Learning Multi-instance Deep Discriminative Patterns for Image Classification

Peng Tang, *Student Member, IEEE*, Xinggang Wang, Bin Feng, and Wenyu Liu, *Senior Member, IEEE*

***Abstract*—Finding an effective and efficient representation is very important for image classification. The most common approach is to extract a set of local descriptors, and then aggregate them into a high-dimensional, more semantic feature vector, like unsupervised Bag-of-Features (BoF) and weakly supervised part-based models. The later one is usually more discriminative than the former due to the use of information from image labels. In this work, we propose a weakly supervised strategy that using Multi-Instance Learning (MIL) to learn discriminative patterns for image representation. Specially, we extend traditional multi-instance methods to explicitly learn more than one patterns in positive class, and find the "most positive" instance for each pattern. Furthermore, as the positiveness of instance is treated as a continuous variable, we can use stochastic gradient decent (SGD) to maximize the margin between different patterns meanwhile considering MIL constraints. To make the learned patterns more discriminative, local descriptors extracted by Deep Convolutional Neural Networks (DCNN) are chosen instead of hand-crafted descriptors. Some experimental results are reported on several widely used benchmarks (Action 40, Caltech 101, Scene 15, MIT-indoor, SUN 397), showing that our method can achieve very remarkable performance.**

***Index Terms*—Image classification, discriminative patterns, multi-instance learning, stochastic gradient decent, deep convolutional neural networks.**

I. Introduction

IMAGE classification is one of the most important tasks in computer vision. It needs to learn a classifier to determine whether a given image belongs to a particular object class or semantic category. Although many methods have been put forward for it, it is still a very challenging problem. The reasons are that, (1) images usually have complex layout and cluttered background, (2) objects in images usually have different scales, different positions, and different appearance, etc. Thus, even images from same class have great variation. So it is necessary to explore robust and powerful representation of images to recognize it.

The Bag-of-Features (BoF) [1] is a very popular representation in image classification community. Simply put, the BoF model represents an image as orderless collections of local descriptors, such as SIFT [2] or HOG [3]. It uses unsupervised clustering methods like k-means to learn the codebook of local descriptors, *i.e.*, a set of "visual words", during training. After words are got, image can be represented using hard or soft assignment ways [4]–[6]. To take the spatial layout of images into account, which is very important for image classification, a very simple but useful strategy Spatial Pyramid Matching (SPM) was presented [7], and has made many remarkable achievements for image classification.

But most of earlier BoF models learn image representations through an unsupervised way, which lead to great losses of information without considering image labels. So some weakly supervised strategies are proposed to discover the discriminative patterns. Inspired by the Deformable Part-based Model (DPM) in object detection [8], some part-based models have been proposed as mid-level visual representation for image classification [9]–[12]. Images have many parts, which can respond to objects (*e.g.*, a face, a car) or a parts of objects (*e.g.*, a wheel of the car). Each image is composed by some of these parts[1], and usually images in the same class have similar parts. These part-based models find a set of important parts for each class as the discriminative patterns to represent images, and have achieved many inspiring results on image classification. Our method also uses weakly supervised way to learn patterns.

In spite of the great success of part-based models, there are still many limitations in it. The first limitation is that these models usually use hand-crafted local descriptors to represent patches in images, like SIFT [2] and HOG [3], which may be very noisy because it is hard to distinguish different objects exactly using these descriptors. To address this problem, in this work, we argue that automatically learned features, such as features learned by Deep Convolutional Neural Networks (DCNN) [13], can eliminate this problem. Due to large and diverse datasets like ImageNet [14] are available, DCNN have achieved many breakthrough accuracies for image classification [13]. And using fine trained DCNN features as a universal image representation for recognition has also been turned out to be effective [15]–[18]. In this paper, we extract features for each image patch using DCNN, and then use these deep features to learn our discriminative patterns.

Another limitation is that for each class, the amount of learned parts is very large (generally hundreds even thousands per class), which is very costly for both part learning and image representation. Aiming at this problem, we propose to learn Multi-instance Deep Discriminative Patterns (MiDDP). Our method is inspired by the Max-margin Multi-instance Dictionary Learning (MMDL) framework [19]. For each class, we use Multi-Instance Learning (MIL) to learn some discrim-

This work was supported in part by the National Natural Science Foundation of China under Grant 61572207, Grant 61503145, and Grant 61502188, as well as the CAST Young Talents Support Program.

P. Tang, X. Wang, B. Feng, and W. Liu are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: {pengtang, xgwang, fengbin, liuwy}@hust.edu.cn). Corresponding author: W. Liu.

[1]"Words", "parts", and "patterns" are interchangeable. "Patterns" is chosen in our paper to represent them.

inative patterns, by regarding images in that class as positive and rest as negative. And inspired by the Relaxed Multi-Instance Support Vector Machine (RMI-SVM) [20] strategy, we relax MIL constraints using the max model instead of the Noisy-OR (NOR) model in RMI-SVM. Generally MIL only focuses on single-positive pattern problem, but when the situation is that there are multi-positive patterns, it is necessary to extend MIL to fit that situation. Obviously we need finding a way to deal with this problem for image classification as there are always more than one pattern in an image. In MiDDP, we maximize the margins between each pattern considering MIL constraints. But the labels of patches in positive images are hard to gotten directly, so we treat it as latent values, using the Latent SVM (LSVM) [8] to train the model. Using our MiDDP strategy, only nine patterns per class are necessary, which is much smaller than other methods.

After relaxing MIL constraints, we can use Stochastic Gradient Descend (SGD) algorithm to efficiently optimize the MiDDP model. But the data is highly unbalanced during training because there is only one class in positive set and are many classes in negative set. Employing hard-negative mining [3], [8] as a part of the learning process is one of the methods to deal with this problem. But it may make the training procedure more complicated. In this paper, following the exemplar SVM training procedure in [21], we use a re-sampling strategy to train our model. Using the re-sampling SGD, the training procedure of our method is more tractable and very simple to implement.

To summarize, the main contributions of our work are as follows.

- A novel method to learn discriminative patterns in a weakly supervised way as semantic dictionary for image classification. The proposed image classification method obtains state-of-the-art performance on various benchmarks.
- An effective and robust optimization method inspired by PEGASOS [22] via stochastic gradient descent for a class of multi-instance learning problem which has multi-positive patterns per class.

Our approach can be proven both effective and efficient. The MiDDP model only needs very small numbers of patterns per class, so it is easy to learn patterns and encode images; and the training process via SGD is very easy to perform; meanwhile it is very discriminative, which achieves very competitive performance on some widely used benchmarks for image classification.

The rest of this paper is organized as follows. In Section II, related work is introduced. In Section III, detail of the method learning discriminative patterns is described. In Section IV, the method described in Section III is extended to represent images. In Section V, some experimental results and analysis are reported. In Section VI, some discussions are reported. In Section VII, some conclusions are given.

II. Related Work

Learning intermediate representations is very popular for image classification. The traditional BoF model [1] usually uses unsupervised learning to learn patterns in images, such as k-means. And strongly supervised methods like Attributes [23]–[25], Poselets [26], and Object Bank [27] have shown more discriminative than unsupervised methods. But they need additional human annotations not only the image level labels to learn patterns, which are very costly and non-trivial to get.

Recently, weakly supervised methods have shown great success for representing images [9]–[12], [28]–[31]. It only uses image level labels to learn patterns, which is easier to get and more discriminative than unsupervised ways. There are many attempts to learn discriminative patterns using a weakly supervised way. [29] learns hundreds of discriminative patches for each class using linear SVM, and then selects some discriminative ones according to their importance. Some part-based models [9]–[12] use heuristic methods to find a collection of parts for each class, and each part scores higher on that class than on others, therefore it is discriminative. [30] chooses association rule mining to find patterns. Our work also along the line of weakly supervised learning, explore discriminative patterns for each class. But the difference is that, our method uses the MIL strategy, which is quite different from their methods, and can find some most representative patterns for each class compared with other classes. This will contribute to better results while require fewer patterns.

Inspired by great success of DCNN for image classification [13], many works [15]–[18] attempt to use DCNN models trained on large scale dataset like ImageNet [14] as the feature extractor. [17] extracts DCNN activations from the fully connected 7-th layer (fc7) as local descriptor at multiple scale levels, with the orderless VLAD pooling of these activations for scene classification and image retrieval. [18] uses output from the fully connected 8-th layer (fc8) at multiple scales, along with the semantic Fisher Vector to classify scene images. [32] uses output from Place DCNN [33] fc7 to learn discriminative parts jointly with the image classifiers. All these works use activation from DCNN as off-the-shelf features. Our work also follows these strategies, using output from the ImageNet fc7 at four different scales as local descriptor, which has shown very effective for image classification.

Our work is based on MIL, which is proposed for drug activity prediction [34]. There are many efforts to develop MIL algorithms for its usefulness in machine learning and computer vision. For example, the mi-SVM and MI-SVM [35] train SVM for MIL. And very recently, [36] models MIL in the deep learning framework and uses it for image classification and auto-annotation. The RMI-SVM [20] relaxes MIL constraints into convex program meanwhile uses it for object detection. Our work follows the path of MMDL [19], which maximizes the instance-level margin to learn the dictionary. However, the optimization problem defined in MMDL is very hard to steer. Here, we relax MIL constraints into convex program to find multi-positive patterns per class, which is very different from normal MIL methods only considering single-positive pattern. And we use SGD to optimize it, which is easier to implement and more tractable. The max model is chosen in our MiDDP framework, so we maximize the instance-level margins, at the same time, find "most positive" instance for each pattern, which is quite different from former works.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2016.2642781, IEEE Transactions on Image Processing

III. Learning Multi-instance Discriminative Patterns

A. MIL with Single-positive Pattern

To formulate our MiDDP strategy, we first present some notations of MIL. In MIL, n bags $X = \{X_1, X_2, ..., X_n\}$ and its labels $Y = \{Y_1, Y_2, ..., Y_n\}$ are given, where each bag X_i is composed of m_i instances $X_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, ..., \mathbf{x}_{im_i}\}$ and each label $Y_i \in \{0, 1\}$. Each instance $\mathbf{x}_{ij}$ is denoted by a d-dimensional vector $\mathbf{x}_{ij} \in \mathbb{R}^{d\times 1}$, and is also associated with the instance label $y_{ij} \in \{0, 1\}$, where the 0 and 1 means negative and positive respectively. It is natural that in MIL, there are some constraints:

- If a bag is negative, all instances in the bag are negative, *i.e.*, if $Y_i = 0$, then $y_{ij} = 0$ for all $j \in \{1, 2, ..., m_i\}$.
- If a bag is positive, at least one instance in the bag is positive, *i.e.*, if $Y_i = 1$, then at least one $y_{ij} = 1$ for all $j \in \{1, 2, ..., m_i\}$.

As in RMI-SVM [20], we relax the the instance label y_{ij} to be the probability of $\mathbf{x}_{ij}$ being positive, denoted as p_{ij}. Without loss of generality, we simple adopt a linear model $f(x) = \mathbf{w}^T\mathbf{x}$ as the instance model. So p_{ij} can be formulated using the logistic function

$$p_{ij} = Pr(y_{ij} = 1|x_{ij}; \mathbf{w}) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}_{ij}}}, \tag{1}$$

where $\mathbf{w}$ is the weight vector of the linear model.

RMI-SVM [20] uses Noisy-OR (NOR) model to represent the probability of a bag being positive, as in Eq. (2).

$$P_i = Pr(Y_i = 1|X_i; \mathbf{w}) = 1 - \prod_{j=1}^{m_i}(1 - p_{ij}). \tag{2}$$

But we can find that there may be a drawback in Eq. (2). That is, when all p_{ij} of bag X_i are modest, and the number of instances m_i is large, *e.g.*, all $p_{ij} = 0.1$ and $m_i = 100$, the P_i will close to 1, there may be some ambiguities. We may distinguish the bag as positive because P_i closes to 1, but it may be more reasonable that the bag is negative because all instances have low probability to be positive. To deal with this problem, we use the max model instead of the NOR model as follows.

$$P_i = Pr(Y_i = 1|X_i; \mathbf{w}) = \max_j p_{ij}. \tag{3}$$

As we can see, when X_i is negative, all p_{ij} should close to 0, then the P_i will close to 0 according to Eq. (3); and when X_i is positive, at least one p_{ij} close to 1, so the resulting P_i will close to 1.

After getting the probability P_i, MIL constraints can be relaxed by introducing the cross-entropy,

$$L_{bag_i} = -\{Y_i \log P_i + (1 - Y_i) \log (1 - P_i)\}. \tag{4}$$

Obviously we can implement the MIL through minimizing the bag loss L_{bag_i}. It can be observed that the relaxed MIL is somewhat similar to the MI-SVM [35], which finds "most positive" instance.

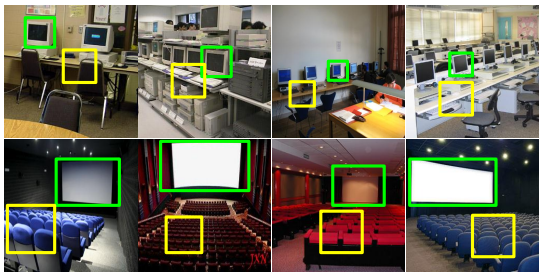


Fig. 1. There are multiple patterns per class, *e.g.*, computer monitors (green rectangles) and desks (yellow rectangles) in computer room (the first row), movie screens (green rectangles) and seats (yellow rectangles) in movie theater (the second row).

B. MIL with Multi-positive Patterns

The definition above can perform well when there is only single-positive pattern. But it is not powerful enough when it encounters the situation that the positive bag has complicated layout and more than one pattern, like images in the scene, as shown in Fig. 1. So the instance label may belong to different patterns, *i.e.*, $y_{ij} \in \{0, 1, 2, ..., K\}$, where K is the number of positive patterns, and $y_{ij} = 0$ means that it is the negative instance. To extend this model, we reformulate the Eq. (1) using the softmax function, just as follows.

$$p_{ijk} = Pr(y_{ij} = k|\mathbf{x}_{ij}; \mathbf{W}) = \frac{e^{\mathbf{w}_k^T\mathbf{x}_{ij}}}{\sum_{m=0}^{K} e^{\mathbf{w}_m^T\mathbf{x}_{ij}}}, \tag{5}$$

where the $\mathbf{W} = [\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K]$ is the weight matrix of the $\{K + 1\}$-class linear classifier, and its component $\mathbf{w}_k \in \mathbb{R}^{d\times 1}, k \in \{1, 2, ..., K\}$ is the $\{k+1\}$-th weight vector of that classifier, *i.e.*, the weight vector of k-th positive pattern, and $\mathbf{w}_0$ is the weight vector of negative pattern.[2]

We can also use Eq. (3) to formulate the P_i for the multi-positive patterns problem, but it may be insufficient because only the most positive pattern will be found. Our goal is to find most positive instances of every pattern in positive bags, so we modify the Eq. (3) to adapt the goal.

$$P_i = Pr(Y_i = 1|X_i; \mathbf{W}) = \prod_{k=1}^{K} p_{ij_kk}, \tag{6}$$

where

$$j_k = arg \max_j \ p_{ijk}. \tag{7}$$

So only when all p_{ij_kk} close to 1, the P_i will close to 1, *i.e.*, only when the bag X_i contains all K different positive patterns, its label Y_i will equal to 1. Substituting it into Eq. (4), the bag loss can be written as

$$\begin{aligned} L_{bag_i} &= -\{Y_i \log P_i + (1 - Y_i) \log (1 - P_i)\} \\ &= -\{Y_i \log (\prod_{k=1}^{K} p_{ij_kk}) + (1 - Y_i) \log (1 - \prod_{k=1}^{K} p_{ij_kk})\}. \end{aligned} \tag{8}$$

[2]In this paper, for each class, patterns are defined as the weight matrix $\mathbf{W}$, which can discover some most semantic patches, *e.g.*, objects.

Note that when $Y_i = 1$, the Eq. (8) will be $-\log(\prod_{k=1}^{K} p_{ij_k k})$, so only when all $p_{ij_k k} = 1$ the loss will equal to 0. But when $Y_i = 0$, the Eq. (8) will be $-\log(1 - \prod_{k=1}^{K} p_{ij_k k})$, and there may exist some problems. For example, when only one $p_{ij_k k}$ equals to 0 and others equal to 1, the loss will equal to 0. It means that some patterns expected to be positive also exist in negative images (patterns with large $p_{ij_k k}$ when $Y_i = 0$), *i.e.*, these patterns are not really positive, except the one with small $p_{ij_k k}$. So it may fall back to the single-positive pattern problem. Obviously it is far from our expectation that multiple patterns should only exist in positive images. And when all the values of $p_{ij_k k}$ are modest, *e.g.*, all $p_{ij_k k} = 0.6$ and $K = 8$, then $P_i = 0.6^8 \approx 0.02$, so the bag loss will very close to 0. This means that the positive patterns are not discriminative enough comparing with the negative patterns.

To deal with the problems described above, we reformulate the bag loss to Eq. (9).

$$\begin{aligned} L_{bag_i} &= -\{Y_i \log(\prod_{k=1}^{K} p_{ij_k k}) + (1 - Y_i) \log \prod_{k=1}^{K} (1 - p_{ij_k k})\} \\ &= -\sum_{k=1}^{K} \{Y_i \log p_{ij_k k} + (1 - Y_i) \log(1 - p_{ij_k k})\}. \end{aligned} \tag{9}$$

It is obvious that only when all $p_{ij_k k}$ in positive bags equal to 1 and all $p_{ij_k k}$ in negative bags equal to 0, the bag loss L_{bag_i} will have the least value 0. So we can find more discriminative positive patterns.

From the above formulations, we can observe that the single-positive pattern can be regarded as a special case of multi-positive patterns. Since when $K = 1$, the weight vector $\mathbf{w}_1$ will be opposite to $\mathbf{w}_0$, *i.e.*, $\mathbf{w} = \mathbf{w}_1 = -\mathbf{w}_0$. So the Eq. (5) can be written as

$$\begin{aligned} p_{ij1} = Pr(y_{ij} = 1 | \mathbf{x}_{ij}; \mathbf{w}) &= \frac{e^{\mathbf{w}^T \mathbf{x}_{ij}}}{e^{\mathbf{w}^T \mathbf{x}_{ij}} + e^{-\mathbf{w}^T \mathbf{x}_{ij}}} \\ &= \frac{1}{1 + e^{-2\mathbf{w}^T \mathbf{x}_{ij}}}, \end{aligned} \tag{10}$$

which is very similar to Eq. (1) except a constant 2 in front of the $\mathbf{w}$. And the P_i and L_{bag_i} are exactly the same as single-positive pattern.

C. Formulation of Multi-instance Discriminative Patterns

The goal of our work is to find the discriminative patterns using relaxed MIL for multi-positive patterns described above, which we call multi-instance discriminative patterns (MiDP). In MiDP, we maximize the margins between different patterns considering the constraints of MIL, just as the objective function shown in Eq. (11).

$$\min_{\mathbf{W}} \ \frac{\lambda}{2} \sum_{k=0}^{K} \|\mathbf{w}_k\|_2^2 + \frac{\beta}{n} \sum_{i=1}^{n} L_{bag_i} + \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i} \sum_{j=1}^{m_i} L_{ins_{ij}}, \tag{11}$$

where the first term in Eq. (11) is for the regularization; L_{bag_i} and $L_{ins_{ij}}$ are the cross-entropy and hinge-loss respectively, which correspond to the loss functions for i-th bag prediction and j-th instance prediction in bag i. The bag loss is computed via Eq. (9), and to formulate the instance loss, the latent variables $z_{ij} \in \{0, 1, 2, ..., K\}$ are introduced as the instance labels, as shown in Eq. (12).

$$L_{ins_{ij}} = \max(0, [m_0 - (\mathbf{w}_{z_{ij}}^T \mathbf{x}_{ij} - \mathbf{w}_{r_{ij}}^T \mathbf{x}_{ij})]), \tag{12}$$

where $r_{ij} = \underset{k \in \{0,1,2,...,K\}, k \neq z_{ij}}{arg\max} \mathbf{w}_k^T \mathbf{x}_{ij}$; m_0 is the margin parameter to separate all $K + 1$ patterns. So each instance $\mathbf{x}_{ij}$ can be classified using

$$z_{ij} = \underset{k \in \{0,1,2,...,K\}}{arg\max} \mathbf{w}_k^T \mathbf{x}_{ij}. \tag{13}$$

The formulation of instance-level loss $L_{ins_{ij}}$ leads to a semi-convex question [8], because the $-\mathbf{w}_{z_{ij}} \mathbf{x}_{ij} = -\underset{k}{max}\, \mathbf{w}_k \mathbf{x}_{ij}$ is concave. But it become convex when the z_{ij} in all bags is given. The approach "coordinate descent" is proposed to address this kind of problem [8]:

- *Relabel instances:* Using Eq. (13) to update instance labels for all bags.
- *Optimize* $\mathbf{W}$*:* Optimize $\mathbf{W}$ according to Eq. (11).

D. Derivations

The Optimize $\mathbf{W}$ step of the coordinate descent method can be solved via stochastic gradient descent (SGD) as all parts in Eq. (11) are differentiable. So we need to compute the partial derivative to $\mathbf{w}_k$ to do the SGD.

For Eq. (4), the chain rule can be performed, and the partial derivative towards $\mathbf{w}_k$ is

$$\frac{\partial L_{bag_i}}{\partial \mathbf{w}_k} = \sum_{m=1}^{K} \frac{\partial L_{bag_i}}{\partial p_{ij_m m}} \frac{\partial p_{ij_m m}}{\partial \mathbf{w}_k}, \tag{14}$$

where

$$\frac{\partial L_{bag_i}}{\partial p_{ij_m m}} = -\frac{Y_i - p_{ij_m m}}{p_{ij_m m}(1 - p_{ij_m m})}. \tag{15}$$

According to Eq. (5), the partial derivative of p_{ij} to $\mathbf{w}_k$ can be got as follows.

$$\frac{\partial p_{ijm}}{\partial \mathbf{w}_k} = \begin{cases} p_{ijm}(1 - p_{ijm})\mathbf{x}_{ij} & if\ k = m \\ -p_{ijm} p_{ijk} \mathbf{x}_{ij} & otherwise. \end{cases} \tag{16}$$

Substituting Eq. (15, 16) into Eq. (14), we can rewrite the Eq. (14) as

$$\frac{\partial L_{bag_i}}{\partial \mathbf{w}_k} = \begin{cases} \sum_{m=1}^{K} \frac{p_{ij_m k}(Y_i - p_{ij_m m})\mathbf{x}_{ij_m}}{1 - p_{ij_m m}} & if\ k = 0 \\ -(Y_i - p_{ij_k k})\mathbf{x}_{ij_k} & \\ \quad + \sum_{m=1, m \neq k}^{K} \frac{p_{ij_m k}(Y_i - p_{ij_m m})\mathbf{x}_{ij_m}}{1 - p_{ij_m m}} & otherwise, \end{cases} \tag{17}$$

where $j_m = \underset{j}{arg\max}\ p_{ijm}$.

The partial derivative of Eq. (12) can be derived as

$$\frac{\partial L_{ins_{ij}}}{\partial \mathbf{w}_k} = \begin{cases} -\mathbf{1}[\mathbf{w}_{z_{ij}}^T \mathbf{x}_{ij} - \mathbf{w}_{r_{ij}}^T \mathbf{x}_{ij} < m_0]\mathbf{x}_{ij} & if\ k = z_{ij} \\ \mathbf{1}[\mathbf{w}_{z_{ij}}^T \mathbf{x}_{ij} - \mathbf{w}_{r_{ij}}^T \mathbf{x}_{ij} < m_0]\mathbf{x}_{ij} & if\ k = r_{ij} \\ 0 & otherwise, \end{cases} \tag{18}$$

where $\mathbf{1}[a < b]$ is the indicator function which equals to 1 when $a < b$ is true and 0 otherwise.

E. SGD Optimization

In the last subsection, we derived the partial derivative of the objective function to perform the SGD, and here we will describe a gradient descent approach SGD for optimizing $\mathbf{W}$. For t-th SGD iteration, we randomly choose a bag (X_{i_t}, Y_{i_t}) from the training set D, and the objective function Eq. (11) can be approximated to Eq. (19) for one sample.

$$L(\mathbf{W}; X_{i_t}) = \frac{\lambda}{2}\sum_{k=0}^{K}\|\mathbf{w}_k\|_2^2 + \beta L_{bag_{i_t}} + \frac{1}{m_{i_t}}\sum_{j=1}^{m_{i_t}} L_{ins_{i_t j}}. \tag{19}$$

So we can compute the gradient of Eq. (19) as follows.

$$\begin{aligned}\nabla_t &= \frac{\partial L(\mathbf{W}; X_{i_t})}{\partial \mathbf{w}_k} \\ &= \lambda \mathbf{w}_k + \beta\frac{\partial L_{bag_{i_t}}}{\partial \mathbf{w}_k} + \frac{1}{m_{i_t}}\sum_{j=1}^{m_{i_t}}\frac{\partial L_{ins_{i_t j}}}{\partial \mathbf{w}_k}\end{aligned} \tag{20}$$

We optimize the weight matrix $\mathbf{W}$ using the varied learning rate $\eta_t = 1/[(t+1)\lambda]$, that is, $\mathbf{w}_{k_{t+1}} \leftarrow \mathbf{w}_{k_t} - \eta_t \cdot \nabla_t$. And after each SGD iteration, we rescale $\mathbf{w}_{k_{t+1}}$ by $min\{1, 1/(\sqrt{\lambda}\|\mathbf{w}_{k_{t+1}}\|_2)\}$, just following the PEGASOS [22], which can significantly accelerate the rate of convergence.

The pseudo-code of our algorithm MiDP is shown in Algorithm 1, which gets the satisfactory accuracy and fast speed although the local optimal of Eq. (11) can only be gotten. Where the T and N are the number of iterations for SGD and coordinate descent respectively.

Algorithm 1 Learning MiDDP using re-sampling strategy

Input: $D, \lambda, \beta, T, m_0, f_p, K, N$
Output: $\mathbf{W}$

1: **Initialize:** Set $\mathbf{W} = 0$. For $Y_i = 0$, set all $z_{ij} = 0$; for $Y_i = 1$, use k-means algorithm to divide the instances in positive bags into K clusters, and set z_{ij} equal to the cluster label.
2: **for** $iter = 1$ **to** N **do**
3: **Optimize W:**
4: Set $\mathbf{W}_1 = \mathbf{W}$
5: **for** $t = 1$ **to** T **do**
6: **if** $t \ mod \ f_p = 0$ **then**
7: Choose X_{i_t} from positive bags, without repetition
8: Set $Y_{i_t} = 1$
9: **else**
10: Choose X_{i_t} from negative bags, without repetition
11: Set $Y_{i_t} = 0$
12: **end if**
13: Set $\eta_t = \frac{1}{\lambda t}$
14: Set $\mathbf{w}_{k_{t+1}} \leftarrow (1-\eta_t\lambda)\mathbf{w}_{k_t} - \eta_t(\beta_t\frac{\partial L_{bag_{i_t}}}{\partial \mathbf{w}_{k_t}} + \frac{1}{m_{i_t}}\sum_{j=1}^{m_{i_t}}\frac{\partial L_{ins_{i_t j}}}{\partial \mathbf{w}_{k_t}})$
15: Set $\mathbf{w}_{k_{t+1}} \leftarrow \min\{1, \frac{1/\sqrt{\lambda}}{\|\mathbf{w}_{k_{t+1}}\|_2}\}\mathbf{w}_{k_{t+1}}$
16: **end for**
17: Set $\mathbf{W} = \mathbf{W}_{T+1}$
18: **Relabel instances:**
19: Use Eq. (13) to update instance labels for all bags.
20: **end for**

IV. Image Representation Using Multi-instance Deep Discriminative Patterns

The purpose of our work is using MiDP to represent images. The images are corresponding to the bags referred above, and the patches are the instances. As we use deep features as the descriptor of patches to learn the MiDP, we call it Multi-instance Deep Discriminative Patterns (MiDDP).

Suppose there are M different classes images in the dataset. During the training procedure, for each image class, the images in this class are used as positive bags, and the rest as negative bags. So M different weight matrix $\mathbf{W}_i, i \in \{1, 2, ..., M\}$ are learned using Algorithm 1, and each $\mathbf{W}_i$ is a $\{K+1\}$-class linear classifier, corresponding to $K+1$ patterns. But there may be a problem when we learn the pattern through this way, that is, there is a big unbalance between the positive and negative images, as there are $M-1$ classes in negative images, but only one class in positive! So it is necessary to expand the MiDDP to adapt the unbalanced dataset. We change the object function Eq. (11) to Eq. (21):

$$\begin{aligned}\min_{\mathbf{W}} \ &\frac{\lambda}{2}\sum_{k=0}^{K}\|\mathbf{w}_k\|_2^2 + \frac{\alpha_1}{n_1}(\beta\sum_{i=1}^{n_1}L_{bag_{1_i}} + \sum_{i=1}^{n_1}\frac{1}{m_{1_i}}\sum_{j=1}^{m_{1_i}}L_{in\,s_{1_i j}}) \\ &+ \frac{\alpha_0}{n_0}(\beta\sum_{i=1}^{n_0}L_{bag_{0_i}} + \sum_{i=1}^{n_0}\frac{1}{m_{0_i}}\sum_{j=1}^{m_{0_i}}L_{ins_{0_i j}}),\end{aligned} \tag{21}$$

where the first term in Eq. (21) is for the regularization; $1, 0$ are corresponding to the bag label; $1_i, 0_i, n_1, n_0$ are the i-th positive bag, i-th negative bag, the number of positive bags, the number of negative bags respectively. So we can control the level of regularization of positive and negative bags, through changing the parameters α_1 and α_0 to aim at the unbalanced dataset.

We can observe that in Eq. (21) there are two different parameters α_1 and α_0 to control the level of regularization. And in SGD, there is an alternative to reach this goal. As the exemplar SVM in [21], we can implicitly choose the penalty weights α_1 and α_0 using the re-sampling strategy. For more detail, we choose a positive bag randomly every f_p to perform the SGD optimization and choose negative bags other iterations. The pseudo-code of the re-sampling strategy can be seen at the 6-th to 12-th rows in Algorithm 1.

When a new image is coming, patch-level DCNN features are densely extracted. For a feature vector $\mathbf{x}$ of a patch, the score given k-th weight vector $\mathbf{w}_{ik}$ in $\mathbf{W}_i$ is $s_{ik} = \mathbf{w}_{ik}^T\mathbf{x}, k \in \{0, 1, 2, ..., K\}$. So we can get $M \times (K+1)$ different scores. The spatial information is also considered using SPM [7]. The SPM divide an image into different grids, each grid contains many patches. To aggregate all representations of patches in a grid into one vector, the pooling strategy is necessary. The generally max-pooling is chosen for aggregation. In each SPM grid, the maximum score of each pattern is computed and then concatenated, resulting in a $M \times (K+1)$ dimensional feature vector. And feature vectors in all grids are concatenated to form the final representation of the coming image.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2016.2642781, IEEE Transactions on Image Processing

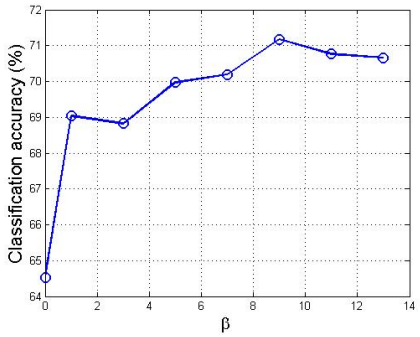


Fig. 2. Classification accuracy of MiDDP on Action 40 over different value of regularization parameter β.

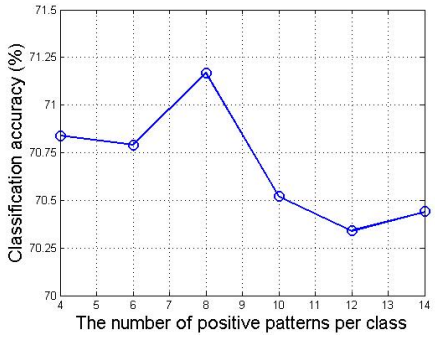


Fig. 3. Classification accuracy of MiDDP on Action 40 over different number of patterns per class K.

Note that the complexity of the image representation is very low. It only refers to the dot product and maximum operation. And it is very effective. For image classification, the representation using MiDDP gets promising results on several benchmark datasets.

V. Experiments

To evaluate performance of the proposed MiDDP method, we perform some experiments on four popular datasets for image classification, including Action 40 [37] for action recognition, Caltech 101 [38] for object recognition, 15 Scenes [7], MIT-indoor [39], and SUN 397 [40] for scene classification.

A. Experimental Setup

In all of our experiments, DCNN features of patches are extracted with the Caffe library [41], using the model trained by [42]. And the features are the activations from the 7-th layer of the model (fc7). All patches in each image are densely sampled from four scales $\{72\times72, 96\times96, 120\times120, 144\times144\}$ by every 32 pixels. We randomly sample at most 100 images as positive images and 1400 images as negative images to train. At most 300 patches for each positive image and 100 patches for each negative image are also sampled randomly for training. The final image representation is as described in Section IV, the max-pooling is used, and three-level SPM $(1\times1, 2\times2, 4\times4)$ is selected to represent images. We set $\lambda = 0.0005$, $m_0 = 0.2$ for all experiments, which is determined by cross validation on Action 40. Other parameters are chosen using the Action 40 as the validation set. In the initializing step of Algorithm 1, the k-means is performed with VLFeat [43]. After image representation is computed, the linear SVM is used to classify images and the LibLinear [44] is adapt to train and test the SVM.

B. Action 40

Some detailed experiments are reported on Action 40 firstly. The Action 40 concludes 9,532 images with 40 classes of different human actions, such as "applauding", "blowing bubbles", "playing guitar", and so on. And for each class, there are 180 to 300 images. Following [37], we use the same 100 images in each class for training and the remains for testing.

1) Impact of Regularization Parameter β: We first study the impact of regularization parameter β, which controls the importance of the bag-level loss L_{bag} relative to the instance-level loss L_{ins}. That is, when the value of β is larger, the L_{bag} is more important. The other parameters are set as $K = 8$, $N = 3$, $T = 10000$, $f_p = 2$ in these experiments. As shown in Fig. 2, we first note that the MIL constraints can lead to better results comparing with no constraints: without MIL constraints ($\beta = 0$), the accuracy is 64.52%, while it is even for weaker constraint ($\beta = 1$), the accuracy can achieve 69.03%, which is much higher than no constraints! We can observe that as the β become larger the accuracy increase, and it reaches the peak value 71.17% when $\beta = 9$. It is reasonable because there is a trade-off between MIL constraints and max-margin constraint, which is controlled by β. So the accuracy can benefit from better β. The $\beta = 9$ will be fixed in all of the following experiments.

2) Impact of the Number of Patterns Per class: The next set of experiments is designed to evaluate the impact of the number of patterns per class. We fix other parameters being $N = 3$, $T = 10000$, $f_p = 2$. The results are shown in Fig. 3. From the results we can conclude as following. When $K = 8$, we can get best performance 71.17%. Too large and too small K work worse because if K is too large, some resulting positive patterns will also exist in negative images, so the learned patterns are not discriminative enough; and if K is too small, we cannot find enough discriminative patterns in positive images, which will obviously harm the performance. We will fix $K = 8$ in all of the rest experiments. Note that the total number of patterns is only $40\times(8+1) = 360$, which is much smaller than many part-based models [10]–[12] using thousands even tens of thousands part detectors.

3) Impact of the Number of Coordinate Descent Iterations: Then we evaluate the impact of different number of coordinate descent iterations N. We set $T = 10000$, $f_p = 2$ in these experiments. As shown in Fig. 4, the results are not very sensitive to the N. There are two reasons resulting in it. First, the local descriptor in our experiments is learned by DCNN, which is very discriminative so even unsupervised clustering method k-means can performs well for labeling patches, so the patch labels change a little after updating labels. Second, the carefully selected regularization parameter β is set to 9,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2016.2642781, IEEE Transactions on Image Processing

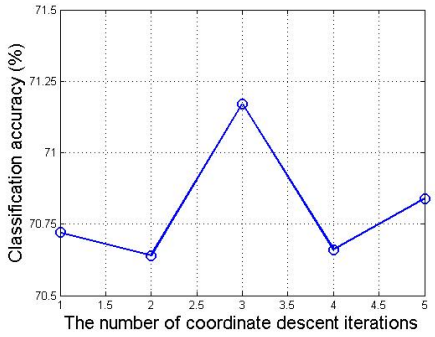


Fig. 4. Classification accuracy of MiDDP on Action 40 over different number of coordinate descent iterations N.

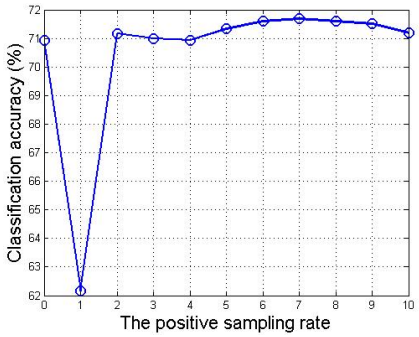


Fig. 6. Classification accuracy of MiDDP on Action 40 over different positive sampling rate f_p.

Fig. 5. Classification accuracy of MiDDP on Action 40 over different number of SGD iterations T.

TABLE I
CLASSIFICATION ACCURACY FOR DIFFERENT METHODS ON ACTION 40.

Methods	Accuracy (%)
Hand-crafted feature	
SB [37]	45.70
SMP [45]	53.00
DCNN feature	
ImageNet fc7 + Linear SVM [33]	54.92
Places fc7 + Linear SVM [33]	42.86
Hybrid fc7 + Linear SVM [33]	55.28
MiDDP	**71.69**

which means the bag-level loss L_{bag} is more important than the instance-level loss L_{ins}, and the L_{bag} only consider the bag labels during the training. Without losing generality, we choose $N = 3$.

4) Impact of the Number of SGD Iterations: In Fig. 5, we evaluate the impact of different number of SGD iterations per coordinate descent iteration. We fix the positive sampling rate f_p equal to 2. As the T increase, the performance is enhanced, and the enhancement tends to saturation after 10000 iterations. It is obvious that more SGD iterations need more runtimes. Considering both training time and classification accuracy, we choose $T = 10000$ to implement following experiments.

5) Impact of the Positive Sampling Rate: The final parameter needed to determine is the positive sampling rate f_p. As shown in Fig. 6, the effect of varying the positive sampling rate during the SGD optimization process is evaluated. The $f_p = 0$ in that figure means there is no re-sampling process, and $f_p = 1$ means only the positive images are used in SGD. It shows that when only positive images are used, the result will be poor (only 62.16%), and the result will be improved about 0.76% when $f_p = 7$ (71.69%) comparing with no re-sampling operation $f_p = 0$ (70.93%). The improvement is not very significant, and it seems the performance does not benefit much from the re-sampling strategy. But we should note that we only sample 1400 negative images during the training, which somewhat alleviates the problem brought by the unbalanced dataset, and can accelerate the convergence rate of SGD. In our experiments, only 1400 negative images can get satisfactory performance. Although the re-sampling strategy seems not very useful, we still put it forward because the performance can benefit from the re-sampling even for this situation, and there may be the case that 1400 negative images are not enough to represent all of the negative images so more negative images are needed leading to the bigger unbalanced problem. The f_p will be fixed to 7 in following experiments.

6) Comparison with Other Methods: To evaluate the effectiveness of our approach, we compare our method with other methods, as shown in Table I. Our method achieves much higher performance comparing with other methods using hand-crafted features or only using DCNN features: +18.69% comparing with SMP [45] and +16.41% comparing with using Hybrid fc7 feature (ImageNet + Places) + Linear SVM method [33].

7) Time Costing for Learning Patterns and Representing Images: All of our programs are written in MATLAB running on PC with Inter(R) Xeon(R) i7-E5 2670 CPU (2.60GHz), NVIDIA GeForce GTX 750 GPU, and 64GB RAM. The time costing for extracting DCNN features is 3 to 5 seconds per image. Actually, this procedure can be optimized through sharing computation on convolutional layers for overlapping patches, just like the SPPnet [46]. After extracting the DCNN features, in the training stage, it takes about 9 minutes to learn 9 patterns per category, so the total training costing is about 6 hours. And in the testing stage, it takes less than 0.015 second for representing image, which is obviously negligible. So it is very efficient.

TABLE II
CLASSIFICATION ACCURACY FOR DIFFERENT METHODS ON CALTECH 101 AND 15 SCENES.

Methods	Accuracy (%)	
	Caltech 101	15 Scenes
Hand-crafted feature		
ScSPM [6]	73.20 ± 0.54	80.40 ± 0.45
Zhu *et al.* [47]	–	81.80 ± 0.40
CA-TM [48]	–	82.50
Xie *et al.* [49]	82.45 ± 0.59	85.13 ± 0.72
EMFS [50]	–	85.70
D-Parts [12]	78.8 ± 0.50	86.0 ± 0.80
MMDL [19]	–	86.70 ± 0.40
IFK+MSRM [51]	77.60 ± 0.67	87.51 ± 0.43
Object-to-Class Kernels [52]	65.54	88.81
DCNN feature		
ImageNet fc7 + Linear SVM [33]	87.22 ± 0.92	84.23 ± 0.37
Places fc7 + Linear SVM [33]	65.18 ± 0.88	90.19 ± 0.34
Hybrid fc7 + Linear SVM [33]	84.79 ± 0.66	91.59 ± 0.48
CNN-S [42]	88.35 ± 0.56	–
SPPnet (OverFeat-7) [46]	93.42 ± 0.50	–
MiDDP	$\mathbf{94.68 \pm 0.20}$	$\mathbf{92.89 \pm 0.23}$

C. Caltech 101

The Caltech 101 contains 9,144 images divided into 102 classes, including a background class. Our experiment only uses the 101 object classes. We randomly split the dataset into training and testing set (30 images per class for training and the rest for testing) five times.

Table II presents the results by MiDDP and other methods. Our method consistently outperforms other methods using hand-crafted features or DCNN features. So it can learn discriminative patterns which is very useful for image classification.

D. 15 Scenes

There are 15 different classes including 4,485 images in the 15 Scenes dataset. And each class contains 200 to 400 images, with average size of 300×250. We randomly split the dataset into training and testing set (100 images per class for training and the rest for testing). This operation is performed five times and the average accuracy is computed.

As shown in Table II, the accuracy of our method can reach 92.89% on 15 Scenes. We can observe that the our MiDDP outperforms the results got by methods using hand-crafted features or using the DCNN feature trained by ImageNet or scene dataset, even by the hybrid DCNN [33].

E. MIT-indoor

The MIT-indoor is a very challenging dataset for scene classification. There are 67 categories of indoor scenes and totally 15,620 images in MIT-indoor. The fixed 80 images for training and 20 images for testing per category are used in our experiment.

The classification accuracy on MIT-indoor is shown in Table III. As we can see, the method using the concatenation of semantic Fisher Vector with fc8 and output from Place DCNN fc7 achieves the best performance on MIT-indoor [18]. It is not surprise because the Place DCNN uses a large scene dataset to train the DCNN model, which mainly focus on scene images, and the ImageNet DCNN and Place CNN are somewhat complementary to each other. And [32] also outperforms our methods because it also benefit from Place DCNN and it uses 13400 parts which is much larger than ours ($9 \times 67 = 603$). Our method only uses the ImageNet DCNN, and outperforms other methods using DCNN features from ImageNet, like sparse code with the output from fully connected 6-th layer (fc6) [53], VLAD with fc7 [17], semantic Fisher Vector with fc8 [18]. We also test the performance of MMDL using the same feature and the same number of patterns as MiDDP. The results shows that our MiDDP achieves higher accuracy than MMDL, meanwhile the MiDDP is easier to optimize and more tractable. And it is obvious that using the DCNN feature can achieve more promising results than using hand-crafted features.

TABLE IV
CLASSIFICATION ACCURACY FOR DIFFERENT METHODS ON SUN 397.

Methods	Accuracy (%)
Hand-crafted feature	
Xiao et al. [40]	38.00
EMFS [50]	40.70
LASC [57]	45.30 ± 0.40
DCNN feature	
ImageNet fc7 + Linear SVM [33]	42.61 ± 0.16
Places fc7 + Linear SVM [33]	54.32 ± 0.14
Hybrid fc7 + Linear SVM [33]	53.89 ± 0.21
ImageNet fc7+VLAD [17]	51.98
ImageNet fc7+FV [17]	53.00 ± 0.40
ImageNet fc8+FV [18]	54.40 ± 0.30
ImageNet fc8+FV+Places fc7 [18]	$\mathbf{61.72 \pm 0.13}$
MiDDP	54.58 ± 0.24

F. SUN 397

We also test our method on SUN 397, which is a very large dataset for scene classification. It has 397 different scene classes, including outdoor and indoor scenes. There are totally more than 100K images in SUN 397, and each class has at least 100 images. The fixed ten different partition of training and testing set by [40] are chosen in our experiments, *i.e.*, 50 images for training and 50 for testing per class per partition.

The classification accuracy is shown in Table IV. The similar conclusion that our method outperforms other methods when only uses DCNN features from ImageNet can arrive.

VI. DISCUSSION

In this section, we will show some learned patterns of our method, and discuss the strength and weakness of our method.

As shown in Fig. 7, some learned patterns in some MIT-indoor categories are presented. All patterns are learned only using image labels, but these patterns have much semantic information. For example, pattern 2 and pattern 4 in movietheater correspond to seats and movie screens respectively. We can observe that our method learns multi-positive patterns per class successfully, which demonstrates the claim in Section III-B.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2016.2642781, IEEE Transactions on Image Processing

TABLE III
CLASSIFICATION ACCURACY FOR DIFFERENT METHODS ON MIT-INDOOR.

Methods	Accuracy (%)	Methods	Accuracy (%)
Hand-crafted feature		DCNN feature	
Zhang *et al.* [54]	32.24	ImageNet fc7 + Linear SVM [33]	56.79
Object-to-Class Kernels [52]	39.85	Places fc7 + Linear SVM [33]	68.24
DPM+Gist-color+SP [9]	43.10	Hybrid fc7 + Linear SVM [33]	70.80
Zhu *et al.* [47]	43.90	DeCaf [55]	59.50
BoP [10]	46.10	ImageNet fc6 + SC [53]	68.20
Xie *et al.* [49]	46.38	ImageNet fc7 + VLAD [17]	68.88
miSVM [56]	46.40	OverFeat + SVM [16]	69.00
EMFS [50]	48.20	ImageNet fc7 + FV [17]	69.70
Patches+GIST+SP+DPM [29]	49.40	MDPM [30]	70.46
MMDL [19]	50.15	ImageNet fc8 + FV [18]	72.86
D-Parts [12]	51.40	Parizi *et al.* (372 parts) [32]	73.30
IFV [10]	60.77	Parizi *et al.* (13400 parts) [32]	77.10
BoP+IFV [10]	63.10	ImageNet fc8 + FV + Places fc7 [18]	**79.00**
LASC [57]	63.40	ImageNet fc7 + MMDL	73.06
Doersch *et al.* [11]	64.03	MiDDP	74.10

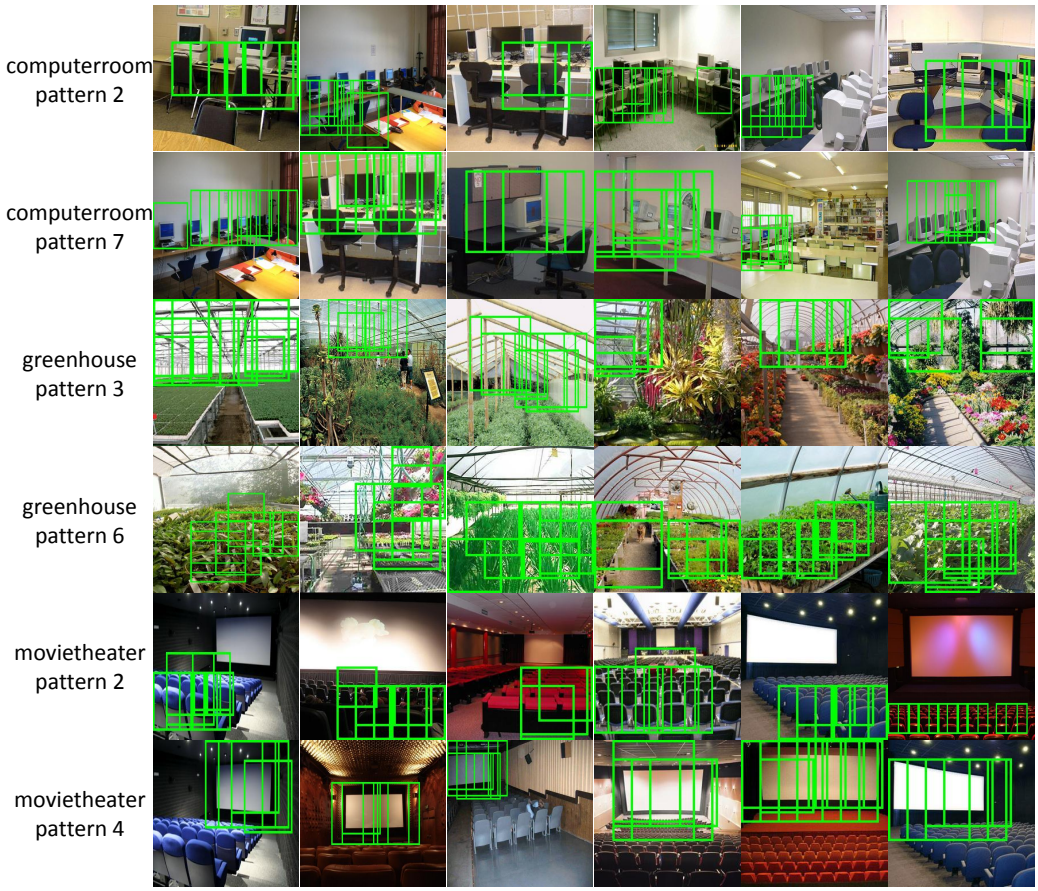


Fig. 7. Some patterns learned by MiDDP for different classes. Each row corresponds to a pattern, where green rectangles show positions where the probability of that patch belonging to the pattern larger than others.

The results in the last section have shown that our method can achieve better performance compared with methods using both hand-crafted features and ImageNet DCNN features. This demonstrates the effectiveness of our method. But it is still insufficient when distinguish similar classes. For example, from the confusion matrix on MIT-indoor (Fig. 8), we can observe that the performance is unsatisfactory when encounters similar categories (*e.g.*, bookstore and library in Fig. 9). These categories share same patterns, so it is hard to find discriminative patterns for it. How to distinguish these images is a challenging problem and waits us to work out.

VII. CONCLUSION

In this work, we propose a novel MIL algorithm to learn multi-positive patterns for each class instead of single-positive pattern, and use it to learn multi-instance deep discriminative

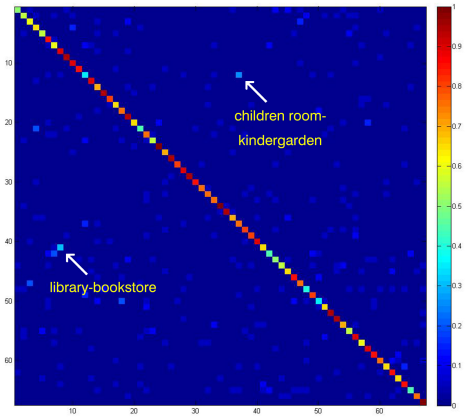


Fig. 8. The confusion matrix on MIT-indoor. The performance for distinguishing similar categories (*e.g.*, library-bookstore) is unsatisfactory.

Fig. 9. Some pictures in bookstore and library.

patterns for image classification. In MiDDP, we find several discriminative patterns for each image class, and concatenate score of each pattern as patch representation, then use max-pooling to aggregate patch representations into the final image representation. We test our method on some datasets for image classification. It only needs nine patterns per class, and achieves remarkable performance on these datasets. Results demonstrate that our method is both effective and efficient. In the future we would like to integrate the MiDDP loss into a DCNN framework for end-to-end learning. Using our MiDDP method for more visual tasks, like image retrieval, and find more powerful methods to find patterns for image representation are also waiting us to explore.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable suggestions.

REFERENCES

[1] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. Workshop Statist. Learn. Comput. Vis., Eur. Con. Comput. Vis.*, 2004, pp. 1–22.
[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.
[4] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2486–2493.
[5] J. VanGemert, C. Veenman, A. Smeulders, and J. Geusebroek, "Visual word ambiguity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1271–1283, 2010.
[6] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. Comput. Vis. Pattern Recognit.*, 2009, pp. 1794–1801.
[7] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. Comput. Vis. Pattern Recognit.*, 2006, pp. 2169–2178.
[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.
[9] M. Pandey and S. Lazebnik, "Scene recognition and weakly supervised object localization with deformable part-based models," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 1307–1314.
[10] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *Proc. Comput. Vis. Pattern Recognit.*, 2013, pp. 923–930.
[11] C. Doersch, A. Gupta, and A. A. Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Proc. Neural Inf. Process. Syst.*, 2013, pp. 494–502.
[12] J. Sun and J. Ponce, "Learning discriminative part detectors for image classification and cosegmentation," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 3400–3407.
[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[14] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[15] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. Comput. Vis. Pattern Recognit.*, 2014, pp. 1717–1724.
[16] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proc. Comput. Vis. Pattern Recognit. Workshop*, 2014, pp. 512–519.
[17] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 392–407.
[18] M. Dixit, S. Chen, D. Gao, N. Rasiwasia, and N. Vasconcelos, "Scene classification with semantic fisher vectors," in *Proc. Comput. Vis. Pattern Recognit.*, 2015, pp. 2974–2983.
[19] X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu, "Max-margin multiple-instance dictionary learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 846–854.
[20] X. Wang, Z. Zhu, C. Yao, and X. Bai, "Relaxed multiple-instance svm with application to object discovery," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 1224–1232.
[21] J. Zepeda and P. Pérez, "Exemplar svms as visual feature encoders," in *Proc. Comput. Vis. Pattern Recognit.*, 2015, pp. 3052–3060.
[22] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: primal estimated sub-gradient solver for svm," *Math. Program.*, vol. 127, no. 1, pp. 3–30, 2011.
[23] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. Comput. Vis. Pattern Recognit.*, 2009, pp. 1778–1785.
[24] D. Pechyony and V. Vapnik, "On the theory of learnining with privileged information," in *Proc. Neural Inf. Process. Syst.*, 2010, pp. 1894–1902.
[25] D. Parikh and K. Grauman, "Relative attributes," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 503–510.
[26] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 1365–1372.
[27] L. J. Li, H. Su, L. Fei-Fei, and E. P. Xing, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *Proc. Neural Inf. Process. Syst.*, 2010, pp. 1378–1386.
[28] Z. Niu, G. Hua, X. Gao, and Q. Tian, "Spatial-disclda for visual recognition," in *Proc. Comput. Vis. Pattern Recognit.*, 2011, pp. 1769–1776.
[29] S. Singh, A. Gupta, and A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 73–86.
[30] Y. Li, L. Liu, C. Shen, and A. van den Hengel, "Mid-level deep pattern mining," in *Proc. Comput. Vis. Pattern Recognit.*, 2015, pp. 971–980.
[31] B. Shi, X. Bai, and C. Yao, "Script identification in the wild via discriminative convolutional neural network," *Pattern Recognition*, vol. 52, pp. 448–458, 2016.
[32] S. N. Parizi, A. Vedaldi, A. Zisserman, and P. Felzenszwalb, "Automatic discovery and optimization of parts for image classification," in *Proc. Int. Conf. Learn. Repr.*, 2015.

[33] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 487–495.

[34] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, no. 1-2, pp. 31–71, 1997.

[35] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. Neural Inf. Process. Syst.*, 2002, pp. 561–568.

[36] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep multiple instance learning for image classification and auto-annotation," in *Proc. Comput. Vis. Pattern Recognit.*, 2015, pp. 3460–3469.

[37] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 1331–1338.

[38] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[39] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. Comput. Vis. Pattern Recognit.*, 2009, pp. 413–420.

[40] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Proc. Comput. Vis. Pattern Recognit.*, 2010, pp. 3485–3492.

[41] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. of the ACM International Conf. on Multimedia*, 2014, pp. 675–678.

[42] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. Brit. Mach. Vis. Conf.*, 2014.

[43] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *Proc. Multimedia*, 2010, pp. 1469–1472.

[44] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008.

[45] F. S. Khan, J. van de Weijer, R. M. Anwer, M. Felsberg, and C. Gatta, "Semantic pyramids for gender and action recognition," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3633–3645, 2014.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015.

[47] J. Zhu, T. Wu, S. C. Zhu, X. Yang, and W. Zhang, "A reconfigurable tangram model for scene representation and categorization," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 150–166, 2016.

[48] Z. Niu, G. Hua, X. Gao, and Q. Tian, "Context aware topic model for scene recognition," in *Proc. Comput. Vis. Pattern Recognit.*, 2012, pp. 2743–2750.

[49] L. Xie, Q. Tian, M. Wang, and B. Zhang, "Spatial pooling of heterogeneous features for image classification," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 1994–2008, 2014.

[50] X. Song, S. Jiang, and L. Herranz, "Joint multi-feature spatial context for scene recognition in the semantic manifold," in *Proc. Comput. Vis. Pattern Recognit.*, 2015, pp. 1312–1320.

[51] S. Bai, X. Bai, and W. Liu, "Multiple stage residual model for image classification and vector compression," *IEEE Trans. Multimedia*, vol. 18, no. 7, pp. 1351–1362, 2016.

[52] L. Zhang, X. Zhen, and L. Shao, "Learning object-to-class kernels for scene classification," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3241–3253, 2014.

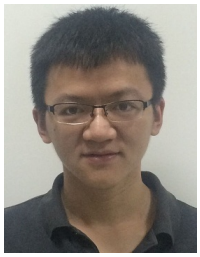
[53] L. Liu, C. Shen, L. Wang, A. van den Hengel, and C. Wang, "Encoding high dimensional local features by sparse coding based fisher vectors," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 1143–1151.

[54] L. Zhang, Y. Gao, R. Zimmermann, Q. Tian, and X. Li, "Fusion of multichannel local and global structural cues for photo aesthetics evaluation," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1419–1429, 2014.

[55] 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.

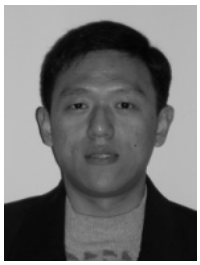
[56] Q. Li, J. Wu, and Z. Tu, "Harvesting mid-level visual concepts from large-scale internet images," in *Proc. Comput. Vis. Pattern Recognit.*, 2013, pp. 851–858.

[57] P. Li, X. Lu, and Q. Wang, "From dictionary of visual words to subspaces: Locality-constrained affine subspace coding," in *Proc. Comput. Vis. Pattern Recognit.*, 2015, pp. 2348–2357.



Peng Tang is a Ph.D. student in the School of Electronics Information and Communications, Huazhong University of Science and Technology (HUST). He received his B.S. degree from HUST in 2010. He is a reviewer of Neurocomputing. His research interests include Computer Vision and Machine Learning. In particular, he focuses on mid-level representation for image understanding.

Xinggang Wang is an assistant professor of School of Electronics Information and Communications of Huazhong University of Science and Technology (HUST). He received his B.S. degree in communication and information system and Ph.D. degree in computer vision both from HUST. From May 2010 to July 2011, he was with the Department of Computer and Information Science, Temple University, Philadelphia, PA., as a visiting scholar. From February 2013 to September 2013, he was with the University of California, Los Angeles, as a visiting graduate researcher. He is a reviewer of IEEE Transaction on Cybernetics, Pattern Recognition, Computer Vision and Image Understanding, Neurocomputing, CVPR, ICCV and ECCV etc. His research interests include computer vision and machine learning.



Bin Feng received the B.S. and Ph.D. degrees in electronics and information engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2001 and 2006, respectively. He is currently an Associate Professor with the School of Electronic Information and Communications, HUST. His research interests include computer vision and intelligent video analysis.

Wenyu Liu received the B.S. degree in Computer Science from Tsinghua University, Beijing, China, in 1986, and the M.S. and Ph.D. degrees, both in Electronics and Information Engineering, from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1991 and 2001, respectively. He is now a professor and associate dean of the School of Electronic Information and Communications, HUST. His current research areas include computer vision, multimedia, and machine learning. He is a senior member of IEEE.