Does End-to-End Trained Deep Model Always Perform Better than Non-End-to-End Counterpart?

Ikuro Sato[1,2]*, Guoqing Liu*[1]*, Kohta Ishikawa*[1]*, Teppei Suzuki*[1]*, and Masayuki Tanaka*[2]
[1] *Denso IT Laboratory, Inc., Japan*
[2] *Tokyo Institute of Technology, Japan*

Abstract

It has been rigorously demonstrated that an end-to-end (E2E) differentiable formulation of a deep neural network can turn a complex recognition problem into a unified optimization task that can be solved by some gradient descent method. Although E2E network optimization yields a powerful fitting ability, the joint optimization of layers is known to potentially bring situations where layers co-adapt one to the other in a complex way that harms generalization ability. This work aims to numerically evaluate the generalization ability of a particular non-E2E network optimization approach known as FOCA (Feature-extractor Optimization through Classifier Anonymization) that helps to avoid complex co-adaptation, with careful hyperparameter tuning. In this report, we present intriguing empirical results where the non-E2E trained models consistently outperform the corresponding E2E trained models on three image-classification datasets. We further show that E2E network fine-tuning, applied after the feature-extractor optimization by FOCA and the following classifier optimization with the fixed feature extractor, indeed gives no improvement on the test accuracy. The source code is available at `https://github.com/DensoITLab/FOCA-v1`*.*

Introduction

End-to-End (often abbreviated as E2E) differentiable formulations using deep neural networks for visual recognition problems have been gaining much attention in the research community. The popularity is largely attributed to the flexibility in formulation to turn sequential sub-tasks into a unified optimization task that can be solved by some sort of gradient descent solvers. One of the best describing examples of replacing intermediate processings by learnable layers is convolutional neural network (see pioneering work, [29, 27]). It replaces a combination of local-feature extractor [13, 32, 4, 49, 8, 2], region pooling [44, 12, 37, 24], and classifier [48, 6, 15, 21, 43, 7]. Some more recent examples include: Spatial Transformer Network [23] replacing spatial-coordinate preprocessing [18] and classification, Faster RCNN [41] replacing region proposal [47] and classification, PointNet [39] replacing 3D-point voxelization and classification [33, 51, 40], OpenPose [10, 9] replacing a framework of person detection and landmark localization [38, 46, 36].

Besides a great success of E2E network optimization approaches, there has been much effort to develop non-E2E network optimization methods. One of the important research goals here is to obtain generic data representations that bring high performance in the task of interest by pre-training network on large-scale dataset(s). Unsupervised representation learning exemplifies the non-E2E network optimization approach, where in most cases the network is first trained unsupervisedly on a large dataset and an add-on classifier is then trained with a different objective on either the same dataset or a different target dataset [5, 35, 20, 3, 17, 11]. Domain adaptation and transfer learning [57, 54, 31, 50, 25, 45, 26] also belong to the non-E2E training category as a whole. Their aim is to extend the dataset to more than one in a sequential fashion, where one expects that the supervised training on the first dataset provides feature extractors beneficial to the objective for the second dataset.

A natural question is, *Which is better, E2E or non-E2E?* It is widely believed that an E2E network optimization generally gives a better performance. Minimizing a unified loss function of the whole network parameters likely yields better *fitting ability*; however, it does not necessarily yield better *generalization ability*. E2E network optimization may even bring situations where layers co-adapt one to the other in a complex way that harms generalization ability [19, 53]. We aim to find solid cases where a non-E2E training method consistently outperforms an E2E counterpart under fair settings. In this work, we employ a particular non-E2E optimization method known as FOCA: Feature-extractor Optimization through Classifier Anonymization [42]. As overviewed in the next section, FOCA is free from co-adaptation between feature extractor and classifier, since the feature extractor is not optimized to a particular strong classifier due to the classifier anonymization technique. In the original work [42], the model performance after FOCA was not particularly high; however, we show that it can be significantly improved by carefully reviewing optimization settings. In this work, we focus on problems of image classification without using external datasets.

Our contributions are summarized as follows.

1. We empirically confirmed that the model performance of FOCA can be significantly improved compared to the original work [42] after careful hyperparameter tuning, and we show solid evidences where FOCA, a non-E2E training method, outperforms strong E2E counterparts in image classification tasks.
2. We found a counter-intuitive phenomenon, where E2E network fine-tuning, after optimizing a feature extractor by FOCA and optimizing the following classifier with the fixed feature extractor, gives no improvement in test performance.

Feature-extractor Optimization through Classifier Anonymization (FOCA): a Review

In this section, we briefly overview the motivation, formulation, optimization strategy, and feature property of FOCA [42] as well as the limitation of work of Sato, *et al*. [42].

The motivation of FOCA is to break co-adaptation between a feature extractor part and a classifier part of a deep neural network. An E2E training of feature extractor and classifier can potentially bring situations where both are tied in a very special way, known as co-adaptation [19, 53], so that the feature distribution adapts to particular decision boundaries and vice versa. This too-specific tiedness likely brings negative effects when transferring the feature extractor after separating the network into two blocks [53]. The idea of FOCA is to train only the feature extractor supervisedly with random weak classifiers generated at every iteration or at every small set of iterations. This *classifier anonymization* technique prevents the feature extractor from adapting to a specific strong classifier. Then, after the feature extractor is obtained by FOCA, a final classifier is optimized with the feature extractor fixed. Apparently, the network is not E2E optimized.

Let x be an input data, $F_\phi(\cdot)$ be a feature extractor with parameter set ϕ, $C_\theta(\cdot)$ be a classifier with parameter set θ, and $L(\cdot;t)$ be a sample-wise loss function with a target value t. In FOCA, the primary optimization, defined as the optimization of the feature extractor, is given by

$$\phi^\star = \arg\min_\phi \sum_{(x,t)\in D} \sum_{\theta\in\Theta_\phi} L(C_\theta(F_\phi(x));t), \tag{1}$$

where Θ_ϕ is a set of weak classifiers[1], and D is the training dataset. Sato, *et al.* [42] introduced a particular way of generating weak classifiers, by first sampling a small batch of data and then computing a strong classifier with respect to that batch. They regard that classifiers generated in this way generally work weakly to the entire dataset (This is why they call it as weak classifier).

Once the feature extractor is obtained, a new classifier is built with fixed feature extractor as

$$\theta^\star = \arg\min_\theta \sum_{(x,t)\in D} L(C_\theta(F_{\phi^\star}(x));t). \tag{2}$$

We call the process to obtain $\theta^\star$ as the secondary optimization.

The optimization strategy in Sato, *et al.* [42] is to sample a minibatch from D, generate a single weak classifier θ, and update feature-extractor ϕ with stochastic gradients at each iteration.

Sato, *et al.* showed theoretically under certain conditions that *features of the same class form a point-like distribution while features of different classes do not* [42]. It is considered as an implicit optimality caused by the classifier anonymization, because the objective itself does not contain any of metric-learning type regularization terms at least explicitly.

The empirical results shown by Sato, *et al.* [42] support the point-like behavior; however, they do not discuss the potential of FOCA as a regularizer. Their experiments used neither modern, strong deep models nor a network normalization technique such as Batch Normalization (BN) [22]. We find that use of BN and global image features instead of local features greatly improve the performance of FOCA, as we discuss in the next section.

Experiments

The main purpose of the experiments is to show test performance comparisons between FOCA, which is a non-E2E network training method, and its E2E counterpart under fair experimental settings. We first show how the FOCA performance itself can be improved from the results presented by Sato, *et al.* [42]. Next, we show main results of the test performance comparisons with strong baseline methods. We then investigate whether the performance of the network trained by FOCA can be further improved by a following E2E network fine-tuning technique. Finally, we visualize feature distributions through low-dimensional analyses to clarify differences among training methods.

General Settings shared in all experiments. CIFAR-10 [28], CIFAR-100 [28], and Tiny ImageNet [1] datasets are evaluated with Wide ResNet [55, 14] and PyramidNet [16, 52] architectures. Cross entropy loss function is used with softmax output. Nesterov's Accelerated Gradient (NAG) method [34] with the momentum rate of 0.9 is used. Batch Normalization [22] is used. All images are zero-padded by 4 pixels on each side. Images from Tiny ImageNet are first resized to 32×32 spatial size. As data augmentation, random cropping is applied to extract 32×32 region in all cases. Random horizontal mirroring with 50% probability is also used. Unless otherwise indicated, following hyperparameters are adopted. Minibatch size of 128 is used. Weight decay is used with rate of 5e-4 for Wide ResNet architecture and 1e-4 for PyramidNet architecture. For Wide ResNet, the training duration is 600 epochs, and the learning rate is dropped by a factor of 10 at 300, 400, and 500 epochs. For PyramidNet, the learning rate is annealed by the 1800-epoch cosine annealing rule just as in [52]. The size of cutout [14] is 16×16 for CIFAR-10, 8×8 for CIFAR-100, and 8×8 for Tiny ImageNet. Parameters for Random Erasing [58] data augmentation are the ones used in [52]. A Value after ± also indicates one standard deviation calculated from 4 trials.

Settings of FOCA shared in all experiments. A random weak classifier is trained with learning rate of 0.03 and weight decay rate of 0.01. The batch size used in the random weak classifier generation is 100 for CIFAR-10, 1000 for CIFAR-100, and 2000 for Tiny ImageNet. A generated random weak classifier is repeatedly used by 32 times. The secondary classifier optimization with a fixed feature extractor uses a constant learning rate of 0.003 and a weight decay rate of 0.01.

Improvement over Sato, et al. [42]

We first report that test performance of FOCA can be significantly improved from the one reported in Sato, *et al.* [42]. Their emphasis was to demonstrate how the obtained features form simple distributions. Our focus here is to bring out its potential as a *regularizer*. We conducted a series of experiments and figured out that employing Batch Normalization (BN) [22] and use of global feature extractor (instead of local feature extractor) as well as careful hyperparameter tuning can significantly improve the test performance of FOCA.

Table 1 shows test performance improvements over the work of Sato, *et al.* [42]. Adding BN, adding BN and Global Average Pooling (GAP) [30], and adding BN and GAP followed by 2-LP (MLP having 2 weight layers) consistently improve the performance. Though Sato, *et al.* [42], adopted the classifier anonymization technique to *local* features, we think that one better adopts the technique to *global* features to obtain a simple feature distribution. The point-like distribution property is only proved under several conditions, including the use of global fea-

[1] While the original paper defines Θ_ϕ as a probability distribution for weak classifiers, we define it as a *set* in this work without a loss of generality

Table 1. The improvement of test performance of FOCA over the work of Sato, *et al.* [42]. The convolutional part of Wide ResNet [55] is used as the base network of the feature extractor. The classifier part has one weight layer in all cases. CIFAR-10 dataset [28] is used. '2-LP' indicates an MLP having 2 weight layers. G.F. indicates global features.

	Method	Error rate
(A)	simple impl. of FOCA	3.90 ± 0.08%
(B)	(A) + BN [22]	3.19 ± 0.10%
(C)	(B) + G.F. (GAP [30])	2.96 ± 0.02%
(D)	(B) + G.F. (GAP [30] → 2-LP)	**2.63** ± 0.06%

Table 2. Test error rate (%) comparison of FOCA and the E2E counterpart using the Wide ResNet (28-10) architecture [55]. TIN represents Tiny ImageNet.

Method	CIFAR-10	CIFAR-100	TIN
original from [55]	3.89	18.85	N/A
cutout (from [14])	3.08 ± 0.16	18.41 ± 0.27	N/A
cutout (by us)	3.10 ± 0.04	17.99 ± 0.03	37.05 ± 0.25
FOCA w/ cutout	**2.63** ± 0.06	**17.22** ± 0.12	**36.71** ± 0.25

Table 3. Test error rate (%) comparison of FOCA and the E2E counterpart using PyramidNet architecture [16]. R.E. represents Random Erasing [58].

Method	CIFAR-10	CIFAR-100
original from [16]	3.31 ± 0.08	16.35 ± 0.24
shakedrop + R.E. (from [52])	2.31	12.19
FOCA w/ shakedrop + R.E.	**1.76** ± 0.06	**11.82** ± 0.1

tures. Local features are computed from a local part of input image so they are likely to be distinct within a class. Table 1 provides a supportive evidence that one better uses global features when applying the classifier anonymization technique.

Comparison with Strong Baselines

The experimental purpose in this subsection is to evaluate the generalization ability of FOCA against E2E optimization. Below we discuss the experimental settings and results for two major deep architectures.

Wide ResNet [55, 14]: Settings. We basically follow [14] for the training detail. The learning rate of the baseline model is 0.1 for CIFAR-10 and -100, and 0.05 for Tiny ImageNet. The training duration is 200 epochs, and the learning rate is dropped by a factor of 5 at 60, 120, and 160 epochs for the baseline models. We tried a few times longer training with similar learning rate scheduling, but there was no improvement in test accuracy. We pre-examined the baseline models with and without dropout, and indeed the one without dropout slightly performed better. Thus, we decided not to use dropout at all. The baseline architecture has the same convolutional block of Wide ResNet, GAP, and 1-LP, just as [14]. We pre-examined a version of 3-LP after GAP, but it did not improve the baseline accuracy. The initial learning rate for the feature extractor training with FOCA is 0.03 for CIFAR-10, 0.1 for CIFAR-100, and 0.05 for Tiny ImageNet. A random weak classifier is updated 64 times for CIFAR-10, 128 times for CIFAR-100, and 256 for Tiny ImageNet. The training duration for the secondary optimization is 1 epoch for CIFAR-10, 15 epochs for CIFAR-100 and Tiny ImageNet.

Wide ResNet [55, 14]: Results. Table 2 shows test error rates of the Wide ResNet models. We put mean values and standard deviations (after ± symbols) calculated from 4 independent trials in our results. FOCA consistently outperforms the strong baseline results for CIFAR-10, CIFAR-100 and Tiny ImageNet. The gaps between the mean test error rates are clearly wider than the corresponding standard deviations. We would like to emphasize again that the feature extractor and the classifier in FOCA are *not* jointly optimized –the classifier is trained with the *fixed* feature extractor.

PyramidNet [16, 52]: Settings. The initial learning rate of for the feature extractor training with FOCA is 0.1 for CIFAR-10 and CIFAR-100. A random weak classifier is updated 64 times for CIFAR-10 and 128 times for CIFAR-100. The training duration for the secondary optimization is 1 epoch for CIFAR-10 and 10 epochs for CIFAR-100.

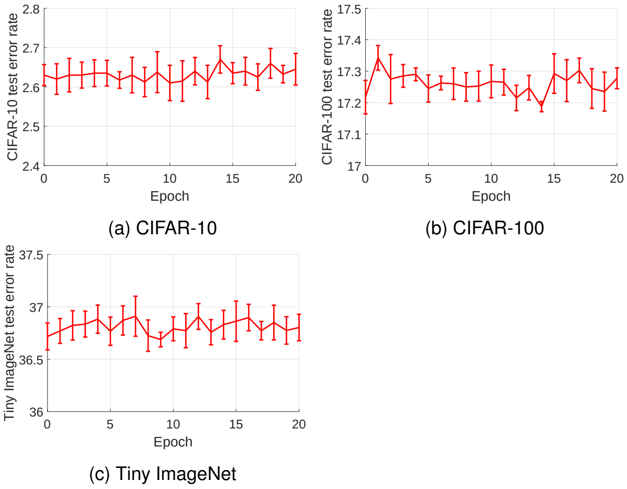
PyramidNet [16, 52]: Results. Table 3 shows test error rates of the PyramidNet model. Again, FOCA outperforms the strong baseline results for CIFAR-10 and CIFAR-100 tested here, and the gaps between the mean test accuracies are clearly wider than the corresponding standard deviations. Combining FOCA with other type of data augmentation methods such as mixup [56] is interesting, thus we leave it for future work.

Effects of Fine-Tuning after FOCA

The experimental purpose in this subsection is to examine the effects of fine-tuning after non-E2E training. All the experiments shown so far do not utilize a technique of network fine-tuning at all. That means, after a feature extractor is generated by FOCA, the following classifier is optimized with frozen features only. Therefore, it is not likely that the feature extractor is optimal for the classifier on the training set. In this subsection, we show empirical results in order to answer the question: *Does E2E fine-tuning after independently optimizing feature extractor with FOCA and classifier improve the test performance?*

Procedure. We take our non-E2E trained models that have the Wide ResNet base architecture and, for each model we fine-tune the whole network parameters in an E2E manner. Learning rates used in the fine-tuning stage are set as the final learning rates used in the primary optimization stages. Decay rate of 5e-4 is used.

Results. Figure 1 shows the test error rate curves for E2E fine-tuning after pre-trained with FOCA. In all cases, the mean values of the test error rates are fluctuated well within the error bars, *i.e.*, the fine-tuning gives no improvements. The possible reason why the E2E fine-tuning after FOCA gives no improvement on the generalization would be that the E2E fine-tuning brings co-adaptation between the feature extractor and classifier at some level. That is, the feature distribution becomes optimized to specific decision boundaries and vice versa, lacking robustness against small change in feature distribution.

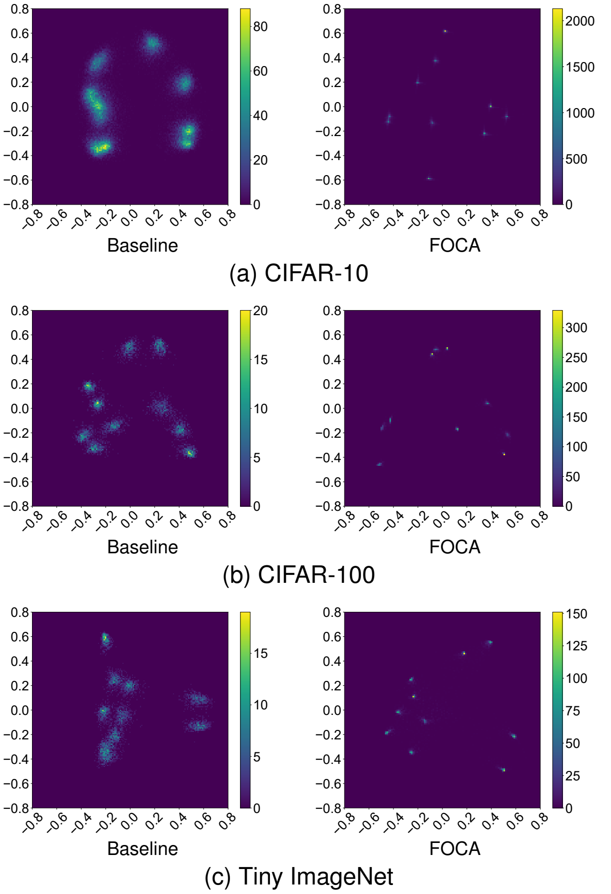


(a) CIFAR-10

(b) CIFAR-100

(c) Tiny ImageNet

Figure 1. *Test error rate curves (in %) during fine-tuning. The networks are pre-trained in a non-E2E fashion by FOCA and then by linear classifier optimization. The sizes of the error bars indicate one standard-deviations calculated from 4 trials. In each case, fine-tuning gives no improvement.*



(a) CIFAR-10

(b) CIFAR-100

(c) Tiny ImageNet

Figure 2. *Histograms of data points in the feature spaces after dimensionality reduction. (b) and (c) show 10 randomly selected classes for visibility. Point-like distributions are clearly observed for FOCA. Distributions are more spread for E2E.*

Visualization of Feature Distribution

In this subsection, we investigate the forms of feature distributions to see if they can characterize generalizability.

Procedure. We take the Wide ResNet models already discussed and compute features of respective training sets. Since feature vectors have large norm discrepancies in a given dataset, we first normalize each feature vector so that the L2 norm becomes one. We then apply PCA to features within a given (sub-)dataset to reduce the dimension into two for visualization. The square grid is prepared for population counting to generate a histogram. The grid size is given by the square root of the maximum eigenvalue given by PCA, divided by 30.

Results. Figure 2 shows histograms of data points in the feature spaces after the dimensionality reduction. FOCA features very clearly exhibit class-wise point-like distribution as discussed in [42], even when various regularizing techniques (e.g., cutout [14], shakedrop [52], Random Erasing [58], BN [22]) are adopted. For example, the right figure of Fig. 2 (a) has 10 strong peaks, each of which consists of data points of an identical class. In contrast, features of the E2E counterparts are much more spread, possibly indicating that point-like feature distribution helps enhancing generalization ability.

Conclusion

To answer the question: *Which is better, E2E or non-E2E?*, we investigate performance gaps between them using strong deep models in image classification problems. We found rich examples where the non-E2E training method known as FOCA [42] outperforms E2E counterparts. Co-adaptation breaking functionality of FOCA likely explains the performance gains.

References

[1] Tiny ImageNet Visual Recognition Challenge. `https://tinyimagenet.herokuapp.com`.

[2] Mitsuru Ambai and Yuichi Yoshida. Card: Compact and real-time descriptors. In *2011 International Conference on Computer Vision*, pages 97–104. IEEE, 2011.

[3] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15509–15519, 2019.

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[5] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

[6] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[7] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[8] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[9] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using

part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

[10] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

[12] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[13] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[14] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[15] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.

[16] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5927–5935, 2017.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[18] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International conference on artificial neural networks*, pages 44–51. Springer, 2011.

[19] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[20] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[21] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.

[23] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[24] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3304–3311. IEEE, 2010.

[25] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *European Conference on Computer Vision*, pages 67–84. Springer, 2016.

[26] Alexander I Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Large scale learning of general visual representations for transfer. *ArXiv*, abs/1912.11370, 2019.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012.

[28] Alex Krizhevskyf and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 4, 2009.

[29] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[30] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[31] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4013–4022, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[32] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[33] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015.

[34] Yurii Nesterov. Introductory lectures on convex programming volume i: Basic course. *Lecture notes*, 3(4):5, 1998.

[35] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[36] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017.

[37] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

[38] Leonid Pishchulin, Arjun Jain, Mykhaylo Andriluka, Thorsten Thormählen, and Bernt Schiele. Articulated people detection and pose estimation: Reshaping the future. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3178–3185. IEEE, 2012.

[39] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[40] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016.

[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[42] Ikuro Sato, Kohta Ishikawa, Guoqing Liu, and Masayuki Tanaka. Breaking inter-layer co-adaptation by classifier anonymization. In

Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5619–5627, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[43] Robert E Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee, et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998.

[44] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE, 2003.

[45] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[46] Min Sun and Silvio Savarese. Articulated part-based model for joint object detection and pose estimation. In *2011 International Conference on Computer Vision*, pages 723–730. IEEE, 2011.

[47] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[48] Vladimir Vapnik. Pattern recognition using generalized portrait method. *Automation and remote control*, 24:774–780, 1963.

[49] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In *2009 IEEE 12th international conference on computer vision*, pages 32–39. IEEE, 2009.

[50] Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6872–6881, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[51] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[52] Yoshihiro Yamada, Masakazu Iwamura, and Koichi Kise. Shakedrop regularization. In *International Conference on Learning Representations (ICLR) Workshop*, 2018.

[53] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? 2014.

[54] Kaichao You, Ximei Wang, Mingsheng Long, and Michael Jordan. Towards accurate model selection in deep unsupervised domain adaptation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7124–7133, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[55] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016.

[56] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[57] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7523–7532, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[58] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020.

Author Biography

Ikuro Sato *received his BS in physics and in mathematics from Ohio University (2000) and his Ph.D. in nuclear physics from University of Maryland (2005). After spending postdoctoral research period at Lawrence Berkeley National Laboratory, he joined Denso IT Laboratory, Japan. He has been cross appointed to Tokyo Institute of Technology as a specially appointed associate professor and Denso IT Laboratory from 2020.*

Guoqing Liu *received his master's degree from Waseda University in 2010. He joined Meitec Corporation in 2013.*

Kohta Ishikawa *received his BS in physics from Hokkaido University (2004) and his MS in condensed matter physics from Tokyo Institute of Technology (2006). He joined Denso IT Laboratory, Japan in 2012. He was a visiting scholar with Neuroscience Institute, UC Berkeley, CA, USA.*

Teppei Suzuki *received his bachelor's and master's degrees in engineering from Keio University in 2016 and 2018. He joined Denso IT Laboratory in 2018. Since 2019, he has been a Ph.D. student at Keio University.*

Masayuki Tanaka *received his bachelor's and master's degrees in control engineering and Ph.D. degree from Tokyo Institute of Technology in 1998, 2000, and 2003. He joined Agilent Technology in 2003. He was a Research Scientist at Tokyo Institute of Technology from 2004 to 2008. Since 2008, he has been an Associate Professor at the Graduate School of Science and Engineering, Tokyo Institute of Technology. He was a Visiting Scholar with Department of Psychology, Stanford University, CA, USA.*