Significance of Softmax-Based Features in Comparison to Distance Metric Learning-Based Features

Shota Horiguchi, *Member, IEEE*, Daiki Ikami, *Student Member, IEEE*, and Kiyoharu Aizawa, *Fellow, IEEE*

Abstract—End-to-end distance metric learning (DML) has been applied to obtain features useful in many computer vision tasks. However, these DML studies have not provided equitable comparisons between features extracted from DML-based networks and softmax-based networks. In this paper, we present objective comparisons between these two approaches under the same network architecture.

Index Terms—Deep learning, distance metric learning, classification, retrieval

1 Introduction

Recent developments in deep convolutional neural networks have made it possible to classify many classes of images with high accuracy. It has also been shown that such classification networks work well as feature extractors. Features extracted from classification networks show excellent performance in image classification [1], detection, and retrieval [2], [3], even when they have been trained to classify 1,000 classes of the ImageNet dataset [4]. It has also been shown that fine-tuning for target domains further improves the features' performance [5], [6].

On the other hand, distance metric learning (DML) approaches have recently attracted considerable attention. These obtain a feature space in which distance corresponds to class similarity; it is not a byproduct of the classification network. End-to-end distance metric learning is a typical approach to constructing a feature extractor using convolutional neural networks and has been the focus of numerous studies [7], [8], [9], [10], [11].

However, there have been no experiments comparing softmax-based features with DML-based features under the same network architecture or with adequate fine-tuning. An analysis providing a true comparison of DML features and softmax-based features is long overdue.

Fig. 1 depicts the feature vectors extracted from a softmax-based classification network and a metric learning-based network. We used LeNet architecture for both networks, and trained on the MNIST dataset [12]. For DML, we used the contrastive loss function [13] to map images in two-dimensional space. For softmax-based classification, we added a two- or three-dimensional fully connected layer before the output layer for visualization. DML succeeds in learning feature embedding (Fig. 1a). Softmax-based classification networks can also achieve a result very similar to that obtained by DML—Images are located near one another if they belong to the same class and far apart otherwise (Figs. 1b and 1c).

• *The authors are with the Department of Information and Communication Engineering, University of Tokyo, Bunkyo, Tokyo 133-8656, Japan. E-mail: {horiguchi, ikami, aizawa}@halt.u-okyo.ac.jp.*

Manuscript received 14 Oct. 2017; revised 28 Feb. 2019; accepted 2 Apr. 2019. Date of publication 15 Apr. 2019; date of current version 1 Apr. 2020. (Corresponding author: Kiyoharu Aizawa.) Recommended for acceptance by G. Shakhnarovich. Digital Object Identifier no. 10.1109/TPAMI.2019.2911075

Our contributions in this paper are as follows:

- We show methods to exploit the ability of deep features extracted from softmax-based networks, such as normalization and proper dimensionality reduction. They are technically not novel, but they must be used for fair comparison between the image representations.
- We demonstrate that deep features extracted from softmax-based classification networks show competitive, or better results on clustering and retrieval tasks comparing to those from state-of-the-art DML-based networks [9], [10], [11] on the Caltech UCSD Birds 200-2011 dataset and the Stanford Cars 196 dataset.
- We show how the clustering and retrieval performances of softmax-based features and DML features change according to the size of the dataset. DML features show competitive or better performance in the Stanford Online Product dataset which consists of very small number of samples per class.
- We show that L2 normalization of softmax-based features is a powerful way to improve their performance. Even though we introduce probability invariant shift, which removes effects of softmax ambiguity and null space ambiguity, L2 normalization still works better.

In order to align the condition of the network architecture, we restrict the network architecture to GoogLeNet [14] which has been used in state-of-the-art of DML studies [9], [10], [11].

2 Background

2.1 Previous Work

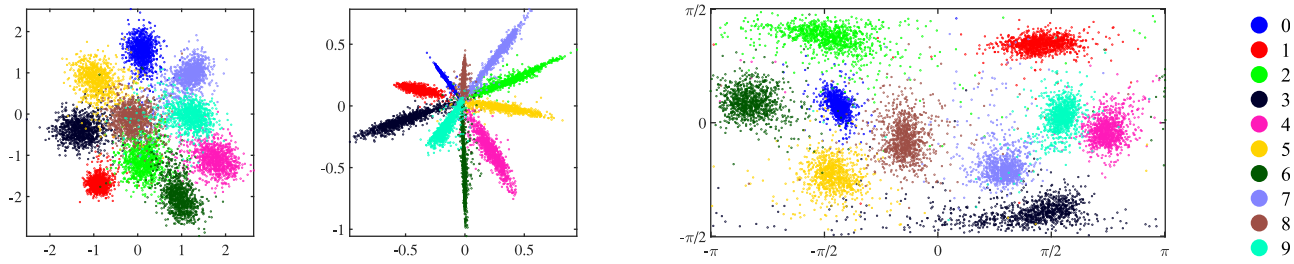
2.1.1 Softmax-Based Classification and Repurposing of the Classifier as a Feature Extractor

Convolutional neural networks have demonstrated great potential for highly accurate image recognition [14], [15], [16], [17]. It has been shown that features extracted from classification networks can be repurposed as a good feature representation for novel tasks [1], [2], [18] even if the network was trained on ImageNet [4]. For obtaining better feature representations, fine-tuning is also effective [6].

2.1.2 Deep Distance Metric Learning

Distance metric learning (DML), which learns a distance metric, has been widely studied [18], [19], [20], [21]. Recent studies have focused on end-to-end deep distance metric learning [7], [8], [9], [10], [11], [12]. However, in most studies comparisons of end-to-end DML with features extracted from classification networks have not been performed using architectures and conditions suited to enable a true comparison of performance.

Bell and Bala [7] compared classification networks and siamese networks, but they used coarse class labels for classification networks and fine labels for siamese networks; thus, it was left unclear whether siamese networks are better for feature-embedding learning than classification networks. Schroff et al. [8] used triplet loss for deep metric learning in their FaceNet, which showed performance that was state-of-the-art at the time, but their network was deeper than that of the previous method (Taigman et al. [23]); thus, triplet loss might not have been the only reason for the performance improvement, and the contribution from adopting triplet loss remains uncertain. Song et al. [9] used lifted structured feature embedding; however, they only compared their method with a softmax-based classification network pretrained on ImageNet (Russakovsky et al., [4]) and did not compare it with a fine-tuned network. Sohn [10], and Song et al. [11] also compared their methods to lifted structured feature embedding, thus the comparisons with softmax-based features have not been shown.



(a) Siamese ($dim = 2$) (b) Softmax ($dim = 2$) (c) Softmax ($dim = 3$) + L2 normalization

Fig. 1. Depiction of MNIST dataset. (a) Two-dimensional features obtained by siamese network. (b) Two-dimensional features extracted from softmax-based classifier; these features are well separated by angle but not by euclidean norm. (c) Three-dimensional features extracted from softmax-based classifier; we normalized these to have unit L2 norm and depict them in an azimuth–elevation coordinate system. The three-dimensional features are well separated by their classes.

2.2 Differences Between Softmax-Based Classification and Metric Learning

For classification, the softmax function (Eq. (1)) is typically used

$$p_c = \frac{\exp(u_c)}{\sum_{i=1}^{C} \exp(u_i)}, \tag{1}$$

where p_c denotes the probability that the vector $\mathbf{u}$ belongs to the class c. The loss of the softmax function is defined by the cross-entropy

$$E = -\sum_{c=1}^{C} q_c \log p_c, \tag{2}$$

where $\mathbf{q}$ is a one-hot encoding of the correct class of $\mathbf{u}$. To minimize the cross-entropy loss, networks are trained to make the output vector $\mathbf{u}$ close to its corresponding one-hot vector. It is important to note that the target vectors (the correct outputs of the network) are fixed during the entire training (Fig. 2).

On the other hand, DML methods use distance between samples. They do not use the values of the labels; rather, they ascertain whether the labels are the same between target samples. For example, contrastive loss [13] considers the distance between a pair of samples. Recent studies [8], [9], [10], [11] use pairwise distances between three or more images at the same time for fast convergence and efficient calculation. However, these methods have some drawbacks. For DML, in contrast to optimization of the softmax cross-entropy loss, the optimization targets are not always consistent during training even if all possible distances within the mini-batch are considered. Thus, the DML optimization converges slowly and is not stable.

3 Methods

3.1 Dimensionality Reduction Layer

One of DML's strength in using fine-tuning is the flexibility of its output dimensionality by a final fully connected layer. When using features of a mid-layer of a softmax classification network, on the other hand, the dimensionality of the features is fixed. Some existing methods [6] use PCA or discriminative dimensionality reduction to reduce the number of feature dimensions. In our experiment, we evaluated three methods for changing the feature dimensionality. Following conventional PCA approaches, we extracted features from a 1,024-dimensional pool5 layer of GoogLeNet [14] (Fig. 3a) and applied PCA to reduce the dimensionality. As a comparison, we also tried random projection for dimensionality reduction via orthogonal projection matrix. In a contrasting approach, we made use of a fully connected layer—we added a fully connected layer having the required number of neurons just before the output layer (FCR1, Fig. 3b). We also investigated a third approach in which a fully connected layer is added followed by a dropout layer (FCR2, Fig. 3c).

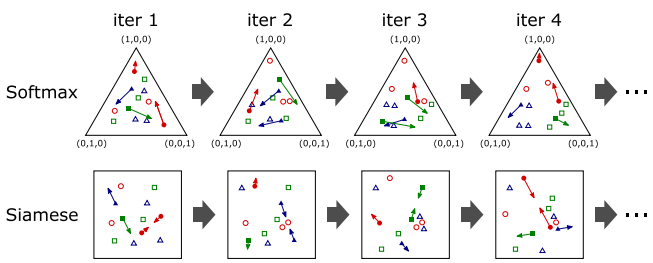
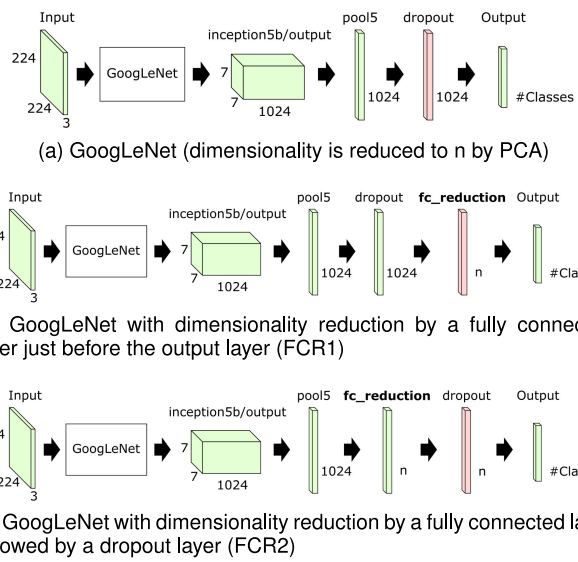


Fig. 2. Illustration of learning processes for softmax-based classification network and siamese-based DML network. For softmax, the gradient is defined by the distance between a sample and a fixed one-hot vector; for siamese by the distance between samples.

3.2 Normalization

In this study, all the features extracted from the classification networks are from the last layer before the last output layer. The outputs are normalized by the softmax function and then evaluated by the cross-entropy loss function in the networks. The output vector $\mathbf{p} = (p_i)$ is given by $\mathrm{softmax}(\mathbf{y})$. For an arbitrary constant c, $\mathrm{softmax}(\mathbf{y})$ equals to $\mathrm{softmax}(\mathbf{y} + c\mathbf{1})$. The features $\mathbf{x}$ we extracted from the networks are given as $\mathbf{y} = W\mathbf{x} + \mathbf{b}$, where W and $\mathbf{b}$ are from the linear projection matrix and the bias, respectively. As pointed out, the vector $\mathbf{y}$ has an ambiguity in the softmax function, thus $\mathbf{x}$ should be normalized for the use of deep features.

In this paper, we show that L2 normalization is empirically effective. Some studies used L2 normalization for deep features



(a) GoogLeNet (dimensionality is reduced to n by PCA)

(b) GoogLeNet with dimensionality reduction by a fully connected layer just before the output layer (FCR1)

(c) GoogLeNet with dimensionality reduction by a fully connected layer followed by a dropout layer (FCR2)

Fig. 3. GoogLeNet [14] architecture we use in this paper. We extracted the features of the red-colored layers. For (a), we applied PCA to reduce the number of feature dimensions. For (b) and (c), the dimensionality is reduced by the fc_reduction layer.

TABLE 1
Properties of Datasets Used in Our Experiments

Dataset	Train	Test	Total
CUB [24]	5,864 100	5,924 100	11,788 200
CAR [25]	8,054 98	8,131 98	16,185 196
OP [9]	59,551 11,318	60,502 11,316	120,053 22,634

Each cell shows the number of images (upper figure) and the number of classes (lower figure).

TABLE 2
CUB: NMI (Clustering) and Recall@K (Retrieval) Scores for the Test Set of the Caltech UCSD Birds 200-2011 (CUB) Dataset

		(clustering)	Recall@K (retrieval)			
	dim	NMI	K = 1	K = 2	K = 4	K = 8
Lifted struct [9]	64	56.5	43.6	56.6	68.6	79.6
	64	(56.0)	(42.7)	(55.0)	(67.2)	(78.1)
N-pair loss [10]	64	57.2	45.4	58.4	69.5	79.5
Clustering loss [11]	64	59.2	48.2	61.4	71.8	81.9
Random projection + L2	64	56.9	47.5	60.1	71.9	81.6
PCA + L2	64	**60.8**	**51.1**	**64.0**	**75.3**	**84.0**
FCR1 + L2	64	59.1	49.0	61.1	72.7	82.3
FCR2 + L2	64	57.4	48.0	60.3	72.2	81.6

TABLE 3
CAR: NMI (Clustering) and Recall@K (Retrieval) Scores for the Test Set of the Stanford Cars 196 (CAR) Dataset

		(clustering)	Recall@K (retrieval)			
	dim	NMI	K = 1	K = 2	K = 4	K = 8
Lifted struct [9]	64	56.9	53.0	65.7	76.0	84.0
	64	(57.1)	(50.5)	(63.6)	(74.9)	(83.6)
N-pair loss [10]	64	57.8	53.9	66.8	77.8	86.4
Clustering loss [11]	64	59.0	58.1	70.6	80.3	87.8
Random projection + L2	64	53.6	63.5	74.4	83.2	89.6
PCA + L2	64	58.3	**69.4**	**80.0**	**87.2**	**92.4**
FCR1 + L2	64	58.7	66.7	77.7	85.2	90.8
FCR2 + L2	64	**60.4**	67.9	78.4	86.1	91.3

TABLE 4
OP: NMI (Clustering) and Recall@K (Retrieval) Scores for the Test Set of the Online Product (OP) Dataset

		(clustering)	Recall@K (retrieval)		
	dim	NMI	K = 1	K = 10	K = 100
Lifted struct [9]	64	88.7	62.5	80.8	91.9
	64	(87.7)	(61.0)	(79.9)	(91.5)
N-pair loss [10]	64	89.4	66.4	83.2	93.0
Clustering loss [11]	64	**89.5**	**67.0**	**83.7**	**93.2**
Random projection + L2	64	85.5	54.3	69.6	81.4
PCA + L2	64	87.5	62.4	78.9	89.7
FCR1 + L2	64	87.7	61.3	78.6	90.1
FCR2 + L2	64	87.9	62.5	79.8	90.8

extracted from softmax-based classification networks [6], [23], whereas many recent studies have used the features without any normalization [9], [15], [26]. $W\mathbf{x}$ and $W\mathbf{x}/|\mathbf{x}|$ do not always result in the same probabilities after the softmax function is applied. Applying L2-normalization for deep features rounds the confidence of predicted results while it keeps the magnitude relationship between probabilities of every classes. However, as Fig. 1b clearly indicates, the distance between features extracted from a softmax-based classifier should be evaluated by cosine similarity, not by the Euclidean distance. In this study, we mainly validated the efficiency of L2 normalization of deep features.

We also considered another way to cope with the ambiguity introduced by the shift invariance of softmax function and null space of W. We define a distance metric that takes softmax invariance and the null space into account, which treats features that result in the same probabilities as equal. We report the experimental results of using the distance metric with probability invariant shift in Section 4.4.

4 Experiments

In this section, we compared the deep features extracted from classification networks to those from state-of-the-art DML-based networks [9], [10], [11]. The GoogLeNet architecture [14] was used for all the methods—thus, the numbers of parameters are the same between DML-based networks and softmax-based features. All the networks were fine-tuned from the weights pretrained on ImageNet [4]. We used the Caffe [27] framework for the implementation.

4.1 Comparisons Between Softmax-Based Features and DML-Based Features

Here, we give our evaluation of clustering and retrieval scores for the state-of-the-art DML methods [9], [10], [11] and for the softmax classification networks. We used the Caltech UCSD Birds 200-2011 (CUB) dataset [24], the Stanford Cars 196 (CAR) dataset [25], and the Stanford Online Products (OP) dataset [9]. For CUB and CAR, we used the first half of the dataset classes for training and the rest for testing. For OP, we used the training–testing class split provided. The dataset properties are shown in Table 1. We emphasize that the class sets used for training and testing were completely different.

For clustering evaluation, we applied k-means clustering 100 times and calculated Normalized Mutual Information (NMI) [28]; the value for k was set to the number of classes in the test set. For retrieval evaluation, we calculated Recall@K [29].

In Tables 2 and 3, we show comparisons of the performance of clustering and retrieval using NMI and Recall@K scores, respectively, for CUB and CAR datasets. We compared the softmax-based features, lifted structure [9], N-pair loss [10] and the clustering loss [11]. The results of the DML methods were quoted from the paper [11]. Regarding the lifted structure [9], the results in the parenthesis correspond to the scores we obtained from running the publicly available code ourselves, which we confirmed were almost the same as those in [11]. As we can see from Tables 2 and 3, softmax-based features outperformed DML features. The softmax-based features all performed well in the two datasets.

In OP dataset shown in Table 4, contrasting to CUB and CAR datasets, DML features outperform softmax-based features. We will make detailed analysis in the subsequent section.

4.2 Detailed Comparisons Between Softmax-Based Features and Lifted Structure Embedding Features

We made detailed comparisons between softmax-based features and lifted structure embedding [9] when changing dimensionalities and size of data. We conducted these experiments using the code available for lifted structure embedding [9].

First, we show how the performance varies when changing the feature dimensionalities. We changed the dimensionalities of softmax-based features via PCA, FCR1 and FCR2, and investigated how the performance of clustering and retrieval varied. We compared them against those of lifted structure embedding of the same dimensionality.

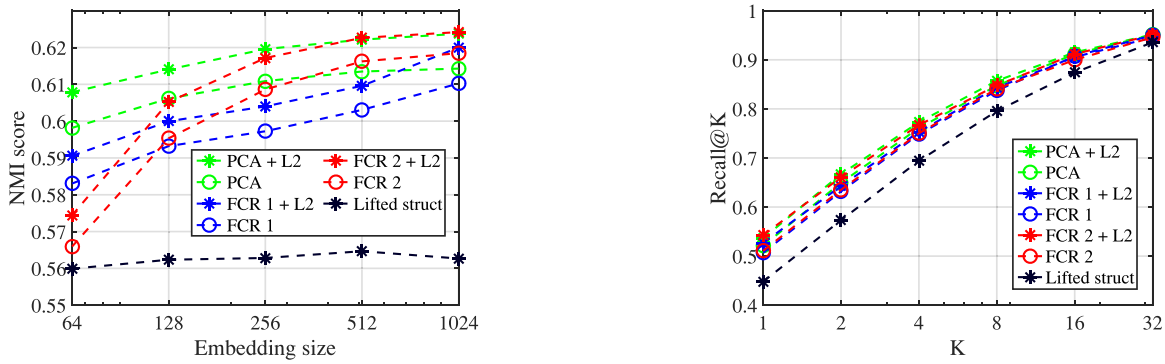


Fig. 4. Comparisons between softmax-based features and lifted structured feature embedding [9] on NMI (clustering) and Recall@K (retrieval) scores for the test set of the Caltech UCSD Birds 200-2011 (CUB) dataset. The dimension of the feature used in the retrieval experiments is 64.

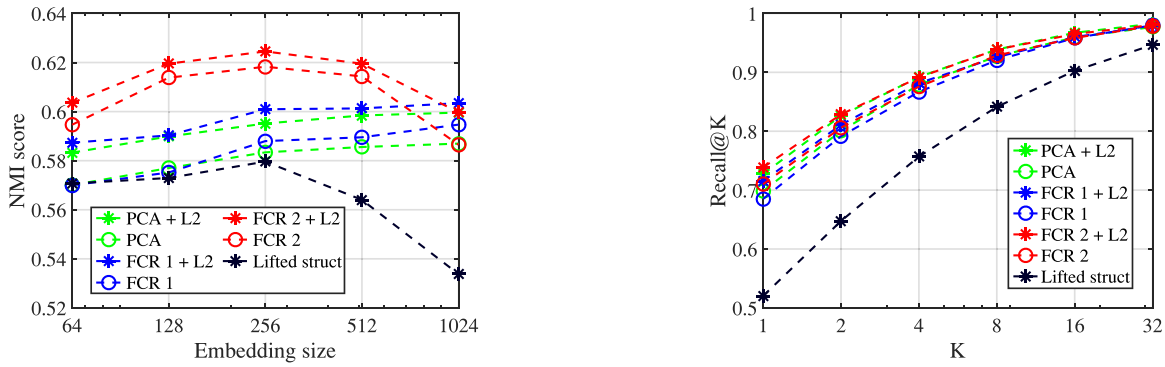


Fig. 5. Comparisons between softmax-based features and lifted structured feature embedding [9] on NMI (clustering) and Recall@K (retrieval) scores for the test set of the Stanford Cars 196 (CAR) dataset. The dimension of the feature used in the retrieval experiments is 64.

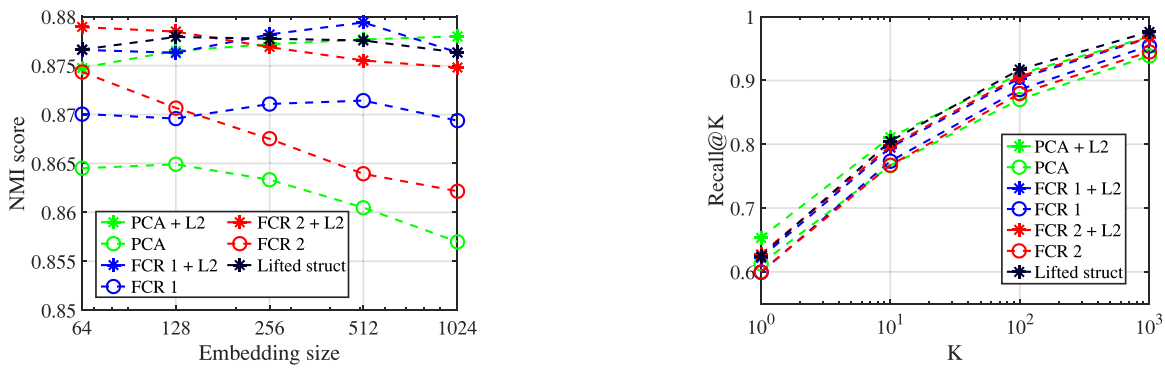


Fig. 6. Comparisons between softmax-based features and lifted structured feature embedding [9] on NMI (clustering) and Recall@K (retrieval) scores for the test set of the Online Products (OP) dataset.The dimension of the feature used in the retrieval experiments is 64.

For training, we multiplied the learning rates of the changed layers (output layers for all models and the fully connected layer added for FCR1 and FCR2) by 10. The batch size was set to 128, and the maximum number of iterations for our training was set to 20,000, which was large enough for the three datasets to converge as mentioned in [11]. These training strategies were exactly the same as those used in [9].

We show the results for CUB and CAR datasets in Fig. 4 and in Fig. 5, respectively, under varying dimensionalities. The deep features extracted from the softmax-based classification networks outperformed the lifted structured feature embedding in clustering (NMI) and retrieval (Recall@K).

For clustering performance measured by NMI, all of the softmax models (PCA, FCR1, and FCR2) showed better scores than the lifted structured feature embedding. Regarding normalization, softmax-based features with L2 normalization showed better performance than those without normalization. The NMI scores of PCA, FCR1 and FCR2 monotonically increased as the feature dimensionality increased for the CUB dataset (Fig. 4). On the other hand, in CAR dataset (Fig. 5), the NMI scores of FCR2 and the lifted structure embeddings decreased from 256 dimensions and those of PCA and FCR1 were saturated above 256 dimensions. This experimental result shows that 1,024 dimensions is too large to represent the image classes of CAR dataset. It also implies that the feature dimensionality should be carefully considered in order to achieve best performance depending on the target data.

For retrieval performance measured by Recall@K metric, the softmax-based features also outperformed features of lifted structured feature embedding. Regarding L2 normalization, features with normalization showed better score than without L2-normalization.

Fig. 6 shows the clustering and retrieval performance measured by NMI, and Recall@K, respectively, for the Online Products dataset. Contrasting to CUB and CAR datasets, the softmax-based

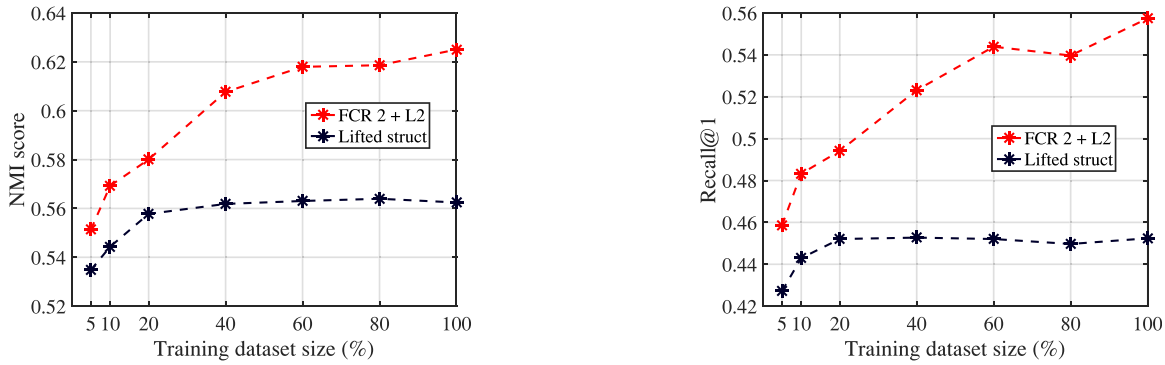


Fig. 7. CUB: NMI (clustering), and Recall@K (retrieval) scores for test set of the Caltech UCSD Birds 200-2011 dataset under different dataset sizes. The feature dimensionality is fixed at 1,024.

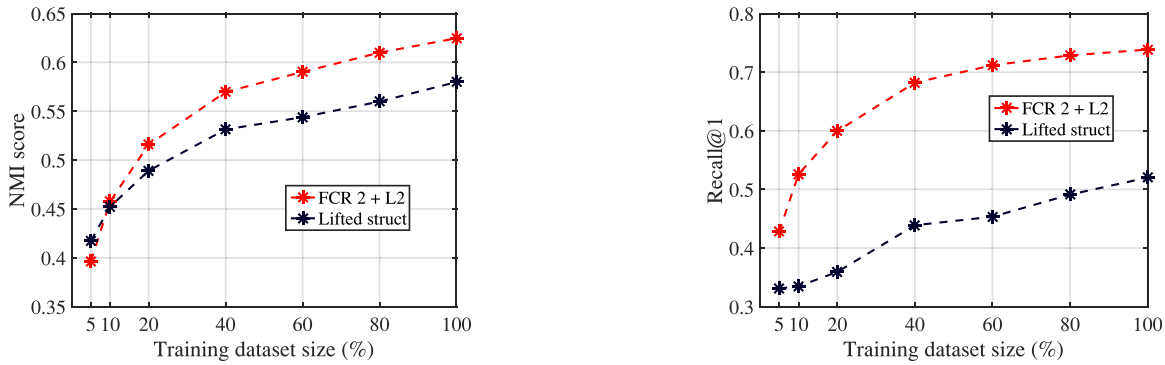


Fig. 8. CAR: NMI (clustering), and Recall@K (retrieval) scores for test set of the Stanford Cars 196 dataset under different dataset sizes. The feature dimensionality is fixed at 256.

features with L2 normalization and the lifted structure embedding showed almost the same performance in the clustering and retrieval. As shown in Table 1, the OP dataset is very different from the CUB and CAR datasets in terms of the number of classes and the number of samples per class—the number of classes is 22 k and the number of samples is 120 k. The number of samples per class in the OP dataset is 5.3 on average, which is far smaller than the CUB and CAR dataset.

4.3 The Effect of the Dataset Scales

From the results for these three datasets, we conjecture that the dataset size—that is the number of samples per class—has a considerable influence on softmax-based features. Hence, we changed the size of datasets by sampling the images of CUB and CAR datasets for each class and ran the experiments again. We constructed seven datasets of different sizes, containing 5, 10, 20, 40, 60, 80, and 100 percent of the whole dataset, respectively. Among them, 5 percent corresponds to approximately 3 and 4 images per class in the CUB and the CAR dataset, respectively. As shown in Figs. 7 and 8, the differences between the scores for softmax and DML were small if the size of the training dataset was small. The gap between softmax and DML became larger as the dataset size increased. The softmax-based classifier was largely influenced by the size of the dataset.

4.4 Distance Metric with Probability Invariant Shift

We define a distance metric that considers the softmax invariance and null space of the linear projection matrix W. When two feature vectors are mapped to the same probability, the distance between the two becomes zero. Assume a vector $\mathbf{u}$ such that

$$W\mathbf{u} = \mathbf{1}. \tag{3}$$

The shift operation $\mathbf{x} + c\mathbf{u}$ has no influence on the softmax operation because $\mathrm{softmax}(W\mathbf{x}) = \mathrm{softmax}(W(\mathbf{x} + c\mathbf{u})) = \mathrm{softmax}(W\mathbf{x} + c\mathbf{1})$, where c is an arbitrary constant. $\mathbf{u}$ exists when the dimensionality of the feature $\mathbf{x}$ is larger than the number of classes to be classified. $\mathbf{u}$ is represented by

$$\mathbf{u} = c_0 W^{\mathsf{T}}\left(WW^{\mathsf{T}}\right)^{-1}\mathbf{1} + \sum_{i=1}^{D-1} c_i \mathbf{v}_i, \tag{4}$$

where $W^{\mathsf{T}}\left(WW^{\mathsf{T}}\right)^{-1}$ is the pseudo-inverse of the linear projection matrix W, $\{\mathbf{v}_1 \dots \mathbf{v}_{D-1}\}$ are the basis vectors that span the null space of W, and $\{c_0 \dots c_{D-1}\}$ are arbitrary constants. The shift operation is called the probability invariant shift in this paper. Using the probability invariant shift, the distance between $\mathbf{x}_1$ and $\mathbf{x}_2$, defined below, removes the effects of the softmax ambiguity and dimensionality reduction

$$d(\mathbf{x}_1, \mathbf{x}_2) = \min_{\{c_0,\dots,c_{D-1}\}} \left\| \mathbf{x}_1 - \mathbf{x}_2 - c_0 W^{\mathsf{T}}\left(WW^{\mathsf{T}}\right)^{-1}\mathbf{1} - \sum_{i=1}^{D-1} c_i \mathbf{v}_i \right\|. \tag{5}$$

TABLE 5
NMI Scores for the Test Set of the Caltech UCSD Birds 200-2011 (CUB) Dataset: Comparisons of the Distance Metric with a Probability Invariant Shift and L2 Normalization

	dimensionality			
	128	256	512	1024
FCR 1	59.4	59.7	60.4	61.1
FCR 1+Shift	59.2	59.1	59.3	59.3
FCR 1+L2	**60.1**	**60.4**	**60.9**	**62.0**
FCR 2	59.6	60.9	61.6	61.8
FCR 2+Shift	59.3	60.0	60.0	60.1
FCR 2+L2	**60.5**	**61.7**	**62.2**	**62.3**

TABLE 6
NMI Scores for the Test Set of the Stanford Cars 196 (CAR) Dataset: Comparisons of the Distance Metric with a Probability Invariant Shift and L2 Normalization

	dimensionality			
	128	256	512	1024
FCR 1	57.6	58.8	58.9	59.4
FCR 1+Shift	57.4	58.2	57.9	57.8
FCR 1+L2	**59.0**	**60.1**	**60.2**	**60.3**
FCR 2	61.4	61.8	61.4	58.7
FCR 2+Shift	61.2	61.3	60.7	58.2
FCR 2+L2	**62.0**	**62.5**	**61.9**	**60.0**

In this section, we present comparative experiments on the distance with a probability invariant shift and with L2 normalization using the CUB and the CAR datasets. Because the CUB and CAR have 100 and 98 classes in their training datasets respectively, we use $\{128, 256, 512, 1024\}$ dimensionality for the features for the experiments.

Tables 5 and 6 show the results of the comparisons. In all cases, the L2 normalization was the most effective. The results demonstrated that the distance metric with a probability invariant shift had little effect on the clustering performance.

5 Conclusion

Because there was no equitable comparison in previous studies, we conducted comparisons of the softmax-based features and the state-of-the-art DML features using a design that would enable these methods to objectively demonstrate their true performance capabilities. Our results showed that the features extracted from softmax-based classifiers performed better than those from state-of-the-art DML methods [9], [10], [11] on fine-grained classification, clustering, and retrieval tasks when the size of the training dataset (samples per class) is large. The results also showed that the size of the dataset largely influenced the performance of softmax-based features. When the size of the dataset was small, DML showed better or competitive performance. DML methods have advantages when the number of classes is very large and the softmax-based classifier is no longer applicable. In DML studies, softmax-based feature have rarely been compared fairly with DML-based feature under the same network architecture or with adequate fine-tuning. This paper revealed that the softmax-based features are still strong baselines. The results suggest that fine-tuned softmax-based features should be taken into account when evaluating the performance of deep features.

5.1 Limitations

- When the number of classes is huge, it is hard to train classification networks due to GPU memory constraints. DML-based methods are suitable for such cases because they do not need the output layer which is proportional to the number of classes.
- For cross-domain tasks, such as sketches to photos [30], [31] or aerial views to ground views [32], DML is effective. Classification-based learning needs complicated learning strategies like in [33]. DML-based methods can learn cross-domain representation only by using a pair of networks.
- For datasets with continuous labels, DML-based methods might be helpful because classifier-based method cannot deal with them. However, most recent DML studies are specialized to datasets with discrete labels. To utilize the methods to datasets with continuous labels, some extensions are necessary.

Acknowledgments

This work is partially supported by JST CREST JPMJCR19F4 and JSPS KAKENHI 18H03254.

References

[1] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647–655.

[2] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2014, pp. 512–519.

[3] Y. Liu, Y. Guo, S. Wu, and M. S. Lew, "DeepIndex for accurate and efficient image retrieval," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2015, pp. 43–50.

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[5] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 157–166.

[6] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 584–599.

[7] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 98.

[8] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.

[9] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4004–4012.

[10] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 1857–1865.

[11] H. O. Song, S. Jegelka, V. Rathod, and K. Murphy, "Deep metric learning via facility location," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2206–2214.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[13] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1735–1742.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[18] Q. Qian, R. Jing, S. Zhu, and Y. Lin, "Fine-grained visual categorization via multi-stage metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3716–3724.

[19] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "Siamese" time delay neural network," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 1994, pp. 737–744.

[20] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 539–546.

[21] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *J. Mach. Learn. Res.*, vol. 11, pp. 1109–1135, 2010.

[22] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. Int. Workshop Similarity-Based Pattern Recognit.*, 2015, pp. 84–92.

[23] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1701–1708.

[24] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200–2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.

[25] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. 4th Int. IEEE Workshop 3D Representation Recognit.*, 2013, pp. 554–561.

[26] L. Wei, Q. Huang, D. Ceylan, E. Vouga, and H. Li, "Dense human body correspondences using convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1544–1553.

[27] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *Proc. ACM Multimedia*, 2014, pp. 675–678.
[28] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[29] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.
[30] Q. Yu, F. Liu, Y.-Z. Song, T. Xiang, T. M. Hospedales, and C.-C. Loy, "Sketch me that shoe," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 799–807.
[31] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, "The sketchy database: Learning to retrieve badly drawn bunnies," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 119:1–119:12, 2016.
[32] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, "Learning deep representations for ground-to-aerial geolocalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5007–5015.
[33] L. Castrejon, Y. Aytar, C. Vondrick, H. Pirsiavash, and A. Torralba, "Learning aligned cross-modal representations from weakly aligned data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2940–2949.