for DONE). When we checked the all two 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 shows significantly greater numbers of interference than ViT (Wilcoxon signed-rank test), 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.

Moreover, we also compared DONE with Qi's method. Black and orange open circles show the results using Qi's method instead of DONE in the same tests shown as black and orange closed circles, respectively. Both the mean of the eight median accuracy and the mean of decreased accuracy of the eight median models in the ImageNet validation test were not significantly different between DONE and Qi's method (paired sample t-test). However, the number of the new-class answer of the added-model with EfficientNet in the ImageNet validation test was significantly greater by Qi's method (lower orange numbers; some exceeded 1000) than by DONE (Wilcoxon signed-rank test). Also, a significant outlier of decreased accuracy in the ImageNet validation test was observed (orange open circle depicted by a black arrow; Smirnov-Grubbs test). Thus, the interference is significantly greater in Qi's method than DONE, which is reasonable because Qi's method modifies the backbone model without considering the difference in distribution between $\boldsymbol{x}$ and $\boldsymbol{W}$.

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 showed greater interference with original ImageNet classification than ViT. We below show that the 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, because in practical uses, users will prepare not just one training data but multiple training data for each class.

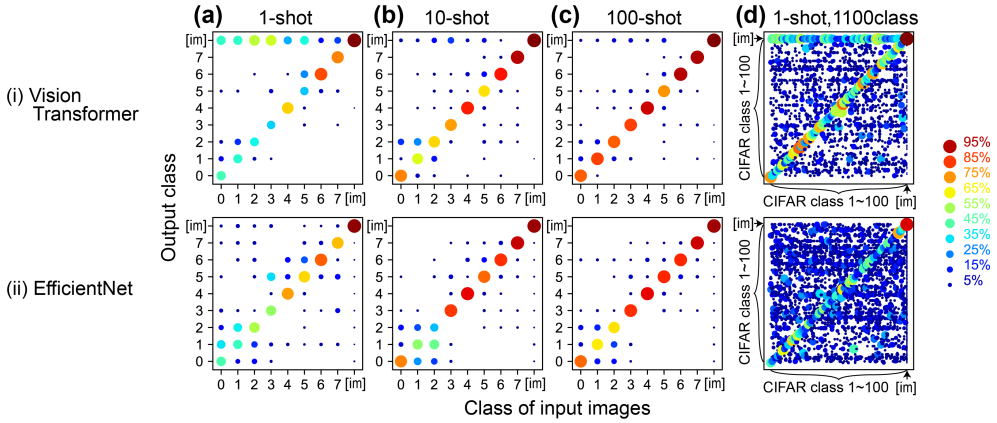

Figure 3: Multi-class addition and K-shot learning. (a), (b), and (c) show the results of the 1008-class model constructed by 1, 10, and 100-shot learning, respectively. (d) shows the results of the 1100-class model constructed by 1-shot learning. The horizontal and vertical axes show the class of the 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.

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 $\boldsymbol{x}$, and the mean vector of the ten $\boldsymbol{x}$ vectors was converted into $\boldsymbol{w}_j$, according to the Qi's method. 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 $\boldsymbol{x}$ 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. 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 ((ii) 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). Combined with another fact that

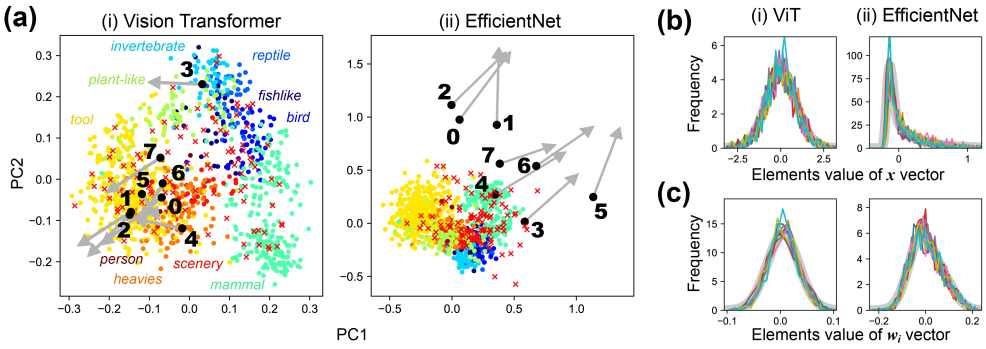

Figure 4: Difference between ViT and EfficientNet in DONE. (a) PCA of each $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ vector in the one-shot 1008-class model shown in Figure 3(a). Different colors show a coarse 10 categorization of the classes. (b) and (c) show the frequency distribution of $\boldsymbol{x}$ when the input were ImageNet images and $\boldsymbol{w}_i$, respectively. The colored and gray lines show 10 example vectors and average, respectively.

the accuracy in these multi-class additions was significantly lower than that in one-class addition only with EfficientNet (paired sample t-test), 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).

When we used Qi's method, the decreased accuracy of 1, 10, 100-shot 1008-class models with EfficientNet for ImageNet validation images was 10.5%, 14.3%, 13.5%, respectively. The mean of those decreased accuracy was significantly greater than that using DONE (paired sample t-test), while the difference was not significant with ViT. Also, there was no significant difference in the mean accuracy of 1, 10, 100-shot 1008-class models for the added 8 classes between DONE and Qi's method, with both backbone DNNs. Thus, again the interference in the case of EfficientNet is significantly greater in Qi's method than DONE, which indicates the wider range of applicable backbone DNN models for DONE.

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 [24, 37], more robust to input perturbations [35], and more suitable at classifying OODs [36] than CNNs like EfficientNet. Such difference between ViT and EfficientNet may also appear in DONE. To compare ViT with EfficientNet in our case, we analyzed $\boldsymbol{W}$ matrix ($\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ 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 $\boldsymbol{w}_j$ vectors (black circles, with the ID number of newly-added 8 classes) were comparable to those of the original classes $\boldsymbol{w}_i$ (colored circles), e.g., $\boldsymbol{w}_j$ vector of a new class "caterpillar (3 in Figure 4(a))" was near $\boldsymbol{w}_i$ of original "invertebrate" classes. On the other hand, in EfficientNet, newly added $\boldsymbol{w}_j$ were all far from $\boldsymbol{w}_i$ of original 1000 classes. Also, even when we got $\boldsymbol{w}_j$ by inputting ImageNet images (red crosses; validation ID from the front to the 100th), some of them were outside of the cluster of $\boldsymbol{w}_i$ of original classes in EfficientNet, unlike ViT. Therefore, the difference between $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ in EfficientNet is not just due to the difference beween ImageNet and CIFAR. These differences between $\boldsymbol{w}_j$ (constructed from $\boldsymbol{x}$) and original $\boldsymbol{w}_i$ (optimized by ImageNet training) might be due to the fact that statistical properties of the elements of $\boldsymbol{x}$ (Figure 4(b)) and $\boldsymbol{w}_i$ (Figure 4(c)) are more different in EfficientNet than in ViT because their differences complicate

Table 1: 5-way few-shot classification accuracy on CIFAR-FS with various backbone DNNs. Asterisks mean significant differences between DONE and Qi's method (Dwass-Steel-Critchlow-Fligner test).

	ViT-B/32	EfficientNet-B0	InceptionV1	ResNet-12
1-shot Qi's	$80.7 \pm 9.6\%$	$67.2 \pm 10.6\%$	$59.8 \pm 11.0\%$	$58.4 \pm 12.3\%$
1-shot DONE	$81.4 \pm 10.3\%$	$71.7 \pm 11.9\%$*	$65.1 \pm 10.8\%$*	$60.8 \pm 11.8\%$
5-shot Qi's	$92.3 \pm 4.2\%$	$83.0 \pm 8.5\%$	$81.6 \pm 7.7\%$	$79.0 \pm 8.7\%$
5-shot DONE	$92.6 \pm 4.6\%$	$86.4 \pm 7.5\%$*	$82.4 \pm 8.2\%$	$79.5 \pm 7.6\%$

the cosine similarity at the original optimization, although these statistical properties themselves are normalized in DONE at the conversion of $\boldsymbol{x}$ to $\boldsymbol{w}_j$.

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

4.4 Transfer few-shot learning

DONE is recommended for easy addition of new classes, not for transfer learning. However, DONE is convenient for the evaluation of DNN especially as a baseline accuracy in few-shot learning. 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 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 with standard deviation were obtained (Table 1). Also 5-way 5-shot task was tested in a similar way.

We found ViT significantly outperformed the other DNNs in all conditions (Dwass-Steel-Critchlow-Fligner test). Note that DONE is a baseline method and other one-shot learning methods with training can obtain similar degree of accuracy with smaller backbone DNN (e.g., 82% [41] and 87% [43] with ResNet-12). Compared to Qi's method, DONE shows significantly greater accuracy with some models, and never significantly worse. Thus DONE outperformed Qi's method also in accuracy, although we did not expect it.

Anyhow, the results suggest that DONE with ViT is already at a level of practical uses. 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 and Future work

This paper has proposed DONE, the simplest one-shot learning method that allowed us to add new classes to a pretrained DNN at a practical-level accuracy without optimization or modification of the DNN. Good performance of DONE completely depends on well-trained DNNs, which is similar to the case of brain as the product of training through learning and evolution. DONE is a one-shot learning method that applies Hebbian weight imprinting, which is an implementation of Hebbian theory by quantile normalization, to the final dense layer of a DNN model. Given the simplicity and wide applicability, not only DONE but also Hebbian weight imprinting alone are expected to be applied in a wide range of the field of neural networks, including those other than one-shot learning tasks. This study has just proposed the method, and its scalability and expected applications are yet to be elucidated. We look forward to future developments of DONE and Hebbian weight imprinting, as well as the further understanding of the brain with them.

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/c399862d3b9d6b76c8436e924a68c45b-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/5ca3e9b122f61f8f06494c97b1afccf3-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] C. Szegedy et al., "Going Deeper With Convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html

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

[17] 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.

[18] 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/fb87582825f9d28a8d42c5e5e5e8b23d-Abstract.html

[19] 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

[20] 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.

[21] 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/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html

[22] 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/cb8da6767461f2812ae4290eac7cbc42-Abstract.html

[23] 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

[24] 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

[25] C. Luo, J. Zhan, X. Xue, L. Wang, R. Ren, and Q. Yang, "Cosine Normalization: Using Cosine Similarity Instead of Dot Product in Neural Networks," in Artificial Neural Networks and Machine Learning - ICANN 2018, Cham, 2018, pp. 382–391. doi: 10.1007/978-3-030-01418-6_38.

[26] H. Qi, M. Brown, and D. G. Lowe, "Low-Shot Learning With Imprinted Weights," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5822–5830. Accessed: May 02, 2022. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Qi_Low-Shot_Learning_With_CVPR_2018_paper.html

[27] 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://proceedings.neurips.cc/paper/2018/hash/1714726c817af50457d810aae9d27a2e-Abstract.html

[28] 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

[29] 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

[30] 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.

[31] 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

[32] G. S. Dhillon, P. Chaudhari, A. Ravichandran, and S. Soatto, "A Baseline for Few-Shot Image Classification," presented at the International Conference on Learning Representations, Sep. 2019. Accessed: May 18, 2022. Available: https://openreview.net/forum?id=rylXBkrYDS

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

[34] 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

[35] 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

[36] 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/3941c4358616274ac2436eacf67fae05-Abstract.html

[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/8420d359404024567b5aefda1231af24-Abstract.html

[38] N. Passalis, A. Iosifidis, M. Gabbouj, and A. Tefas, "Hypersphere-Based Weight Imprinting for Few-Shot Learning on Embedded Devices," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 2, pp. 925–930, Feb. 2021, doi: 10.1109/TNNLS.2020.2979745.

[39] Y. Li et al., "Classification Weight Imprinting for Data Efficient Object Detection," in 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE), Jun. 2021, pp. 1–5. doi: 10.1109/ISIE45552.2021.9576432.

[40] 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.

[41] 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.

[42] P. Cristovao, H. Nakada, Y. Tanimura, and H. Asoh, "Few Shot Model based on Weight Imprinting with Multiple Projection Head," in 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM), Jan. 2022, pp. 1–7. doi: 10.1109/IMCOM53663.2022.9721726.

[43] 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

[44] H. Zhu et al., "Weight Imprinting Classification-Based Force Grasping With a Variable-Stiffness Robotic Gripper," IEEE Transactions on Automation Science and Engineering, vol. 19, no. 2, pp. 969–981, Apr. 2022, doi: 10.1109/TASE.2021.3054655.

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

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

[47] M. Andrews, Trained image classification models for Keras. Available: https://github.com/mdda/deep-learning-models

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

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Methodology.
   (c) Did you discuss any potential negative societal impacts of your work? [No] There are no new negative societal impacts created by our method.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Not yet but they will be submitted as supplemental material later.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] Because the amount of compute is small and no special hardware is required. Information about them will be included in the supplemental material later.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] Because they are all OSS.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] Not yet but they will be submitted as supplemental material later.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] Because they are all OSS.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] Because they are all OSS.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

Code and data will be submitted as supplemental material later.