SIGNIFICANCE OF SOFTMAX-BASED FEATURES OVER METRIC LEARNING-BASED FEATURES

Shota Horiguchi, Daiki Ikami & Kiyoharu Aizawa
Department of Information and Communication Engineering
The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo, JP
{horiguchi,ikami,aizawa}@t.u-tokyo.ac.jp

ABSTRACT

The extraction of useful deep features is important for many computer vision tasks. Deep features extracted from classification networks have proved to perform well in those tasks. To obtain features of greater usefulness, end-to-end distance metric learning (DML) has been applied to train the feature extractor directly. End-to-end DML approaches such as Magnet Loss and lifted structured feature embedding show state-of-the-art performance in several image recognition tasks. However, in these DML studies, there were no equitable comparisons between features extracted from a DML-based network and those from a softmax-based network. In this paper, by presenting objective comparisons between these two approaches under the same network architecture, we show that the softmax-based features are markedly better than the state-of-the-art DML features for tasks such as fine-grained recognition, attribute estimation, clustering, and retrieval.

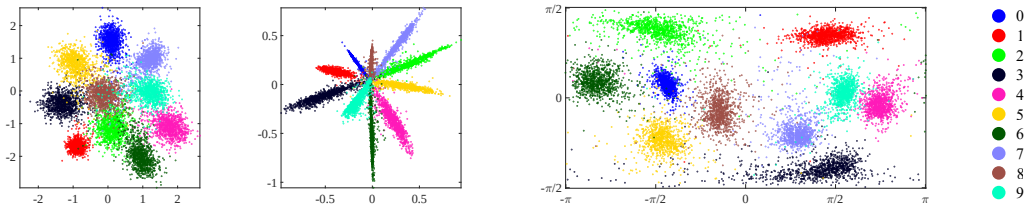
1 INTRODUCTION

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 (Donahue et al., 2014), detection, and retrieval (Razavian et al., 2014; Liu et al., 2015), even when they have been trained to classify 1000 classes of the ImageNet dataset (Russakovsky et al., 2015). It has also been shown that fine-tuning for target domains further improves the features' performance (Wan et al., 2014; Babenko et al., 2014).

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 (Bell & Bala, 2015; Schroff et al., 2015). Some DML methods have been reported to show state-of-the-art performance in fine-grained classification (Rippel et al., 2016) and clustering and retrieval (Song et al., 2016) contexts.

However, there have been few experiments comparing softmax-based feature extraction with DML-based feature extraction 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. As we explain more fully in the following section, we contend that there is no reason that DML, which learns feature embedding explicitly, should outperform a softmax-based feature extractor.

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 (LeCun et al., 1998). For DML, we used the contrastive loss function (Hadsell et al., 2006) 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 (Fig. 1b, Fig. 1c).



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

Figure 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.

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. This is not technically novel, but this must be useful for fair comparison between image representations.
- We demonstrate that deep features extracted from softmax-based classification networks show markedly better results on fine-grained classification, attribute estimation, clustering, and retrieval tasks than those from DML-based networks in almost all datasets.
- We show that DML-based methods offer performance competitive to softmax-based methods only when the training dataset consists of a very small number of samples per class.

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 (Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; Szegedy et al., 2015; He et al., 2016). It has been shown that features extracted from classification networks can be repurposed as a good feature representation for novel tasks (Donahue et al., 2014; Razavian et al., 2014; Qian et al., 2015) even if the network was trained on ImageNet (Russakovsky et al., 2015). For obtaining better feature representations, fine-tuning is also effective (Babenko et al., 2014).

2.1.2 Deep Distance Metric Learning

Distance metric learning (DML), which learns a distance metric, has been widely studied (Bromley et al., 1994; Chopra et al., 2005; Chechik et al., 2010; Qian et al., 2015). Recent studies have focused on end-to-end deep distance metric learning (Bell & Bala, 2015; Schroff et al., 2015; Li et al., 2015; Rippel et al., 2016; Song et al., 2016). 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 & Bala (2015) 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. (2015) 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., 2014); thus, triplet loss might not have been the only reason for the performance improvement, and the contribution from adopting triplet loss remains uncertain. Rippel et al. (2016) used the Magnet

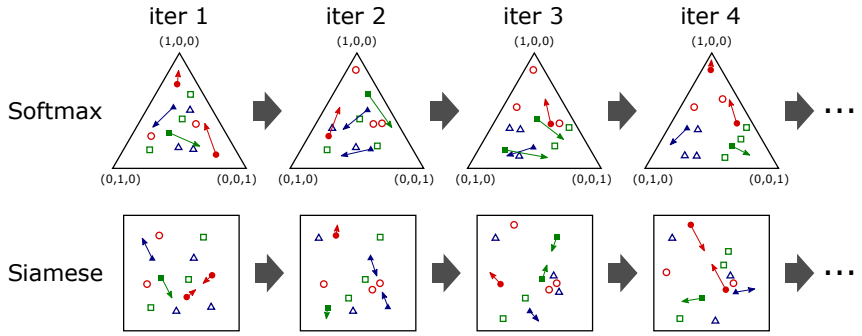


Figure 2: Illustration of learning processes for softmax-based classification network and siamese-based DML network. For softmax, the gradient is defined by the distance between a sample and a fixed one-hot vector, and for siamese by the distance between samples.

Loss function for their DML. They tried softmax-based features as a comparison, but their results are unfairly low from our results as shown in Section 4.2 and 4.3. Song et al. (2016) used lifted structured feature embedding, another state-of-the-art DML method; however, they only compared their method with a softmax-based classification network pretrained on ImageNet (Russakovsky et al., 2015) and did not compare it with a fine-tuned network.

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 Hadsell et al. (2006) considers the distance d between a pair of samples:

$$E = \frac{1}{2} q d^2 + (1 - q) \max(\alpha - d, 0), \tag{3}$$

where α represents the margin and $q \in \{0, 1\}$ indicates whether the images in a pair are in the same class (1) or not (0). Recent studies (Schroff et al., 2015; Rippel et al., 2016; Song et al., 2016) use pairwise distances between three or more images at the same time for fast convergence and efficient calculation. However, these methods have some drawbacks. DML methods sometimes require complicated operations such as hard negative sampling (Schroff et al., 2015; Rippel et al., 2016) and k-means clustering for every epoch (Rippel et al., 2016). 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 very slowly and is not stable and unsteadily. An additional problem is that methods for sampling positive pairs and negative pairs have not been established.

3 METHODS

3.1 DIMENSIONALITY REDUCTION LAYER

One of DML's strength in using fine-tuning is the flexibility of its output dimensionality. 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 (Babenko et al., 2014) 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 1024-dimensional pool5 layer of GoogLeNet (Szegedy et al., 2015; Ioffe & Szegedy, 2015) (Fig. 3a) and applied PCA to reduce the dimensionality. 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 (FCR 1, Fig. 3b). We also investigated a third approach in which a fully connected layer is added followed by a dropout layer (FCR 2, Fig. 3c). We intend to show that the features extracted from the pool5 layer of FCR 2 provide better performance than those from FCR 1 even though they differ only in the positions of their dropout layers.

3.2 NORMALIZATION

In this study, all the features extracted from the classification networks were from the last layer before the last output layer. The outputs were normalized by the softmax function and then evaluated by the cross-entropy loss function in the networks. Assume that the output vector is $\mathbf{p} = \{p_i | \sum_i p_i = 1\}$. For arbitrary positive constant α, $\mathbf{y} = \{\log \alpha p_i\}$ returns the same vector p after the softmax function is applied. The features $\mathbf{x}$ we extract from the networks are given as $\mathbf{x} = W^{-1}\mathbf{y}$, where W denotes the linear projection matrix from the layer before the output layer to the output layer. The vector $\mathbf{y}$ has an ambiguity in its scale, thus vector $\mathbf{x}$, a linear transform of $\mathbf{y}$, also has an ambiguity in the scale; therefore $\mathbf{x}$ should be normalized. 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.

Some studies used L2 normalization for deep features extracted from softmax-based classification networks (Taigman et al., 2014), whereas many recent studies have used the features without any normalization (Krizhevsky et al., 2012; Rippel et al., 2016; Song et al., 2016; Wei et al., 2016). In this study, we also planned to validate the efficiency of normalizing deep features.

4 EXPERIMENTS

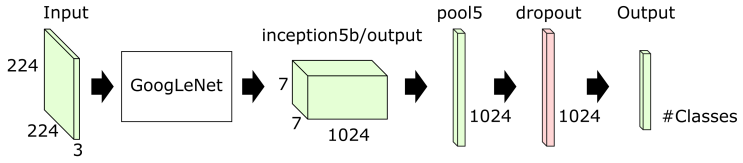
In this section, we compare the deep features extracted from classification networks to those reported from state-of-the-art deep metric learning methods (Rippel et al., 2016; Song et al., 2016) in their performance on several tasks.

4.1 PROCEDURE

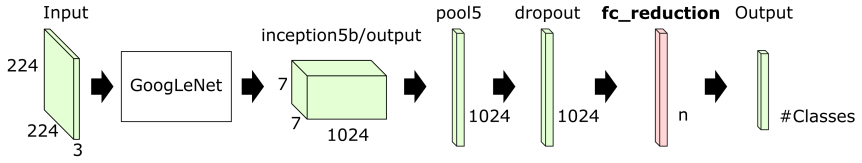
All our networks were fine-tuned from the weights that were pretrained on ImageNet (Russakovsky et al., 2015). To evaluate fine-grained classification and attribute estimation performances, we used GoogLeNet with batch normalization (Ioffe & Szegedy, 2015) and did not use any dimentionality reduction layers described in Section 3.1. To evaluate clustering and retrieval performances we used GoogLeNet without batch normalization (Szegedy et al., 2015) and dimentionality reduction layers. We used the Caffe (Jia et al., 2014) framework for our experiments.

4.2 FINE-GRAINED CLASSIFICATION

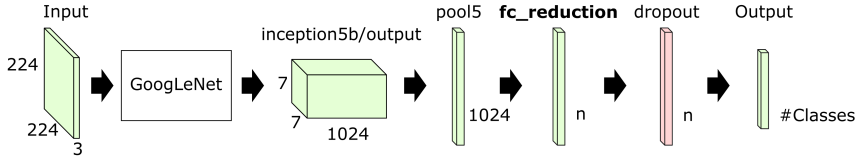
For the evaluation of deep features in fine-grained classification tasks, we used three image datasets: Stanford Dogs (Khosla et al., 2011), Oxford 102 Flowers (Nilsback & Zisserman, 2008), and Oxford-IIIT Pet (Parkhi et al., 2012). For the softmax-base method we fine-tuned the classifier from weights that were pretrained on ImageNet. We defined the learning rate using validation data, setting the learning rate to 0.0001 for the Stanford Dogs dataset and the Oxford-IIIT Pet dataset and to 0.001 for the Oxford 102 Flowers dataset. Learning rates were not changed during the training.



(a) GoogLeNet (dimensionality reduction by PCA)



(b) GoogLeNet with dimensionality reduction by a fully connected layer just before the output layer (FCR 1)



(c) GoogLeNet with dimensionality reduction by a fully connected layer followed by a dropout layer (FCR 2)

Figure 3: GoogLeNet Szegedy et al. (2015) architecture we use in this paper. We extracted the features from the red-colored layers. For (a), we applied PCA to reduce the number of feature dimensions. For (b) and (c), the dimensionality is already reduced to the required number by the fc_reduction layer.

We rescaled all the input up by 30% and randomly cropped 224×224. These strategies are exactly the same as those of the previous method (Rippel et al., 2016).

We show the mean error rates for the three datasets in Table 1. All our results were evaluated using a 1-nearest neighbor search of the 1024-dimensional vectors extracted from the pool5 layer of GoogLeNet with batch normalization (Ioffe & Szegedy, 2015). In all the experiments, the features extracted from the fine-tuned classification network show the best fine-grained classification performance. Our results of softmax-based classification are better than the results in Rippel et al. (2016). The experiments of softmax-based classification in Rippel et al. (2016) were not the best.

4.3 Attribute Estimation

Rippel et al. (2016) evaluated features' expressiveness using mean attribute precision and showed that the features generated by their proposed method contain intra-class diversity. In this section, we investigate the intra-class diversity of softmax features. We use the ImageNet Attribute dataset (Rippel et al., 2016), which consists of overlap between the ImageNet training set (Russakovsky et al., 2015) and the Object Attribute dataset (Russakovsky & Fei-Fei, 2010). We used only the images and their class labels during our training of the softmax classifier and did not use attributes.

Table 2 shows the error rates of 90-way classification under different training methods. Our fine-tuned softmax classifier outperformed those of Rippel et al. (2016) by a considerable margin. Fig. 4 shows the mean attribute precision for the ImageNet Attribute dataset. Our fine-tuned softmax features markedly outperformed those from Rippel et al. (2016). These results implicitly indicate that the features extracted from the pool5 layer contain intra-class diversity that is better than those from DML networks designed to keep intra-class diversity.

Table 1: Error rates for various fine-grained image datasets.

(a) Stanford Dogs.

Approach	Error
Angelova & Long (2014)	51.7%
Xie et al. (2015)	50.6%
Gavves et al. (2013)	49.9%
Gavves et al. (2015)	43.0%
Qian et al. (2015)	30.9%
Rippel et al. (2016) (Softmax prob)	26.6%
Rippel et al. (2016) (Triplet)	35.8%
Rippel et al. (2016) (Magnet)	24.9%
Ours (Softmax prob)	**18.3%**
Ours (Softmax pool5)	21.0%
Ours (Softmax pool5 + L2)	20.3%

(b) Oxford 102 Flowers.

Approach	Error
Angelova & Zhu (2013)	23.3%
Angelova & Long (2014)	19.6%
Murray & Perronnin (2014)	15.4%
Razavian et al. (2014)	13.2%
Qian et al. (2015)	11.6%
Rippel et al. (2016) (Softmax prob)	11.2%
Rippel et al. (2016) (Triplet)	17.0%
Rippel et al. (2016) (Magnet)	8.6%
Ours (Softmax prob)	8.69%
Ours (Softmax pool5)	7.90%
Ours (Softmax pool5 + L2)	**7.09%**

(c) Oxford -IIIT Pet.

Approach	Error
Angelova & Zhu (2013)	49.2%
Parkhi et al. (2012)	46.0%
Angelova & Long (2014)	44.6%
Murray & Perronnin (2014)	43.2%
Qian et al. (2015)	19.6%
Rippel et al. (2016) (Softmax prob)	11.3%
Rippel et al. (2016) (Triplet)	13.5%
Rippel et al. (2016) (Magnet)	10.6%
Ours (Softmax prob)	**9.04%**
Ours (Softmax pool5)	9.09%
Ours (Softmax pool5 + L2)	**9.04%**

Table 2: Classification error rates for the ImageNet Attribute dataset.

Approach	Error
Rippel et al. (2016) (Softmax prob)	14.1%
Rippel et al. (2016) (Triplet)	26.8%
Rippel et al. (2016) (Magnet)	15.9%
Ours (Softmax prob)	**7.68%**
Ours (Softmax pool5)	11.9%
Ours (Softmax pool5 + L2)	10.7%

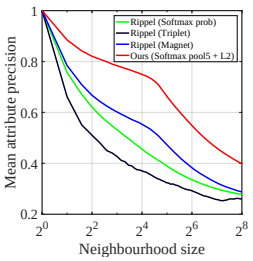


Figure 4: Mean attribute precision for the ImageNet Attribute dataset.

Table 3: Properties of datasets used in Section 4.4. Each cell shows the number of images (upper figure) and the number of classes (lower figure).

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

4.4 Clustering and Retrieval

Here, we give our evaluation of clustering and retrieval scores for the state-of-the-art DML method (Song et al., 2016) and for the softmax classification networks. We used the Caltech UCSD Birds 200-2011 (CUB) dataset (Wah et al., 2011), the Stanford Cars 196 (CAR) dataset (Krause et al., 2013), and the Stanford Online Products (OP) dataset (Song et al., 2016). 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 3. We emphasize that the class sets used for training and testing are completely different. We multiplied the learning rates of the changed layers (output layers for all models and the fully connected layer added for FCR 1 and FCR 2) by 10. The batch size was set to 128, and the maximum number of iterations for our training was set to 20,000. These training strategies are exactly the same as those used in the earlier study (Song et al., 2016).

For clustering evaluation, we applied k-means clustering 100 times and calculated the average standard F_1 and NMI (Manning et al., 2008); the value for k was set to the number of classes in the test set. For retrieval evaluation, we used the Recall@K metric (Jegou et al., 2011).

We show the results for the CUB dataset in Fig. 5 and for the CAR dataset in Fig. 6. We notice that we have been able to reproduce nearly exactly the scores of lifted structured feature embedding (Song et al., 2016). However, the deep features extracted from the softmax-based classification networks outperformed the lifted structured feature embedding in all the evaluation metrics.

For F_1 and NMI, all of the softmax models, including PCA, FCR 1, and FCR 2, show markedly better scores than does lifted structured feature embedding. It is clear that L2 normalization improves the scores of all the softmax-based models. The scores of PCA and FCR 1 drop slightly as the feature dimensionality decreases from 1024 for both the CUB dataset and the CAR dataset. On the other hand, FCR 2, which has a fully connected layer followed by a dropout layer, improves the scores in spite of the reduction in dimensionality, as shown in Fig. 6. It may be that 1024 dimensions is too large to describe the image classes. This result may imply that to obtain the best features we

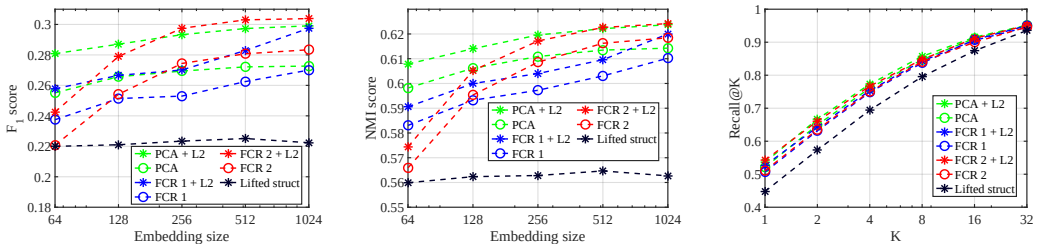


Figure 5: F_1, NMI, and Recall@K scores for the test set of the Caltech UCSD Birds 200-2011 dataset.

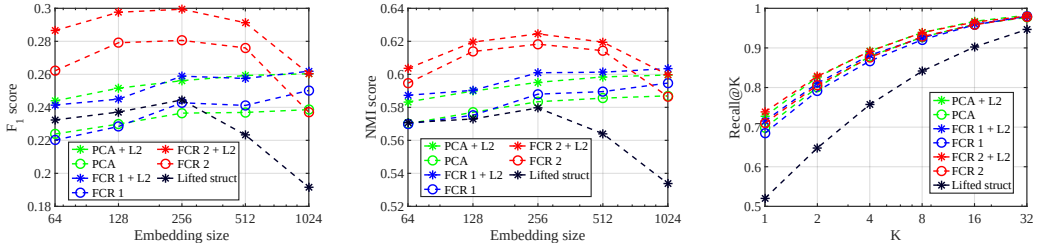


Figure 6: F_1, NMI, and Recall@K scores for the test set of the Stanford Cars 196 dataset.

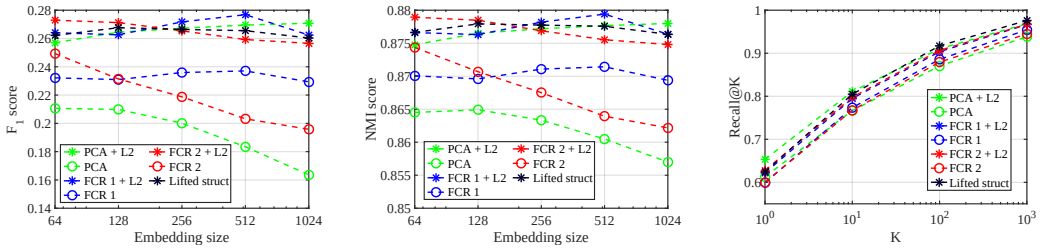


Figure 7: F_1, NMI, and Recall@K scores for the test set of the Online Products dataset.

need to first determine the optimum dimensionality of the feature space for the dataset and then apply PCA.

For the Recall@K metric, we used 1024-dimensional features for the CUB dataset and 256-dimensional features for the CAR dataset. The softmax-based features outperformed the DML-based features. The differences between PCA, FCR 1, and FCR 2 are very minor. Regarding feature normalization, features without normalization show worse scores than do L2-normalized features.

Fig. 7 shows the standard F_1, NMI, and Recall@K for the Online Products dataset. We used 1024-dimensional features for the Recall@K metric. As shown in Table 3, 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 samples per class in the OP dataset is limited to 5.3 on average. In contrast to CUB and CAR, in the OP dataset the scores for softmax and for lifted structured feature embedding are nearly the same.

From the results for these three datasets, we conjecture that the number of images contained in the dataset has a considerable effect on softmax-based classification. In other words, it is difficult for DML to make use of the rich information from a large number of samples because of the randomness described in the previous section. Hence, we changed the size of datasets by subsampling 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 %, respectively, of the whole dataset. As shown in Fig. 8 and Fig. 9, the difference between the scores for softmax and DML is small or close to zero if the size of the training dataset is small. The gap between softmax and DML becomes larger as the dataset size increases. It is surprising that the scores of lifted structured feature embedding on the CUB dataset did not increase even though we used more images for the training

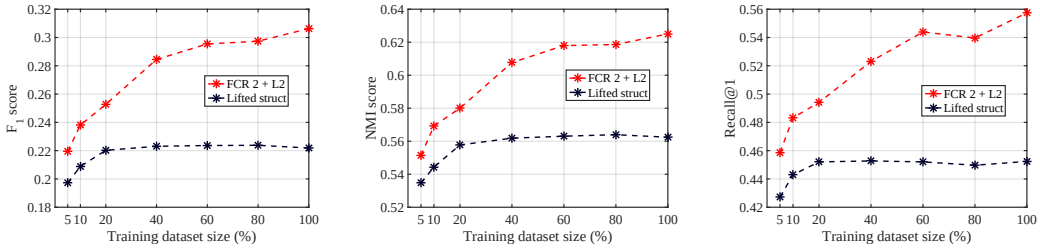


Figure 8: F_1, NMI, and Recall@K scores for test set of the Caltech UCSD Birds 200-2011 dataset under different dataset sizes. The feature dimensionality is fixed at 1024.

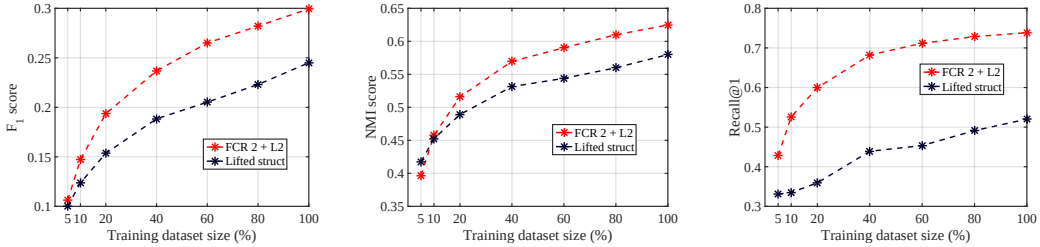


Figure 9: F_1, NMI, and Recall@K scores for test set of the Stanford Cars 196 dataset under different dataset sizes. The feature dimensionality is fixed at 256.

(Fig. 8). It can be said that DML cannot exploit large training datasets, whereas the softmax-based classifier can obtain features of high expressiveness.

5 Conclusion

Because there was no equitable comparison in previous studies, we conducted comparisons of the softmax-based classifier and DML methods using a design that would enable the methods to objectively demonstrate their true performance capabilities. Our results show that the features extracted from softmax-based classifiers perform better than those from state-of-the-art DML methods (Rippel et al., 2016; Song et al., 2016) on fine-grained classification, clustering, and retrieval tasks, especially when the size of the training dataset is large. The experimental results also show that softmax-based features exhibit rich intra-class diversity even though the softmax classifier is not explicitly designed to do so, unlike to the previous method (Rippel et al., 2016). It is obvious that the softmax-based features are still strong baselines. We hope that softmax-based features are taken into account when evaluating the performance of deep features.

Limitations. When the number of classes are 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 (Yu et al., 2016; Sangkloy et al., 2016) or aerial views to ground views (Lin et al., 2015), DML is also effective. Classification-based learning needs complicated learning strategies like in Castrejon et al. (2016). DML-based methods can learn cross-domain representation only by using a pair of networks.

References

Anelia Angelova and Philip M. Long. Benchmarking large-scale fine-grained categorization. In *WACV*, pp. 532–539, 2014.

Anelia Angelova and Shenghuo Zhu. Efficient object detection and segmentation for fine-grained recognition. In *CVPR*, pp. 811–818, 2013.

Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *ECCV*, pp. 584–599, 2014.

Sean Bell and Kavita Bala. Learning visual similarity for product design with convolutional neural networks. *SIGGRAPH*, 34(4), 2015.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *NIPS*, pp. 737–744. 1994.

Lluis Castrejon, Yusuf Aytar, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Learning aligned cross-modal representations from weakly aligned data. In *CVPR*, pp. 2940–2949, 2016.

Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *JMLR*, 11:1109–1135, 2010.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, pp. 539–546, 2005.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *ICML*, pp. 647–655, 2014.

E. Gavves, B. Fernando, C. G. M. Snoek, A. W. M. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, pp. 1713–1720, 2013.

Efstratios Gavves, Basura Fernando, Cees G. M. Snoek, Arnold W. M. Smeulders, and Tinne Tuytelaars. Local alignments for fine-grained categorization. *IJCV*, 111(2):191–212, 2015.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, pp. 1735–1742, 2006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducint internal covariate shift. In *ICML*, pp. 448–456, 2015.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *TPAMI*, 33(1):117–128, 2011.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, CVPR*, 2011.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition*, pp. 554–561, 2013.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pp. 1097–1105, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pp. 2278–2324, 1998.

Yangyan Li, Hao Su, Charles Ruizhongtai Qi, Noa Fish, Daniel Cohen-Or, and Leonidas J. Guibas. Joint embeddings of shapes and images via CNN image purification. *ACM TOG*, 34(6):234:1–234:12, 2015.

Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocalization. In *CVPR*, pp. 5007–5015, 2015.

Yu Liu, Yanming Guo, Song Wu, and Michael S. Lew. DeepIndex for accurate and efficient image retrieval. In *ICMR*, pp. 43–50, 2015.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

Naila Murray and Florent Perronnin. Generalized max pooling. In *CVPR*, pp. 2473–2480, 2014.

M. E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 722–729, 2008.

O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, pp. 3498–3505, 2012.

Qi Qian, Rong Jing, Shenghuo Zhu, and Yuanqing Lin. Fine-grained visual categorization via multi-stage metric learning. In *CVPR*, pp. 3716–3724, 2015.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *CVPR Workshops*, pp. 512–519, 2014.

Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. Metric learning with adaptive density discrimination. In *ICLR*, 2016.

Olga Russakovsky and Li Fei-Fei. Attribute learning in large-scale datasets. In *ECCV, International Workshop on Parts and Attributes*, 2010.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: Learning to retrieve badly drawn bunnies. *ACM TOG*, 35(4):119:1–119:12, 2016.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, pp. 815–823, 2015.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, pp. 4004–4012, 2016.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pp. 1–9, 2015.

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. DeepFace: Closing the gap to human-level performance in face verification. In *CVPR*, pp. 1701–1708, 2014.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Ji Wan, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. In *ACMMM*, pp. 157–166, 2014.

Lingyu Wei, Qixing Huang, Duygu Ceylan, Etienne Vouga, and Hao Li. Dense human body correspondences using convolutional networks. In *CVPR*, pp. 1544–1553, 2016.

Saining Xie, Tinbao Yang, Xiaoyu Wang, and Yuanqing Lin. Hyper-class augmented and regularized deep learning for fine-grained image classification. In *CVPR*, pp. 2645–2654, 2015.

Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M. Hospedales, and Chen-Change Loy. Sketch me that shoe. In *CVPR*, pp. 799–807, 2016.