Learning Discriminative and Shareable Features for Scene Classification

Zhen Zuo[†], Gang Wang[†‡], Bing Shuai[†], Lifan Zhao[†],
Qingxiong Yang[§], and Xudong Jiang[†]

[†]Nanyang Technological University, Singapore,
[‡]Advanced Digital Sciences Center, Sinapore, [§]City University of Hong Kong

Abstract. In this paper, we propose to learn a discriminative and shareable feature transformation filter bank to transform local image patches (represented as raw pixel values) into features for scene image classification. The learned filters are expected to: (1) encode common visual patterns of a flexible number of categories; (2) encode discriminative and class-specific information. For each category, a subset of the filters are activated in a data-adaptive manner, meanwhile sharing of filters among different categories is also allowed. Discriminative power of the filter bank is further enhanced by enforcing the features from the same category to be close to each other in the feature space, while features from different categories to be far away from each other. The experimental results on three challenging scene image classification datasets indicate that our features can achieve very promising performance. Furthermore, our features also show great complementary effect to the state-of-the-art ConvNets feature.

Keywords: Feature learning, Discriminant analysis, Information sharing, Scene Classificsion

1 Introduction

Generating robust, informative, and compact local features has been considered as one of the most critical factors for good performance in computer vision. In the last decade, numerous hand-crafted features, such as SIFT [1] and HOG [2], have ruled the local image representation area. Recently, a number of papers [3–9] have been published to learn feature representations from pixel values directly, aiming to extract data-adaptive features which are more suitable. However, most of these works operate in an unsupervised way without considering the class label information. We argue that extracting discriminative features is important for classification, as information on local patches is usually redundant, features which are discriminative for classification should be extracted.

In this paper, we develop a method to learn transformation filter bank to transform pixel values of local image patches into features, which is called Discriminative and Shareable Feature Learning (DSFL). As shown in Fig. 1, we aim to learn an over-complete filter bank, which is able to cover the variances

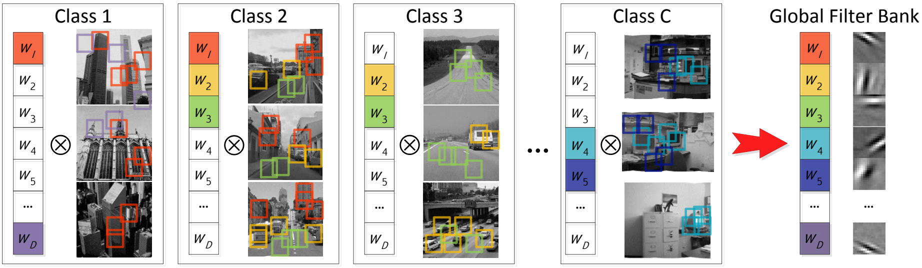


Fig. 1. Illustration of DSFL. $w_1, ..., w_D$ represent the filters in the global filter bank W. For each class, we force it to activate a small subset of filters to learn class-specific patterns, and different classes can share the same filters to learn shareable patterns. Finally, the feature of a image patch x_i can be represented as $f_i = \mathcal{F}(Wx_i)$. (Best viewed in color)

of images from different classes, meanwhile keeping the shareable correlation among different classes and discriminative power of each category. To build such a global filter bank, an intuitive way is to independently learn a filter bank for each class, and concatenate them together. However, if filters learned from different classes are not shared, the number of filters will increase linearly with the number of categories, which is not desirable for local feature representation. To learn a more compact global filter bank, we force each category to only activate a subset of the global filters during the learning procedure. Beyond reducing feature dimensions, sharing filters can also lead to more robust features. Images belonging to different classes do share some information in common (e.g. in scene classification, both 'computer room' and 'office' contain 'computer' and 'desk'). The amount of information shared depends on the similarity between different categories. Hence, we allow filters to be shared, meaning that the same filters can be activated by a number of categories. We introduce a binary selection variable vector to adaptively select what filters to share, and among what categories.

To improve the discrimination power, we introduce a discriminative term to force features from the same category to be close and features from different categories to be far away. (e.g. there might be patches corresponding to bookshelf in 'office', which can hardly be found in 'computer room'). However, not all the patches from the same categories are close, as they are very diverse. Hence, we introduce a method to select discriminative exemplars from each category, and a feature should be similar to a subgroup of the exemplars from the same category. Furthermore, not all the local patches from different classes should be forced to be separable, thus, we relax the discriminative term to allow sharing similar patches across different classes, and focus on separating the less similar patches from different classes.

We tested our method on three widely used scene image classification datasets: Scene 15, UIUC Sports, and MIT 67 Indoor. The experimental results show that our features can outperform most of the existing ones. By combining our feature with the ConvNets [3,10] features (supervised pretrained on ImageNet [11]), we can achieve state-of-the-art results on Scene 15, UIUC Sports and MIT 67 Indoor with a classification accuracy of 92.81% , 96.78%, and 76.23% respectively.

2 Related Works

Our work focuses on learning local feature descriptors. Hand-crafted features including SIFT [1], HOG [2], GIST [12], and LBP [13] were popular used in this area. However, even though they are very powerful, they can hardly capture any information other than what have been defined by prior knowledge. In this paper, we aim to learn a data adaptive local feature representation.

Recently, directly learning features from image pixel values [4–9,14–18] emerges as a hot research topic in computer vision because it is able to learn data adaptive features. And many of them have achieved superior performance on many important computer vision tasks such as digital image recognition [6], and action recognition [17]. However, most existing feature learning works adopt unsupervised learning methods to learn filters for feature extraction. Different from them, we argue that discriminative information can be critical for classification and discriminative patterns can be learned. We experimentally show that our discriminative feature learning works better than unsupervised feature learning on scene datasets by encoding the shareable and discriminative class correlation clues into feature representation. While in the supervised feature learning line, the ConvNets [3] is a very deep feature learning structure (5 convolutional layers, 2 fully connected layers, and 1 softmax layer), it focuses on progressively learning multi-levels of visual patterns. When pre-trained on ImageNet, it is the state-of-the-art feature extractor on many tasks [10, 19, 20]. In contrast, our DSFL focuses on encoding the shareable and discriminative correlation among different classes into each layer's feature transformation. In the Section 4, we will show that our DSFL learns significant complementary information to this powerful feature, and combines with which, we can update the current state-of-the-art on all of the three scene classification datasets.

There are also some related papers trying to extract discriminative representations from images. For example, [21–24] learn discriminative dictionaries to encode local image features. Another line of work [25–28] that represents scene images in terms of weakly-supervised mined discriminative parts gained increasingly popularity and success. The basic idea is to build a discriminative framework, and use it to mine a set of representative and distinct parts (multi-scale patches) for every class. Afterwards, images can be represented with the max pooled responses of such mid-level patterns. Different from these works, we focus on discriminatively learning filters to transform local image patches into features, and allowing sharing local feature transformation filters between different categories. To the best of our knowledge, this hasn't been done before. Furthermore, in [29, 30], object part filters at the middle level are shared to represent a large number of object categories for object detection. Compared to them, our training examples (image patches) don't have strong supervised labels except image-level class labels, so we develop an exemplar selection scheme and a nearest neighbour based maximum margin method to make it more robust to noise.

3 Discriminative and Shareable Feature Learning

In this section, we first describe the three components of our Discriminative and Shareable Feature Learning (DSFL) framework. Then we will provide an alternating optimization strategy to solve this problem.

3.1 DSFL Learning Components

We aim to learn features that can preserve the information of the original data, be shareable and be discriminative. To achieve these goals, we have three learning components in the DSFL learning framework. We write $x \in \mathbb{R}^{D_o}$ as a vector of raw pixel values of an image patch. Given a number of x from different categories, we aim to learn a feature transformation filter bank $W \in \mathbb{R}^{D \times D_0}$ (each row represents one filter, and there are D filters). By multiplying W with x, and applying an activation function $\mathcal{F}(\cdot)$, we expect to generate feature $f_i = \mathcal{F}(Wx_i)$, which is discriminative and as compact as possible. For this purpose, W should be learned to encode information which is discriminative among classes and only has a small number of rows (filters). In our learning framework, we force each class to activate a subset of filters in W to learn class-specific patterns. And we allow different classes to share filters to reduce the number of filters.

The Global Reconstruction Term To ensure that the feature transformation matrix $W \in \mathbb{R}^{D \times D_0}$ can preserve the information hidden in the original data, we utilize a global reconstruction term, which aims to minimize the error between the reconstructed data and the original data. The cost function is shown as following:

$$\begin{gathered} L_u = \sum_{i=1}^{N} \mathcal{L}_u(x_i, W) + \lambda_1 \sum_{i=1}^{N} \|f_i\|_1 \\ \text{where } \mathcal{L}_u(x_i, W) = \left\|x_i - W^T W x_i\right\|_2^2 \\ \text{and } f_i = \mathcal{F}(Wx_i), \ \mathcal{F}(\cdot) = \text{abs}(\cdot) \end{gathered} \tag{1}$$

where N is the total number of training patches. $\mathcal{L}_u$ is the empirical loss function with respect to global filter bank W and unlabelled training patch x_i. $W^T W x_i$ denotes the reconstructed data of x_i. This auto-encoder [4,31] style reconstruction cost penalization term can not only prevent W from degeneration, but also allow W to be over-complete. The term $\|f_i\|_1$ is used to enforce the sparsity of the learned feature f_i. Following [5,17], we set $\mathcal{F}(\cdot) = \text{abs}(\cdot)$. Then the sparse term $\|f_i\|_1$ degenerates to summation of all the dimensions of f_i.

Shareable Constraint Term Equation 1 can only learn a generative W without encoding any class-specific information. A method to overcome this limitation is to force a subset of filters to only respond to a specific class. Thus, we propose a constraint term to ensure that only a subset of the filters will be activated by one class, while the same filters can potentially be activated by multiple classes. For each class c, we write $\alpha^c \in \mathbb{R}^D$ as a vector to indicate the selection status of rows of W. If $\alpha_d^c = 1,\ d = 1, ..., D$, then the d-th row of W is activated, otherwise not activated. We use $A^c = \operatorname{diag}(\alpha^c)$ for representation convenience. The cost of our shareable constraint term of class c is formulated as following:

$$\begin{gathered} L_{\text{sha}}^c = \sum_{j=1}^{N_c} \mathcal{L}_{\text{sha}}^c \left(x_j^c, A^c W\right) + \lambda_2 \|\alpha^c\|_0 \\ s.t.\ \ \alpha_d^c \in \{0,1\}\,,\ d = 1, ..., D \\ \text{where}\ \ \mathcal{L}_{\text{sha}}^c \left(x_j^c, A^c W\right) = \left\| x_j^c - \left(A^c W\right)^T \left(A^c W\right) x_j^c \right\|_2^2 \end{gathered} \tag{2}$$

where N_c is the number of training patches from class c, and C is the total number of classes. For the shareable term, similar to $\mathcal{L}_u$, $\mathcal{L}_{\text{sha}}^c$ is the reconstruction cost function with respect to the filter bank subset $\alpha^c W$ and training patch x_j^c from class c. We apply l_0 norm on α^c to force each class to activate a small number of rows. Consequently, for the d-th element in α^c, if it is only set to 1 for class c, then it means the d-th row of W will only be activated and learned with training patches from class c. If the d-th element is set to 1 for class c_1 and class c_2, then the d-th row of W is a shareable filter, which should be activated and learned with training data from class c_1 and c_2. When α^c is updated in each iteration, the corresponding training data for each filter will also be updated.

Discriminative Regularization Term To enhance the discriminative power of feature descriptors, we further introduce a discriminative term based on the assumption that discriminative features should be close to the features from the same category, and be far away from the features from different categories in the feature space. In the image level scenario [32, 33], labels are consistent with the targets. However, in patch level scenario, local features from the same class are inherently diverse, and directly forcing all of them to be similar to each other is not suitable. Similar to [34–36], we adopt the nearest neighbour based 'patch-to-class' distance metric to enforce discrimination. For a training patch x_j^c, its positive nearest neighbour patch set from the same category is denoted as $\mathit{\Gamma}\left(x_j^c\right)$; and its negative nearest neighbour patch set from the categories other than c is denoted as $\bar{\mathit{\Gamma}}\left(x_j^c\right)$. The k-th nearest neighbour in the two sets are represented as $\mathit{\Gamma}_k\left(x_j^c\right)$ and $\bar{\mathit{\Gamma}}_k\left(x_j^c\right)$ respectively.

In the class-specific feature space of class c (transformed by $A^c W$), the feature representation of the k-th positive and negative nearest neighbour patches sets are denoted as $\mathit{\Gamma}_k\left(f_j^c\right) = \mathcal{F}\left(A^c W \mathit{\Gamma}_k\left(x_j^c\right)\right)$ and $\bar{\mathit{\Gamma}}_k\left(f_j^c\right) = \mathcal{F}\left(A^c W \bar{\mathit{\Gamma}}_k\left(x_j^c\right)\right)$

correspondingly. We aim to minimize the distance between each feature to its positive nearest neighbours, while maximize the distance between each feature to its negative nearest neighbours. Furthermore, according to the maximum margin theory in learning, we should focus on the 'hard' training samples. Hence, we develop a 'hinge-loss' like objective function to learn $A^c W$:

$$\begin{aligned} L_{\text{dis}}^{c} = \sum_{j=1}^{N_c} \max\left(\delta + \mathrm{Dis}\left(x_j^{\mathrm{c}}, \varGamma\left(x_j^{\mathrm{c}}\right)\right) - \mathrm{Dis}\left(x_j^{\mathrm{c}}, \bar{\varGamma}\left(x_j^{\mathrm{c}}\right)\right), 0\right) \\ \text{where } \mathrm{Dis}\left(x_j^{\mathrm{c}}, \varGamma\left(x_j^{\mathrm{c}}\right)\right) = \frac{1}{K}\sum_{k=1}^{K} \left\| f_j^{\mathrm{c}} - \varGamma_k\left(f_j^{\mathrm{c}}\right)\right\|_2^2 \\ \mathrm{Dis}\left(x_j^{\mathrm{c}}, \bar{\varGamma}\left(x_j^{\mathrm{c}}\right)\right) = \frac{1}{K}\sum_{k=1}^{K} \left\| f_j^{\mathrm{c}} - \bar{\varGamma}_k\left(f_j^{\mathrm{c}}\right)\right\|_2^2 \end{aligned} \tag{3}$$

in which, δ is the margin, we set it to 1 in our experiments, and K is the number of nearest neighbours in the nearest neighbour patch sets, we fixed it as 5.

However, there are two limitations with the above nearest neighbour based learning method. Firstly, as mentioned in [36], the local patch level nearest neighbour search is likely to be dominated by noisy feature patches. Thus, some of the searched nearest neighbours in Equation 3 might not carry discriminative patterns, consequently the performance will be suppressed. Secondly, it is expensive to search nearest neighbours from the whole patch set. A straight forward solution is applying clustering and using the cluster centroids as the exemplars for each class [36]. However, conventional clustering methods may consider non-informative dominant patterns as inliers of clusters, while treating informative class-specific patterns as outliers. Thus, we propose a method to select discriminative exemplars for each category.

Inspired by the image-level exemplar selection method in [37], we propose an exemplar selection methods that is suitable for patch-level patterns. We firstly define the 'coverage set' of a patch x. Given X as the original global patch set, which is combined with patches densely extracted from all the training images. For each patch $x \in X$, we search its M nearest neighbours from X, and define these M patches as the 'coverage set' of x. Then for each class, we define their exemplar patches as the ones that cannot be easily covered by patches from many classes other than c. To reach this goal, we design a 'patch-to-database' (P2D) distance to measure the discriminative power of a patch x_i^c from class c:

$$\mathrm{P2D}\left(x_j^{\mathrm{c}}\right) = \frac{1}{C-1}\sum_{\bar{c} \neq c} \frac{1}{N_{\bar{c}}} \sum_{n=1}^{N_{\bar{c}}} \left\| x_j^{\mathrm{c}} - x_n^{\bar{\mathrm{c}}} \right\|_2 \tag{4}$$

where $x_n^{\bar{c}}$ is a patch from classes $\bar{c}, \bar{c} \neq c$, $N_{\bar{c}}$ is the number of patches from classes $\bar{c}$ whose coverage sets contain x_j^c, and C is the number of classes. If $\mathrm{P2D}\left(x_j^c\right)$ is small, it means that x_j^c represents a common pattern among many classes, and should be removed, otherwise, it should be kept as a discriminative

Algorithm 1: Discriminative Exemplar Selection

```
Input:
$X$: Global patch set
$X_c$: Patch set of class $c$
$\varepsilon$: Threshold for selecting discriminative exemplars
$M$: Number of patches in each coverage set
Output:
$E_c$: Exemplars of class $c$

1. Calculate the coverage set of each patch from $X$
for $c = 1$ to $C$ do
    2. For each patch from $X_c$, calculate its P2D distance based on Equation 4
    3. Descendingly rank the patches from $X_c$ based on their P2D distances.
    4. Select the top $\varepsilon$ percent ranked patches as the exemplars $E_c$
end
return $E_c$
```

exemplar. For each class, we rank the patches based on their P2D $(\cdot)$ distances descendingly, and select the top 10% of them as discriminative exemplars. The selecting procedures are shown in Algorithm 1. The exemplars will replace the original patch set, and be used to search for the nearest neighbours in Equation 3. Specifically, for each training patch x_j^c, we search its nearest neighbours set $\Gamma\left(x_j^c\right)$ from the exemplars in class c, and search its negative nearest neighbours set $\bar{\Gamma}\left(x_j^c\right)$ from the exemplars belonging to classes other than c.

3.2 DSFL Objective Function and Optimization

Combining the global unsupervised reconstruction term L_u, the shareable constraint term L_{sha} and the discriminative regularization L_{dis}, we write the objective function of DSFL as:

$$\begin{gathered}
\min_{W,\alpha^c} L_u + \gamma \sum_{c=1}^{C} L_{\text{sha}}^c + \eta \sum_{c=1}^{C} L_{\text{dis}}^c \\
\text{where } \ L_u = \sum_{i=1}^{N} \mathcal{L}_u\left(x_i, W\right) + \lambda_1 \sum_{i=1}^{N} \|f_i\|_1 \\
L_{\text{sha}}^c = \sum_{j=1}^{N_c} \mathcal{L}_{\text{sha}}^c\left(x_j^c, A^c W\right) + \lambda_2 \|\alpha^c\|_0 \\
L_{\text{dis}}^c = \sum_{j=1}^{N_c} \max\left(\delta + \text{Dis}\left(x_j^c, \Gamma\left(x_j^c\right)\right) - \text{Dis}\left(x_j^c, \bar{\Gamma}\left(x_j^c\right)\right), 0\right) \\
\text{s.t. } \alpha_d^c \in \{0,1\}, d = 1, ..., D
\end{gathered} \tag{5}$$

In Equation 5, when α^c is fixed, it is convex in W, and when W is fixed, a suboptimal α^c can also be obtained. However, the function cannot be jointly optimized. Thus, we adopt an alternating optimization strategy to iteratively update W and each α^c.

- ***Fix $\boldsymbol{\alpha^c}$ to update $\boldsymbol{W}$*:**

$$\min_{W} \sum_{i=1}^{N} \mathcal{L}_u\left(x_i, W\right) + \lambda_1 \sum_{i=1}^{N} \|f_i\|_1 + \gamma \sum_{c=1}^{C} \sum_{j=1}^{N_c} \mathcal{L}_{\text{sha}}^{c}\left(x_j^c, A^c W\right) + \eta \sum_{c=1}^{C} L_{\text{dis}}^{c} \qquad (6)$$

As mentioned in Section 3.1, $\|f_i\|_1$ degenerates to summation of different dimensions in f_i, thus, Equation 6 can be easily optimized by unconstrained solvers, e.g. L-BFGS.

- ***Fix $\boldsymbol{W}$ to update $\boldsymbol{\alpha^c}$*:**

$$\min_{\alpha^c} \sum_{j=1}^{N_c} \mathcal{L}_{\text{sha}}^{c}\left(x_j^c, A^c W\right) + \lambda_2 \|\alpha^c\|_0 + \eta L_{\text{dis}}^{c} \qquad (7)$$

For the optimization of α^c, we update one α^c each time for the c-th class, and fix $\alpha^{\bar{c}}$ ($\bar{c} \neq c$). To get such binary filter selection indicators, we apply a greedy optimization method. We first set all the elements in α^c as 0, then we search for the single best filter that can minimize Equation 7, and activate that filter by setting the corresponding element in α^c to 1. Afterwards, based on the previously activated filters, we search for next filter that can further minimize the cost function. After several rounds of searching, when the loss $\mathcal{L}_{\text{sha}}^{c}$ is smaller than a threshold, the optimization of α^c terminates, we stop updating α^c, and send the renewed α^c as the input to Equation 6 again to further optimize W.

The learning algorithm and initialization procedure are shown in Algorithm 2. The alternative optimization terminated until the values of both W and α^c converge (takes about 5 rounds).

3.3 Hierarchical Extension of DSFL

DSFL can be easily stacked to extract features at multiple levels. Features at lower level may represent edges and lines, while features at higher level may represent object parts, etc. In our implementation, we stack another layer on the top of the basic DSFL structure[1]. In the first layer DSFL network, 400 dimensional features are learned from 16x16 pixel raw images patches, which are densely extracted from the original/resized images with step size 4. In the second layer, another 400 dimensional feature is learned based on first layer features. To get the inputs for the second layer, we concatenate the first layer features densely extracted within 32x32 image areas. We further process PCA to reduce the dimension to 300 and send it to the second layer. Finally, we combine the features learned from both layers as our DSFL feature.

[1] Adding more layers can slightly improve the performance, but the computational cost is high, thus we apply two layer DSFL to reach a compromise.

Algorithm 2: DSFL: Discriminative and Shareable Feature Learning

```
Input:
$x_i$: Unlabelled training patch
$x_j^c$: Image-level labelled training patch from class $c$
$D$: Number of filters in the global filter bank
$\gamma$, $\eta$, $\lambda_1$, $\lambda_2$: Trade off parameters for controlling weight of shareable term,
discriminative term, and sparsity
Output:
$W$: Global filter bank (feature transformation matrix)

1. Initialize $\alpha^c = \mathbf{0}^T$
2. Set $W$ as a random number $D \times D_0$ matrix
3. Learn $W$ with only unsupervised term $L_u$ as the initialized $W$ to the DSFL
4. Select exemplars for each class based on Equation 4
5. Search the positive and negative nearest neighbour exemplar sets for each $x_j^c$
while $W$ and $\alpha^c$ not converge do
    for $c = 1$ to $C$ do
        6. Fix $W$ and solve Equation 7 by updating $\alpha^c$
    end
    7. Fix $\alpha^c, c = 1, ..., C$ and solve Equation 6 by updating $W$
end
return $W$
```

4 Experiments and Analysis

4.1 Datasets and Experiment Settings

We tested our DSFL method on three widely used scene image classification datasets: Scene 15 [38], UIUC Sports [39], and MIT 67 Indoor [40]. In order to make fair comparisons with other types of features, we only used gray scale information for all these datasets.

We tested on all the three datasets with the most standard settings: on Scene 15, we randomly selected 100 images per category for training, and the rest for testing; on UIUC sports, we randomly selected 70 images per class as training images, and 60 images per class as testing images; on MIT 67 Indoor, we followed the original splits in [40], which used around 80 training images and 20 testing images for each category. For UIUC sports and MIT 67 Indoor, since the resolution of the original images are too high for learning local features efficiently, we resized them to have maximum 300 pixels along the smaller axis. For Scene 15 and UIUC sports, we randomly split the training and testing dataset for 5 times. The average accuracy numbers over these 5 rounds are reported for comparison. For all the local features, we densely extracted features from six scales with rescaling factors $2^{-i/2}, i = 0, 1, ..., 5$. Specifically, RICA [4] and DSFL features were extracted with step size 3 for the first layer, and step size 6 for the second layer; SIFT features [1] were extracted from 16x16 patches with stride 3; HOG2x2 features [41] were extracted based on cells of size 8x8, and the stride is 1 cell; LBP features [13] were extracted from cells of size 8x8.

Mehods	Scene 15	UIUC Sports	MIT 67 Indoor
GIST [12]	73.28%	-	22.00%
CENTRIST [42]	83.10%	78.50%	36.90%
SIFT [1]	82.06%	85.12%	45.86%
HOG2x2 [41]	81.58%	83.96%	43.76%
LBP [13]	82.95%	80.04%	39.25%
RICA [4]	79.85%	82.14%	47.89%
DSFL	84.19%	86.45%	52.24%
DeCAF [3, 10]	87.99%	93.96%	58.52%
SIFT [1] + DeCAF [3, 10]	89.90%	95.05%	70.51%
DSFL + DeCAF [3, 10]	92.81%	96.78%	76.23%

Table 1. Comparison results between our feature and other features. (DeCAF is the feature learned by the deep ConvNets pre-trained on ImageNet)

For each training image, we randomly picked 400 patches (200 for MIT Indoor), and used them as training data to learn W. In the objective function Equation 5, the value of margin δ was fixed as 1, and we sequentially learnt the weight parameters λ_1, λ_2, γ and η by cross validation. In Algorithm 1, the threshold of exemplar selection ε was set to 10%, and the coverage set size M was set to 10. In Algorithm 2, the maximum number of iterations of updating W and α_c was set to 5.

We tested our local features based on the LLC framework [43], which used locality-constrained linear coding to encode local features, and performed max-pooling and linear-SVM afterwards. The size of the codebook was fixed as 2000, and each image was divided into 1x1, 2x2, and 4x4 spatial pooling regions [38]. We've also tested on other frameworks with different coding strategies (e.g. vector quantization) and pooling schemes (e.g. average pooling), our DSFL can consistently outperform traditional local features.

4.2 Comparison with Other Features

As shown in Table 1, we compared our DSFL with popular features which have shown good performance on scene images classification: SIFT [1], GIST [12], CENTRIST [42], and HoG [2, 44], LBP [13]. Our DSFL feature is able to outperform all of the hand crafted features. We also compared our DSFL with RICA [4], which is the baseline unsupervised feature learning method without encoding any discriminative or class-specific information. As shown in Table 1, our method consistently and significantly outperforms RICA. We've also tested the performance of only using the features learned by the first partially connected layer, and for the three datasets, the results were 82.61%, 83.92%, and 47.16%, which are less powerful than the two layer features.

In Table 1, the DeCAF feature [10] is an implementation of the 7 layer ConvNets [3]. Here we used the 6-th layer DeCAF feature. According to [10, 20], empirically the 6-th layer feature will lead to better results than the 7-th layer

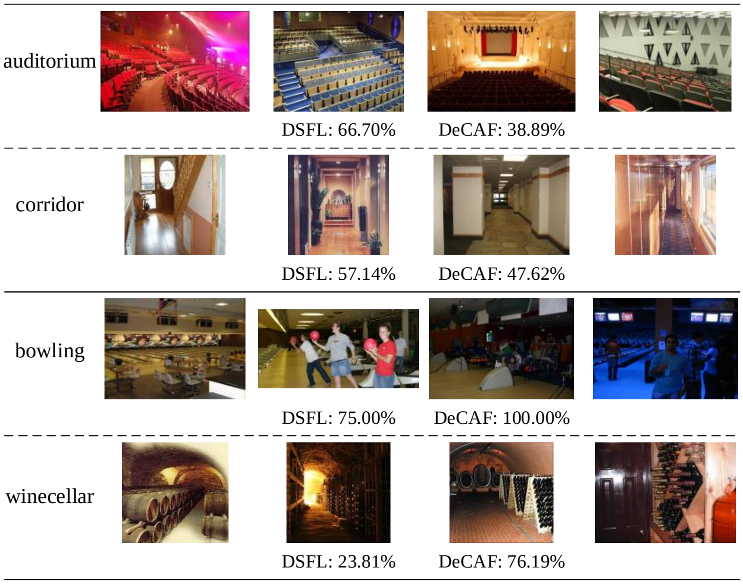


Fig. 2. Comparison results on MIT 67 Indoor. The first two rows show the two categories on which DSFL works better than DeCAF, the last two rows show the classes that are better represented by DeCAF. DSFL and DeCAF are complimentary. Combining them can result in better results for scene classification.

feature. On the three datasets, we also tested with the 7-th layer feature, and got 87.35%, 93.44%, and 58.27% respectively. Thus, the 6-th layer DeCAF features were used for evaluation. Although this pre-trained DeCAF feature is very powerful, yet directly comparing our feature with it is not fair. We do not utilize the huge amount of image data from ImageNet [11], we haven't used color information, and we focus on local feature representation rather than global image representation. The ConvNets was trained on the ImageNet with a large amount of object images. We suppose the features learned from these two frameworks should be complementary. In Fig. 2, we tested on MIT 67 to show the complementary effect. In the first two rows, our DSFL worked better than DeCAF, and we show the testing images which were correctly classified by DSFL, but wrongly classified by DeCAF. In the last two rows, DeCAF outperformed DSFL, and we show the testing images which our DSFL failed to recognize but DeCAF could. To quantitatively analyze the complementation effect, we combined our DSFL with the DeCAF feature. As shown in the last row of Table 1, we are able to get much better performance than purely using the powerful ConvNets features and produce the state-of-the-art performance. We also tested the combination of SIFT and DeCAF. The accuracy numbers are not as good as those of the combination of DSFL and DeCAF, which indicates that our DSFL can learn more effective complementary information by considering data adaptive information. The traditional hand-crafted features such as SIFT usually extracted 'garbor-like' features, most of which can be learned by the lower levels in ConvNets. However, ConvNets adopts backpropagation for optimization based on huge training datasets, the bottom layers of the network were usually not well trained. In contrast, we explicitly used supervised information to train bottom layer features. Our method is more suitable for relatively small datasets, as evidenced by the experimental results, while previous attempts on trying to train a CNN classifier on small datasets usually failed. So these two lines of works are expected to be complimentary.

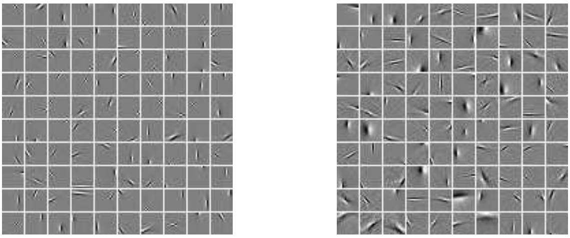
Mehods	Scene 15	UIUC Sports	MIT 67 Indoor
ROI + GIST [40]	-	-	26.50%
DPM [45]	-	-	30.40%
Object Bank [46]	80.90%	76.30%	37.60%
Discriminative Patches [47]	-	-	38.10%
LDC [36]	80.30%	-	43.53%
macrofeatures [48]	84.30%	-	-
Visual Concepts + 3 combined features [25]	83.40%	84.80%	46.40%
MMDL + 5 combined features [49]	86.35%	**88.47%**	50.15%
Discriminative Part Detector [27]	86.00%	86.40%	51.40%
LScSPM [50]	**89.78%**	85.27%	-
IFV [28]	-	-	60.77%
MLrep + IFV [26]	-	-	**66.87%**
DSFL + DeCAF [3,10]	**92.81%**	**96.78%**	**76.23%**

Table 2. Comparison Results of our method and other popular methods on Scene 15, UIUC sports, and MIT 67 Indoor.

We also compared our method (combining DSFL and DeCAF) with other methods applied on these three scene datasets. As shown in Table 2, our method achieved the highest accuracy on all of the three datasets. Note that Visual Elements [26] utilized numerous patches extracted at scales ranging from 80x80 to the full image size, and the patches were represented by standard HOG [2] plus a 8x8 color image in L*a*b space, and very high dimensional IFV [28] features. While MMDL [49] combined 5 types of features on 3 scales. Furthermore, most of the previous works were based on hand-crafted local feature descriptions, which means that our learned DSFL features can be combined with them to achieve better results. For example, LScSPM [50] focused on coding, which can be used to encode our DSFL features.

4.3 Analysis of the effect of different components

In this section, we aim to compare our shareable and discriminative learning method to the baseline without encoding such information, which is equivalent to the RICA method in [4]. We first show the visualization of the filters learned from UIUC Sports in Fig. 3(a) and Fig. 3(b). We can see that our DSFL is able to capture more sharply localized patterns, corresponding to more class-specific visual information.



(a) RICA

(b) DSFL

Fig. 3. Visualization of the filters learned by RICA and our DSFL on the UIUC Sports dataset.

Effect of learning shareable filter bank We tested the DSFL with or without the feature sharing terms, and got the intermediate results in Table 3. The first row of the table shows the baseline unsupervised RICA features learned by solving Equation 1. In the second row, $L_u + L_{\text{sha}}$ corresponds to features learned with Equation 2. The improvement in accuracy shows that learning shareable features is effective for classification. However, if we removed the global reconstruction error term $\mathcal{L}_u$ and only kept the shareable terms, as shown in the third row, the performance dramatically dropped.

Effect of Discriminative Regularization and Exemplar Selection According to the fourth row and the fifth row of Table 3, we can find that if we didn't select exemplars for learning, we could not achieve much improvement because noisy training examples might overwhelm the useful discriminative patterns. However, once we learned using selected exemplars, our method could achieve significant improvement in classification accuracy. This shows that discriminative exemplar selection is critical in our learning framework.

Furthermore, it's obvious that only using 10% of the whole patch set dramatically increased the efficiency of nearest neighbour search afterwards. Thus, our exemplar selection method is both effective and efficient.

Effect of the Size of Filter Bank To further analyze the influence caused by the size of filters, we test on Scene 15 dataset with 128, 256, 512, 1024, and 2048 filters for the DSFL. The results are shown in Fig. 4. At the beginning, when the size is small, the learned features are relatively weak. When the number of

Mehods	Scene 15	UIUC Sports	MIT 67 Indoor
L_u (RICA [4])	79.85%	82.14%	47.89%
$L_u + L_{\text{sha}}$	82.01%	83.67%	49.70%
L_{sha}	72.69%	72.52%	24.12%
$L_u + L_{\text{sha}} + L_{\text{dis}}$ (without Exemplar)	82.50%	83.43%	51.28%
$L_u + L_{\text{sha}} + L_{\text{dis}}$ (Full DSFL)	84.19%	86.45%	52.24%

Table 3. Analysis of the effect of each components

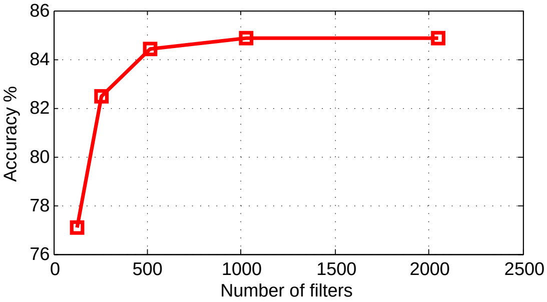


Fig. 4. Results of varying number of filters in Scene 15

filters increases, and W becomes over-complete, the performance is substantially improved. Thus, learning over-complete filter bank does help to obtain better feature representation because the resulting filter bank captures more information. However, when the number of filters further increases, the performance does not change much, while the learning process will be extremely slow. In our experiment, we use 400 as a compromise between efficiency and accuracy.

5 Conclusion

In this paper, we propose a weakly supervised feature learning method, called DSFL, to learn a discriminative and shareable filter bank to transform local image patches into features. In our DSFL method, we learn a flexible number of shared filters to represent common patterns shared across different categories. To enhance the discriminative power, we force the features from the same class to be locally similar, while features from different classes to be separable. We test our method on three widely used scene image classification benchmark datasets, and the results consistently show that our learned features can outperform most of the existing features. By combining our features with the ConvNets features pre-trained on ImageNet, we can greatly enhance the representation, and achieve state-of-the-art scene classification results. In the future, we will integrate our learning method with deeper learning structure to extract multi-level features for more effective classification.

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2) (2004) 91–110
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005) 886–893
3. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012) 1106–1114
4. Le, Q.V., Karpenko, A., Ngiam, J., Ng, A.Y.: Ica with reconstruction cost for efficient overcomplete feature learning. In: NIPS. (2011) 1017–1025
5. Zou, W.Y., Zhu, S.Y., Ng, A.Y., Yu, K.: Deep learning of invariant features via simulated fixations in video. In: NIPS. (2012) 3212–3220
6. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural computation (2006)
7. Coates, A., Lee, H., Ng, A.Y.: An analysis of single-layer networks in unsupervised feature learning. In: International Conference on Artificial Intelligence and Statistics. (2011) 215–223
8. Sohn, K., Jung, D.Y., Lee, H., Hero, A.O.: Efficient learning of sparse, distributed, convolutional feature representations for object recognition. In: ICCV. (2011) 2643–2650
9. Zuo, Z., Wang, G.: Learning discriminative hierarchical features for object recognition. Signal Processing Letters **21**(9) (2014) 1159–1163
10. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. In: ICML. (2014) 647–655
11. Deng, J., Berg, A.C., Li, K., Fei-Fei, L.: What does classifying more than 10,000 image categories tell us? In: ECCV. (2010) 71–84
12. Oliva, A., Torralba, A.: Building the gist of a scene: The role of global image features in recognition. Progress in brain research **155** (2006) 23–36
13. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. Pattern Analysis and Machine Intelligence, IEEE Transactions on **24**(7) (2002) 971–987
14. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multistage architecture for object recognition? In: ICCV. (2009) 2146–2153
15. Taylor, G.W., Fergus, R., LeCun, Y., Bregler, C.: Convolutional learning of spatio-temporal features. In: ECCV. (2010) 140–153
16. Le, Q.V., Ranzato, M.A., Monga, R., Devin, M., Chen, K., Corrado, G.S., Dean, J., Ng, A.Y.: Building high-level features using large scale unsupervised learning. In: ICML. (2012)
17. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: CVPR. (2011) 3361–3368
18. Wang, A., Lu, J., Wang, G., Cai, J., Cham, T.J.: Multi-modal unsupervised feature learning for rgb-d scene labeling. In: ECCV. (2014)
19. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)
20. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv preprint arXiv:1311.2524 (2013)

21. Jiang, Z., Lin, Z., Davis, L.S.: Learning a discriminative dictionary for sparse coding via label consistent k-svd. In: CVPR. (2011) 1697–1704
22. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Supervised dictionary learning. In: NIPS. (2008)
23. Yang, M., Zhang, L., Feng, X., Zhang, D.: Fisher discrimination dictionary learning for sparse representation. In: ICCV. (2011) 543–550
24. Kong, S., Wang, D.: A dictionary learning approach for classification: separating the particularity and the commonality. In: ECCV. (2012)
25. Li, Q., Wu, J., Tu, Z.: Harvesting mid-level visual concepts from large-scale internet images. In: CVPR. (2013)
26. Doersch, C., Gupta, A., Efros, A.A.: Mid-level visual element discovery as discriminative mode seeking. In: NIPS. (2013) 494–502
27. Sun, J., Ponce, J., et al.: Learning discriminative part detectors for image classification and cosegmentation. In: ICCV. (2013)
28. Juneja, M., Vedaldi, A., Jawahar, C., Zisserman, A.: Blocks that shout: Distinctive parts for scene classification. In: CVPR. (2013) 923–930
29. Song, H.O., Zickler, S., Althoff, T., Girshick, R., Fritz, M., Geyer, C., Felzenszwalb, P., Darrell, T.: Sparselet models for efficient multiclass object detection. In: ECCV. (2012) 802–815
30. Song, H.O., Darrell, T., Girshick, R.B.: Discriminatively activated sparselets. In: ICML. (2013) 196–204
31. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786) (2006) 504–507
32. Wang, G., Forsyth, D., Hoiem, D.: Improved object categorization and detection using comparative object similarity. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35**(10) (2013) 2442–2453
33. Wang, Z., Gao, S., Chia, L.T.: Learning class-to-image distance via large margin and l1-norm regularization. In: ECCV. (2012)
34. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR. (2008) 1–8
35. McCann, S., Lowe, D.G.: Local naive bayes nearest neighbor for image classification. In: CVPR. (2012) 3650–3656
36. Wang, Z., Feng, J., Yan, S., Xi, H.: Linear distance coding for image classification. Image Processing, IEEE Transactions on **22**(2) (2013) 537–548
37. Yao, B., Fei-Fei, L.: Action recognition with exemplar based 2.5 d graph matching. In: ECCV. (2012)
38. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR. Volume 2. (2006) 2169–2178
39. Li, L.J., Fei-Fei, L.: What, where and who? classifying events by scene and object recognition. In: ICCV. (2007)
40. Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: CVPR. (2009)
41. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: CVPR. (2010) 3485–3492
42. Wu, J., Rehg, J.M.: Centrist: A visual descriptor for scene categorization. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33**(8) (2011) 1489–1501
43. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR. (2010) 3360–3367
44. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. Pattern Analysis and Machine Intelligence, IEEE Transactions on **32**(9) (2010) 1627–1645

45. Pandey, M., Lazebnik, S.: Scene recognition and weakly supervised object localization with deformable part-based models. In: ICCV. (2011) 1307–1314
46. Li, L.J., Su, H., Fei-Fei, L., Xing, E.P.: Object bank: A high-level image representation for scene classification & semantic feature sparsification. In: NIPS. (2010)
47. Singh, S., Gupta, A., Efros, A.A.: Unsupervised discovery of mid-level discriminative patches. In: ECCV. (2012) 73–86
48. Boureau, Y.L., F. Bach, Y.L., Ponce, J.: Learning mid-level features for recognition. In: CVPR. (2010)
49. Wang, X., Wang, B., Bai, X., Liu, W., Tu, Z.: Max-margin multiple-instance dictionary learning. In: ICML. (2013)
50. Gao, S., Tsang, I.H., Chia, L.T.: Laplacian sparse coding, hypergraph laplacian sparse coding, and applications. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35**(1) (2013) 92–104