Training Data Reduction using Support Vectors for Neural Networks

Toranosuke Tanio*, Kouya Takeda*, Jeahoon Yu*, and Masanori Hashimoto*
* Osaka University, Osaka, Japan
E-mail: {t-tanio, k-takeda, yu.jaehoon, hasimoto}@ist.osaka-u.ac.jp

***Abstract*—In the field of machine learning, deep learning is widely used to improve versatility and accuracy. Deep learning has a higher expression ability compared with conventional models but requires large amounts of data and time for training. To tackle this issue, we propose a training data reduction method using support vectors that are the closest data to the classification boundary obtained by support vector machine. The proposed method chooses a subset of training data consisting of support vectors and uses them for training neurak networks. Experimental evaluation shows that it is possible to reduce the number of training data by 12% and reduce the learning time of neural network by 9.5% in a test case of ResNet with CIFAR-10 dataset.**

I. Introduction

With the advent of deep learning, the development of machine learning based on neural networks has reached its heyday. In multilayer perceptron, one hidden layer performs an approximation of an arbitrary function by combining nonlinear transformation. However, in deep neural network (DNN), the expressive ability of the neural network is dramatically improved by increasing the number of hidden layers and performing an iterative nonlinear transformation. In the massive image recognition competition ILSVRC [1], since AlexNet [2] was proposed, DNNs have shown the remarkable increase in their inference performance. These networks make it possible to approximate complex functions and are expected to be applicable to practical problems. However, recent neural networks using billions of parameters require a large amount of computing resource, training data, and training time. Fig. 1 shows the annual layer increase of ILSVRC winners, where the size of the neural network is increasing exponentially. Due to this scale explosion, DNNs require more data for training, which causes serious problems for practical use.

In this research, to tackle this problem, we propose a method that reduces training data for DNN while maintaining accuracy. A basic idea of the proposed method is to exploit differences of importance in training data and improve training efficiency by performing the training process only with training data having high importance. To archive this, we focus on the role of support vectors in the support vector machine (SVM) [3]. Support vectors refer to a group of data defining a boundary hyperplane for spatial separation in SVM. Therefore, support vectors locate at the outermost part of the training data in the same class, which are expected to play an essential role in the learning of neural networks. While existing research on training data reduction using support vectors has provided a classification result for MNIST [4], it does not clarify the learning efficiency for more complicated models. In this study, we aim to explain the effect of support vectors in DNN learning using ResNet [5] and CIFAR-10, which are a recent neural network and an image classification dataset, respectively.

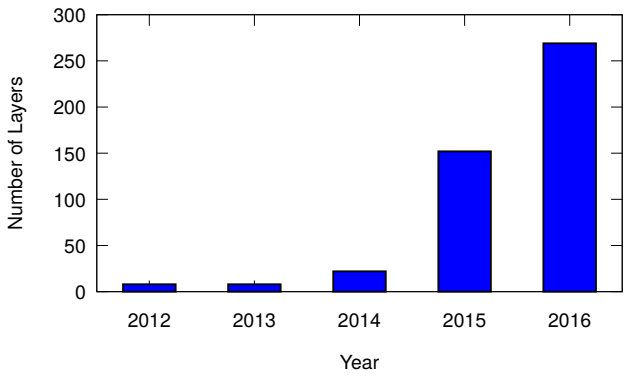


Fig. 1. Annual layer increase of ILSVRC winners.

The rest of this paper is as follows. Section II explains the concept of deep learning, the basics of neural networks, and SVM used in this research. Next, Section III describes the proposed training data reduction method using support vectors and confirms the effects of support vectors in neural network learning with preliminary experiments on multiple two-dimensional data. Section IV shows the results of evaluation experiments using ResNet and CIFAR-10 dataset and discusses remaining issues. Finally, Section V concludes this paper.

II. Related research

This section describes deep learning and SVMs as related research required to understand the proposed method.

A. Deep learning

Deep learning is a machine learning method based on DNNs. A major difference between DNNs and conventional neural networks is in the number of hidden layers. Whereas conventional neural networks have only one hidden layer, DNNs have two or more hidden layers. Their deeper structure makes it possible to improve inference accuracy but it requires massive training data, processing time, and power consumption.

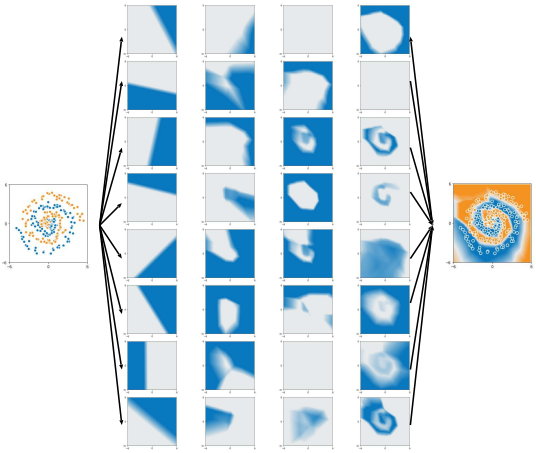


Fig. 2. a classification process of the spiral dataset.

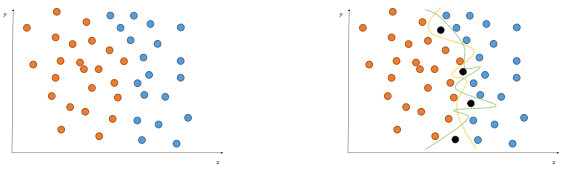
DNNs are also largely different from other conventional machine learning methods such as SVM and ensemble learning. Conventional machine learning methods require manually engineered feature extraction. However, DNNs can extract features by itself using nonlinear transformation performed in each layer. As an example, Fig. 2 shows a classification process of the spiral dataset. In Fig. 2, the leftmost and rightmost boxes represent input data and trained space, respectively, and each box between them represents the trained space in each neuron of hidden layers. As shown in Fig. 2, DNNs can directly learn non-linearly separated spaces from raw data.

B. Support vector machine

SVM [3] is one of conventional machine learning methods and can handle both classification and regression problems. This section explains the concept of SVM by taking a classification example shown in Fig. 3(A). This figure depicts the distribution of 2-D training dataset with two classes colored with orange and blue. Even with this simple dataset, there exist an infinite number of separation hyperplanes for classification. Fig. 3(B) shows orange and green lines as examples of available hyperplanes. Both hyperplanes correctly separate training data into two groups. However, once black data points are given for inference, the green line incorrectly classifies them in this particular example. This problem is known as overfitting. SVM handles the overfitting problem by using regularization and finds out the optimal separation hyperplane with the maximum margin between two groups of training data. Support vectors are the nearest data to its separation hyperplane. Fig. 4 shows an example of support vectors in a training dataset, and the proposed method described in Section III utilizes these support vectors.

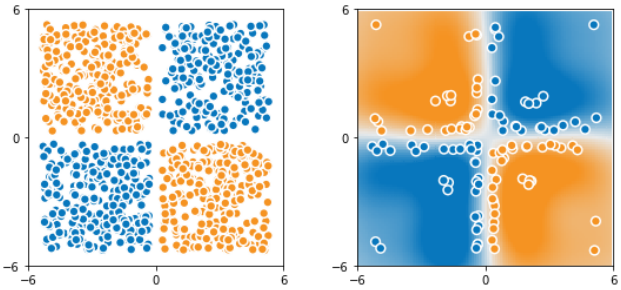
For handling nonlinearly separable problems, SVM uses nonlinear transformation called kernel. Its basic idea is simple. As an example, circularly distributed nonlinear data shown in Fig. 5(A) cannot be linearly classified, but after transforming the input space with

$$(z_1, z_2, z_3) = (x^2, y^2, \sqrt{2}xy), \tag{1}$$



(A) training data (B) available hyperplanes

Fig. 3. Example of different classification hyperplanes for the same data.



(A) training data (B) support vectors

Fig. 4. Training data and support vectors in XOR dataset.

we can get a linearly separable space shown in Fig. 5(B). In SVM, two types of kernels, radial basis function (RBF) and polynomial kernels, are widely used for dealing with nonlinear problems. First, the RBF kernel is defined as

$$K_p(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2). \tag{2}$$

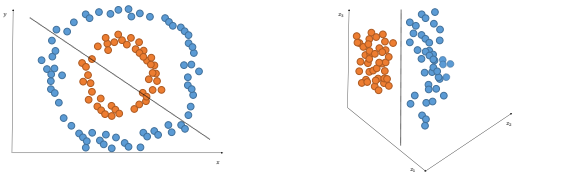
The RBF kernel is the most frequently used kernel and it has γ as hyperparameter. Next, the polynomial kernel is defined as

$$K_p(x_i, x_j) = x_i x_j + r^d. \tag{3}$$

Obiously, this kernel has a polynomial of order d, where r is an added hyperparameter. Sections III and IV show how effective these two kernel functions are for reducing training data.

C. Training data reduction method

There are two existing studies on training data reduction [6][7]. The first study is conducted by Nguyen et al., and it uses support vectors to examine the influence of training data reduction on classification accuracy with 2-D data [6]. This study evaluated its effectiveness with three different



(A) linearly inseparable data (B) linearly separable data

Fig. 5. Kernel effect on nonlinear problems.

TABLE I
RESULTS OF EXPERIMENTS CONDUCTED BY DAHIYA ET AL.

training data	method	SVM kernel	accuracy(%)
60,000	Original	-	97.62
20,000	Suport vector	RBF	97.66
10,000	Suport vector	Polynomial	97.48
20,000	Random	-	95.32
10,000	Random	-	94.67

data distributions: Gaussian, sine, and ellipse, and showed the trade-off between reduction amount and classification accuracy when using support vectors instead of entire training data. This work provides an upper limit on how much data reduction is possible by the support vectors, and has shown that support vectors can reduce an appropriate amount of training data without much deterioration in acuuracy.

The second study conducted by Dahiya et al. [7] also reports training data reduction using support vectors for neural networks. In this study, using support vectors instead of entire training data maintained accuracy against the classification problem of handwritten numbers called MNIST [8]. TABLE I shows its experimental results. This study reported that accuracy degradation did not occur while SVMs with the RBF kernel and the polynomial kernel reduced the training data by two third and four fifth, respectively. As a control experiment, this study also evaluated the case using randomly sampled training data and reported 2 to 3% accuracy degradation.

III. TRAINING DATA REDUCTION IN DEEP LEARNING

This section proposes a training data reduction method using support vectors for DNNs. The proposed method focuses on the high possibility that support vectors include important training data.

A. Training data reduction using support vectors

Fig. 6 compares the training processes of the proposed method and conventional one. The procedure of the proposed method consists of:

1) Train SVM with training dataset D,
2) Extract support vectors D_{SV}, and
3) Train neural networks using D_{SV} as training data.

The SVM training of 1) requires additional computational time, but it is much smaller than the amount of training time reduction of neural network in 3). Therefore, the overall training time can be reduced.

Training data reduction has a trade-off relationship with accuracy. Therefore, aggressive reduction has a risk causing serious degradation of inference accuracy. Also, even if support vectors successfully reduce training data with ignorable accuracy degradation, the accuracy needs to be better than that of randomly extracted training data. This situation can occur in simple datasets or excessively redundant datasets.

B. Preliminary experiment

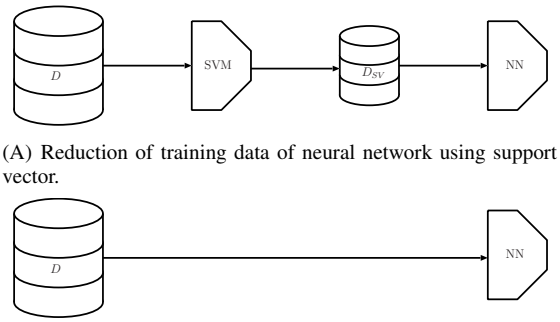
TensorFlow Playground is an educational content and available online[1], visualizing the learning process of the neural network. A preliminary experiment utilizes similar visualization with Tensorflow Playground. The purpose of the experiment is to confirm the effectiveness of the proposed method. The program implemented for the preliminary experiment can train a neural network with 2-D data and can change the activation function, the number of hidden layers, and the number of neurons arbitrarily. The experimental setup is described in TABLE II. The datasets used in this section are two regression and four classification problems: Gaussian (R_GAUSS) and straight (R_PLANE) for regression and Gaussian (C_GAUSS), spiral (C_SPIRAL), circle (C_CIRCLE), and XOR (C_XOR) for classification problems.

In the preliminary experiment, we evaluate the classification accuracy, regression accuracy, and learning time in the following three cases:

1) Learning using all the training data,
2) Learning using only support vectors, and
3) Learning using randomly extracted data as many as support vectors.

The number of epochs is set to 100 except C_SPIRAL. The number of epochs for C_SPIRAL is set to 500 because it is more complicated than the other datasets. Accuracy is evaluated by loss: smaller loss represents higher accuracy. TABLE III shows the result of the preliminary experiment.

In C_GAUSS, using support vectors does not degrade classi-



(A) Reduction of training data of neural network using support vector.

(B) Normal neural network learning.

Fig. 6. Reduction of training data by support vector.

TABLE II
SETTINGS IN PRELIMINARY EXPERIMENT.

Act. Func.	# Layers	# Neurons in Each Layer
ReLU	3	5

TABLE III
RESULTS OF PRELIMINARY EXPERIMENTS.

dataset	(A) All training data		(B) Support vector		(C) Random	
	data	loss	data	loss	data	loss
C_GAUSS	1,000	0.001	126	0.001	126	0.001
R_PLANE	2,400	0.004	292	0.001	292	0.006
R_GAUSS	2,400	0.192	263	0.011	263	0.563
C_SPIRAL	1,000	0.009	137	0.039	137	0.105
C_CIRCLE	1,000	0.001	129	0.002	129	0.177
C_XOR	1,000	0.003	117	0.001	117	0.005

[1] https://playground.tensorflow.org/

TABLE IV
RESULT OF APPLYING SUPPORT VECTOR FOR CIFAR-10 (FIVE LABELS).

training data	data	accuracy(%)	time(sec)
(A) all data	25,000	78.300	1058.243
(B-1) support vector(one-against-one)	22,035	78.580	961.394
(B-2) support vector(one-against-all)	21,975	78.340	957.629
(C-1) random	22,035	77.400	962.348
(C-2) random	21,975	77.260	959.526

TABLE V
RESULT OF APPLYING SUPPORT VECTOR FOR CIFAR-10 (TEN LABELS).

training data	data	accuracy(%)	time(sec)
(A) all data	50,000	77.560	2116.688
(B-1) support vector(one-against-one)	47,870	77.420	2062.906
(B-2) support vector(one-against-all)	47,799	76.970	2060.405
(C-1) random	47,870	76.580	2052.179
(C-2) random	47,799	76.290	2057.681

fication accuracy. However, because C_GAUSS is a relatively simple dataset, randomly extracted data does not either. This can be visually confirmed as in Fig. 7. Although the hyperplanes trained with support vectors or randomly extracted data are different from each other and from the hyperplane trained with all training data, each hyperplane properly classifies dataset. R_PLANE also shows similar results to C_GAUSS. In R_GAUSS, using support vectors archieves better accuracy than other two cases: even better than the result using all training data. On the other hand, using randomly extracted data is unstable and sometimes severely deteriorates regression accuracy as shown in Fig. 8. Also, in C_XOR, the proposed method does not cause any accuracy loss. On the other hand, using randomly extracted data shows unstable results again as in Fig. 9. This also can be confirmed in C_SPIRAL and C_CIRCLE.

IV. EVALUATION

This section applies the proposed method to a DNN and a massive image dataset as mentioned above, which are ResNet 18 [5] and CIFAR-10 [9].

CIFAR-10 is an image dataset with 60,000 images: 50,000 training data and 10,000 test data. As shown in Fig. 10, each datum consists of a pair of a 32×32 color image and a target label. CIFAR-10, as can be inferred from its name, includes ten classes of labels: airplane, car, bird, cat, deer, dog, frog, horse, ship, and truck. We evaluate the proposed method under two circumstances: classification against five classes and all ten classes. The five classes used for the former classification are airplane, bird, deer, frog, and ship. In both cases, the number of training epoch is 100.

In multiclass classification, there are two strategies for support vectors derivation; using one-against-one classifiers and using one-against-all classifiers [10]. In the one-against-one classifier, ${}_N C_2$ classifiers are used to classify N classes from class C_1 to class C_N. One-against-all classifiers use N classifiers that solve the binary classification problem.

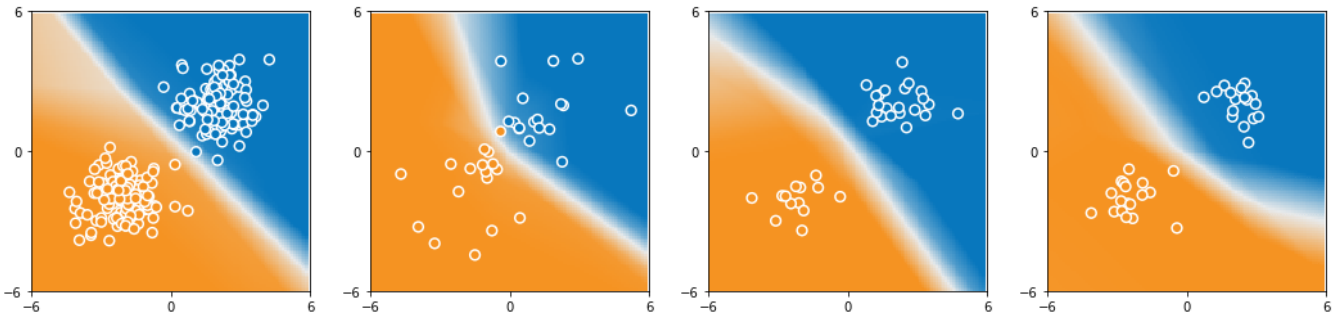
In the evaluation, we confirm the relationship between classification accuracy and learning time for three cases (A) when using all training data, (B-1) when using only support vectors in a one-against-one classifier and (B-2) when using only support vectors in a one-against-all classifier. TABLE IV shows the experimental results for five classes. We can see that the proposed method using support vector reduces the calculation time of ResNet while achieving the same classification accuracy as using all the training data. For example, focusing on the calculation accuracy, the calculation accuracy of (A) is 78.300%, while the calculation accuracies of (B-1) and (B-2) are 78.580% and 78.340%, respectively. As for the calculation time, (A) requires 1058.243 seconds for learning, while (B-1) require 961.395 seconds and (B-2) requires 957.629 seconds. In other words, using only the support vector reduced the calculation time by 9.16% in (B-1) and by 9.50% in (B-2). TABLE V shows the experimental results for ten classes. The accuracy of (B-1) is almost the same as (A) with less calculation time. On the other hand, the accuracy of (B-2) degrades. However, the accuracy is still higher than (C-1) and (C-2). According to TABLE V the calculation time was reduced by 2.54% in (B-1) and by 2.66% in (B-2).

The calculation time in TABLE IV and TABLE V does not include the SVM computation. Thus, the calculation time of the entire learning is longer than that in TABLE IV and TABLE V. However, this increase is not crucial in this research since it is one-time effort and the reduced data set can be repeatedly used. The calculation time reduced in this research is only 9% and 2%, but this reduction is exploited every time the data set is used for training. Thus, the proposed method reduces the number of data required for training by using only the support vectors, which contributes to the calculation time reduction.

Looking at the number of training data required by the conventional method, (B-1) and (B-2), according to TABLE IV, the number of data is reduced by 11.86% in (B-1) and 12% in (B-2) at five labels, respectively. Similarly, according to TABLE V, the reduction rate of the number of training data at ten labels is 4.26% and 4.40%. Similar to the reduction rate of calculation time, the reduction rates of the number of training data at five labels are higher than at ten labels.

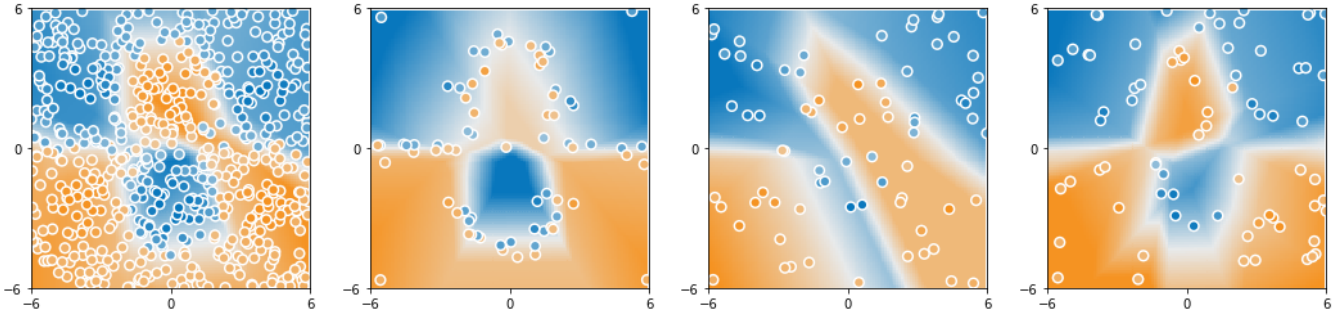
We discuss the reason why the data reduction becomes less significant as the number of labels increase. The preliminary experiments in Section III-B reduced the original data by 86 to 88% when determining the support vectors. However, when the support vectors were extracted from the training data of CIFAR-10, only 12% reduction is obtained in five label case.

The cause of this problem is that classifying color images is much more complicated than that of the preliminary experiments. In the case of a relatively simple data set as treated in the preliminary experiment, no deterioration in accuracy was observed even if the number of data extracted as support vectors was 12 to 14%. However, in the case of relatively complex data sets such as CIFAR-10, as there are many data responsible for the formation of the classification plane, the number of training data extracted as support vectors is about 88%, which is the majority of the whole. Compared with the results of the preliminary experiments, although it did not reduce the training data as expected, the training data could be reliably reduced. Moreover, since the accuracy has hardly been



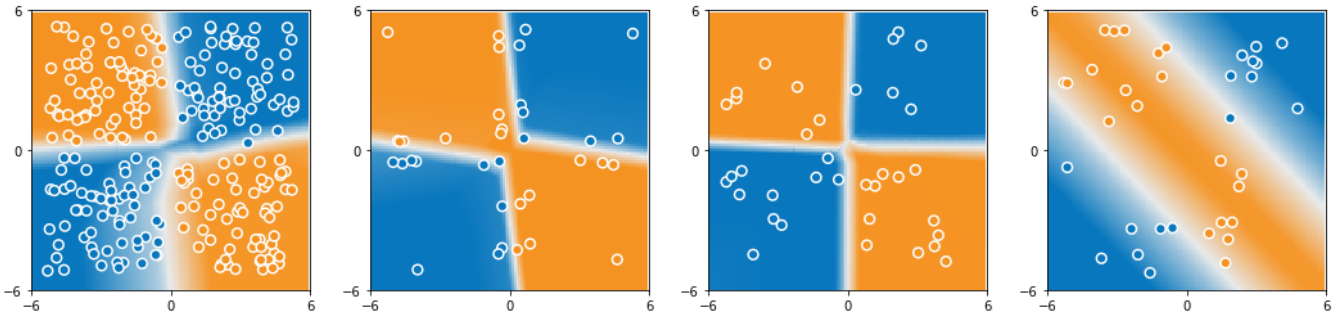
(A) all training data (B) support vector (C) random 1 (D) random 2

Fig. 7. Evaluation in the dataset C_GAUSS.



(A) all training data (B) support vector (C) random 1 (D) random 2

Fig. 8. Evaluation in the dataset R_GAUSS.



(A) all training data (B) support vector (C) random 1 (D) random 2

Fig. 9. Evaluation in the dataset C_XOR.

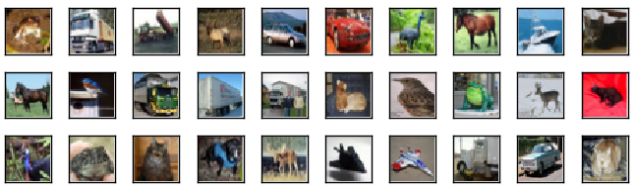


Fig. 10. A part of CIFAR-10 dataset.

deteriorated, it is possible to improve the trade-off between the number of training data and the classification accuracy and to reduce the training data of CIFAR-10 without causing the accuracy deterioration by using the support vector. Compared with the results of the random sampling data performed as a control experiment, the accuracy is also improved, although it is about 1%. Also, the time taken for learning decreased with the reduction of training data, and it can be said that this was also effective. There were more data extracted as support vectors for ten labels than for five labels, and about 95% was extracted. Although only 5%, this was also able to reduce training data. As in the case of five labels, almost no degradation in accuracy occurs, so it can be said that the effect of the support vectors has been confirmed.

V. Conclusions

This paper proposed and evaluated a training data reduction method using support vectors. Important difference from existing studies is that this research verified the feasibility of the training process using support vectors with a ResNet and a CIFAR-10 dataset. The preliminary experiment using simple 2-D datasets confirmed that training data could be reduced by 86 to 88% without accuracy loss. Based on this result, evaluation is conducted under more challenging setup with ResNet 18 and CIFAR-10. The evaluation result shows that training data could be reduced by about 12% in the five classes classification and about 5% in the ten classes classification with ignorable accuracy loss. With the reduction of this data, the learning time of ResNet could also reduced by 9% in the five classes classification and 2% in the ten classes classification.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP19H04079.

References

[1] J. Deng, W. Dong, R. Socher, L. J. Li, , K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[4] D. Decoste and B. Schölkopf, "Training invariant support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 161–190, Jan 2002.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 770–778.

[6] X. Nguyen, L. Huang, and A. D. Joseph, "Support vector machines, data reduction, and approximate kernel matrices," in *Proceedings of Machine Learning and Knowledge Discovery in Databases*, Sep. 2008, pp. 137–153.

[7] K. Dahiya and A. Sharma, "Reducing neural network training data using support vectors," in *Proceedings of Recent Advances in Engineering and Computational Sciences*, Mar. 2014, pp. 1–4.

[8] X.-X. Niu and C. Y. Suen, "A novel hybrid CNN-SVM classifier for recognizing handwritten digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318–1325, Apr. 2012.

[9] A. Krizhevsky, "Convolutional deep belief networks on CIFAR-10," 2010, unpublished.

[10] J. Weston and C. Watkins, "Multi-class support vector machines," Royal Holloway University of London, Tech. Rep., May 1998.