Image-To-Class Distance Metric Learning for Image Classification

Zhengxiang Wang, Yiqun Hu and Liang-Tien Chia

Center for Multimedia and Network Technology, School of Computer Engineering
Nanyang Technological University, 639798, Singapore
wang0460@ntu.edu.sg,yiqun.hu@gmail.com,asltchia@ntu.edu.sg

Abstract. Image-To-Class (I2C) distance is first used in Naive-Bayes Nearest-Neighbor (NBNN) classifier for image classification and has successfully handled datasets with large intra-class variances. However, the performance of this distance relies heavily on the large number of local features in the training set and test image, which need heavy computation cost for nearest-neighbor (NN) search in the testing phase. If using small number of local features for accelerating the NN search, the performance will be poor.

In this paper, we propose a large margin framework to improve the discrimination of I2C distance especially for small number of local features by learning Per-Class Mahalanobis metrics. Our I2C distance is adaptive to different class by combining with the learned metric for each class. These multiple Per-Class metrics are learned simultaneously by forming a convex optimization problem with the constraints that the I2C distance from each training image to its belonging class should be less than the distance to other classes by a large margin. A gradient descent method is applied to efficiently solve this optimization problem. For efficiency and performance improved, we also adopt the idea of spatial pyramid restriction and learning I2C distance function to improve this I2C distance. We show in experiments that the proposed method can significantly outperform the original NBNN in several prevalent image datasets, and our best results can achieve state-of-the-art performance on most datasets.

1 Introduction

Image classification is a highly useful yet still challenging task in computer vision community due to the large intra-class variances and ambiguities of images. Many efforts have been done for dealing with this problem and they can roughly be divided into learning-based and non-parametric methods according to [1]. Compared to learning-based methods, non-parametric methods directly classify on the test set and do not require any training phase. So in most cases, learning-based methods can achieve better recognition performance than non-parametric methods as they have learned the model from the training set, which is useful for classifying test images. But recently a new non-parametric method named as NBNN *et al.* [1] was proposed, which reported comparable performance to those top learning-based methods. They contribute such achievement to the avoidance of descriptor quantization and the use of Image-To-Class (I2C) distance instead of Image-To-Image (I2I) distance, since they proved descriptor quantization and I2I distance lead to significant degradation for classification.

However, the performance of this I2C distance relies heavily on the large number of local features in the training set and test image. For example, the state-of-the-art performance they reported in Caltech 101 dataset is achieved by densely sampling large redundant local features for both training and test images, which results in about 15000 to 20000 features per image. Such large number of features makes the nearest-neighbor (NN) search in I2C distance calculation computationally expensive when classifying a test image, which limits its scalability in real-world application. If only small number of local features is used, the performance of this I2C distance will be poor as shown in the later experiment section, although it needs less testing time.

In this paper, we aim to enhance the performance of I2C distance especially for small number of local features, so as to speed up the testing phase while maintaining excellent result. To achieve this, we propose a training phase to exploit the training information and suggest a distance metric learning method for the I2C distance. Our method avoids the shortcoming of both non-parametric methods and most learning-based methods involving I2I distance and descriptor quantization. This leads to a better recognition performance than NBNN and those learning-based methods. For each class, we learn the class specific Mahalanobis metric to combine with the corresponding I2C distance. When classifying a test image, we select the shortest I2C distance among its Mahalanobis I2C distances to all classes as its predicted class label. Since only the metric of the belonging class can well characterize the local features of the test image, such Per-Class metrics can better preserve the discriminate information between different classes compared to a single global metric.

We adopt the idea of large margin from SVM in our optimization problem for learning these Per-Class metrics, which is also used for distance metric learning by Weinberger *et al.* [18] recently. For each training image, we separate the I2C distance to its belonging class from those to any other class by a large margin, and form a large margin convex optimization problem as an instance of semidefinite programming (SDP). Then we apply an efficient gradient descent method to solve this optimization problem. We also show the incremental learning ability of our Per-Class metric learning, which enables our method to be used for online learning and it can be easily scaled-up for handling large number of classes. Figure 1 gives the illustration of our classification structure. Notations used in the figure will be explained in Section 2.1. Compared to NBNN classifier, the main difference is that we use Mahalanobis distance with our learned Per-Class metrics instead of Euclidean distance in NBNN, while the way to classify a test image is similar.

Moreover, we adopt the idea of spatial pyramid match [9] and learning I2C distance function [17] to generate a more discriminative distance for improving classification accuracy. Since the main computation burden is the NN search in

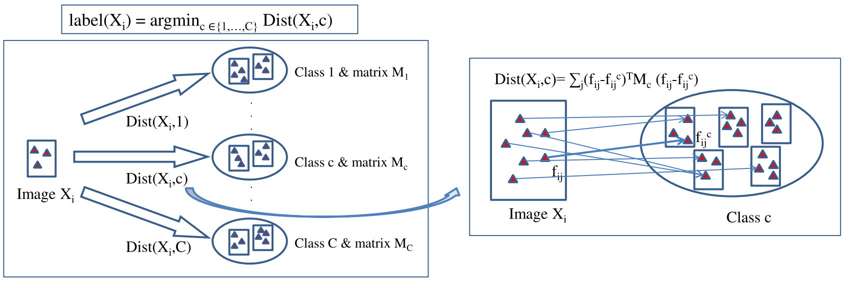

Fig. 1. The classification structure of our method. The rectangular and triangles denote an image and its local feature points respectively. The ellipse denotes a class with images (rectangular) inside it. The I2C distance from image X_i to a class c is formed by the sum of Mahalanobis distance between each local feature f_{ij} and its NN f_{ij}^c in class c with the matrix M_c learned for that class. The predicted label of image X_i is chosen by selecting the shortest I2C distance. Section 2.1 gives a full explanation of these notations.

I2C distance calculation rather than metric learning, we also propose an acceleration method using spatial restriction for speeding up the NN search, which can preserve or even improve the classification accuracy in most datasets. Our objectives for improving the I2C distance are twofold: minimizing the testing time and improving the classification performance.

We describe our large margin optimization problem as well as an efficient solver in Section 2, where we also discuss our improvements in addition to the learned metrics. We evaluate our method and compare it with other methods in Section 3. Finally, we conclude this paper in Section 4.

2 Distance Metric Learning for I2C Distance

In this section, we formulate a large margin convex optimization problem for learning the Per-Class metrics and introduce an efficient gradient descent method to solve this problem. We also adopt two strategies to further enhance the discrimination of our learned I2C distance.

2.1 Notation

Our work deals with the image represented by a collection of its local feature descriptors extracted from patches around each keypoint. So let $F_i = \{f_{i1}, f_{i2}, \dots, f_{im_i}\}$ denote features belonging to image X_i, where m_i represents the number of features in X_i and each feature is denoted as $f_{ij} \in R^d, \forall j \in \{1, \dots, m_i\}$. To calculate the I2C distance from an image X_i to a candidate class c, we need to find the NN of each feature f_{ij} from class c, which is denoted as f_{ij}^c. The original I2C distance from image X_i to class c is defined as the sum of Euclidean distances between each feature in image X_i and its NN in class c and can be formulated as:

$$Dist(X_i, c) = \sum_{j=1}^{m_i} \parallel f_{ij} - f_{ij}^c \parallel^2 \tag{1}$$

After learning the Per-Class metric $M_c \in R^{d \times d}$ for each class c, we replace the Euclidean distance between each feature in image X_i and its NN in class c by the Mahalanobis distance and the learned I2C distance becomes:

$$Dist(X_i, c) = \sum_{j=1}^{m_i} (f_{ij} - f_{ij}^c)^T M_c (f_{ij} - f_{ij}^c) \tag{2}$$

This learned I2C distance can also be represented in a matrix form by introducing a new term ΔX_{ic}, which is a $m_i \times d$ matrix representing the difference between all features in the image X_i and their nearest neighbors in the class c formed as:

$$\Delta X_{ic} = \begin{pmatrix} (f_{i1} - f_{i1}^c)^T \\ (f_{i2} - f_{i2}^c)^T \\ \cdots \\ (f_{im_i} - f_{im_i}^c)^T \end{pmatrix} \tag{3}$$

So the learned I2C distance from image X_i to class c can be reformulated as:

$$Dist(X_i, c) = Tr(\Delta X_{ic} M_c \Delta X_{ic}^T) \tag{4}$$

This is equivalent to the equation (2). If M_c is an identity matrix, then it's also equivalent to the original Euclidean distance form of equation (1). In the following subsection, we will use this formulation in the optimization function.

2.2 Problem Formulation

The objective function in our optimization problem is composed of two terms: the regularization term and error term. This is analogous to the optimization problem in SVM. In the error term, we incorporate the idea of large margin and formulate the constraint that the I2C distance from image X_i to its belonging class p (named as positive distance) should be smaller than the distance to any other class n (named as negative distance) with a margin. The formula is given as follows:

$$Tr(\Delta X_{in} M_n \Delta X_{in}^T) - Tr(\Delta X_{ip} M_p \Delta X_{ip}^T) \geq 1 \tag{5}$$

In the regularization term, we simply minimize all the positive distances similar to [20]. So for the whole objective function, on one side we try to minimize all the positive distances, on the other side for every image we keep those negative distances away from the positive distance by a large margin. In order to allow soft-margin, we introduce a slack variable ξ in the error term, and the whole convex optimization problem is therefore formed as:

$$\begin{aligned} &\min_{M_1,M_2,\ldots,M_C} O(M_1,M_2,\ldots,M_C) = \\ &(1-\lambda)\sum_{i,p\to i} Tr(\Delta X_{ip}M_p\Delta X_{ip}^T) + \lambda \sum_{i,p\to i,n\to i} \xi_{ipn} \\ &s.t.\forall\ i,p,n : Tr(\Delta X_{in}M_n\Delta X_{in}^T) - Tr(\Delta X_{ip}M_p\Delta X_{ip}^T) \geq 1-\xi_{ipn} \\ &\qquad\qquad \forall\ i,p,n : \xi_{ipn} \geq 0 \\ &\qquad\qquad\quad \forall\ c : M_c \succeq 0 \end{aligned} \tag{6}$$

This optimization problem is an instance of SDP, which can be solved using standard SDP solver. However, as the standard SDP solvers is computation expensive, we use an efficient gradient descent based method derived from [20, 19] to solve our problem. Details are explained in the next subsection.

2.3 An Efficient Gradient Descent Solver

Due to the expensive computation cost of standard SDP solvers, we propose an efficient gradient descent solver derived from Weinberger *et al.* [20, 19] to solve this optimization problem. Since the method proposed by Weinberger *et al.* targets on solving only one global metric, we modify it to learn our Per-Class metrics. This solver updates all matrices iteratively by taking a small step along the gradient direction to reduce the objective function (6) and projecting onto feasible set to ensure that each matrix is positive semi-definite in each iteration. To evaluate the gradient of objective function for each matrix, we denote the matrix M_c for each class c at t^{th} iteration as M_c^t, and the corresponding gradient as $G(M_c^t)$. We define a set of triplet error indices N^t such that $(i,p,n) \in N^t$ if $\xi_{ipn} > 0$ at the t^{th} iteration. Then the gradient $G(M_c^t)$ can be calculated by taking the derivative of objective function (6) to M_c^t:

$$G(M_c^t) = (1-\lambda)\sum_{i,c=p} \Delta X_{ic}^T\Delta X_{ic} + \lambda \sum_{(i,p,n)\in N^t,c=p} \Delta X_{ic}^T\Delta X_{ic} - \lambda \sum_{(i,p,n)\in N^t,c=n} \Delta X_{ic}^T\Delta X_{ic} \tag{7}$$

Directly calculating the gradient in each iteration using this formula would be computational expensive. As the changes in the gradient from one iteration to the next are only determined by the differences between the sets N^t and N^{t+1}, we use $G(M_c^t)$ to calculate the gradient $G(M_c^{t+1})$ in the next iteration, which would be more efficient:

$$\begin{aligned} G(M_c^{t+1}) = G(M_c^t) &+ \lambda(\sum_{(i,p,n)\in(N^{t+1}-N^t),c=p} \Delta X_{ic}^T\Delta X_{ic} - \sum_{(i,p,n)\in(N^{t+1}-N^t),c=n} \Delta X_{ic}^T\Delta X_{ic}) \\ &- \lambda(\sum_{(i,p,n)\in(N^t-N^{t+1}),c=p} \Delta X_{ic}^T\Delta X_{ic} - \sum_{(i,p,n)\in(N^t-N^{t+1}),c=n} \Delta X_{ic}^T\Delta X_{ic}) \end{aligned} \tag{8}$$

Since $(\Delta X_{ic}^T\Delta X_{ic})$ is unchanged during the iterations, we can accelerate the updating procedure by pre-calculating this value before the first iteration. The matrix is updated by taking a small step along the gradient direction for each iteration. To enforce the positive semi-definiteness, the updated matrix needs to be projected onto a feasible set. This projection is done by eigen-decomposition of the matrix and truncating all the negative eigenvalues to zeros. As the optimization problem (6) is convex, this solver is able to converge to the global optimum. We summarize the whole work flow in Algorithm 1.

Algorithm 1 A Gradient Descent Method for Solving Our Optimization Problem

```
Input: step size $\alpha$, parameter $\lambda$ and pre-calculated data $(\Delta X_{ic}^T \Delta X_{ic}), i \in \{1, \ldots, N\}, c \in \{1, \ldots, C\}$
for $c := 1$ to C do
  $G(M_c^0) := (1 - \lambda) \sum_{i,p \to i} \Delta X_{ip}^T \Delta X_{ip}$
  $M_c^0 := I$
end for{Initialize M and gradient for each class}
Set $t := 0$
repeat
  Compute $N^t$ by checking each error term $\xi_{ipn}$
  for $c = 1$ to C do
    Update $G(M_c^{t+1})$ using equation (8)
    $M_c^{t+1} := M_c^t + \alpha G(M_c^{t+1})$
    Project $M_c^{t+1}$ for keeping positive semi-definite
  end for
  Calculate new objective function
  $t := t + 1$
until Objective function converged
Output: each matrix $M_1, \ldots, M_C$
```

2.4 Scalability and Incremental Learning

Next we analyze the efficiency of this solver and its scalability. Although the number of triplets is large for dealing with large-scale dataset, for example 151500 triplets in error term for Caltech 101 dataset using 15 images per class for training, we find only a small portion of them are non-zero, which are put into the error index set N^t and used for updating matrices. To speed up calculating N^t in each iteration, we also keep an active set of triplets as proposed in [20] for calculating N^t rather than scanning over all the triplets in each iteration. So this solver runs quickly for updating hundreds of metrics. In our experiment, it needs about 30 iterations to converge for Scene, Sports and Corel datasets, and about 60 iterations for Caltech 101 dataset to converge with an appropriate step size. We can further accelerate the training phase by learning a diagonal matrix for each class, which would alleviate the computation cost especially when there are even more classes, e.g. thousands of classes.

Our method also supports the incremental learning. When new training images of existing class or new class are added, Per-Class metrics do not need to be re-trained from the beginning. The current learned matrices can be used as initial estimates by changing the identity matrix I to current matrix for each class in Algorithm 1, and new triplets are added to update all matrices. The updating procedure will converge quickly since pre-learned matrices are relatively close to the optimal. This incremental learning ability shows that our method can be scaled-up to handle large number of classes and support for on-line learning.

2.5 More Improvements Based on Mahalanobis I2C Distance

To generate a more discriminative I2C distance for better recognition performance, we improve our learned distance by adopting the idea of spatial pyramid match [9] and learning I2C distance function [17].

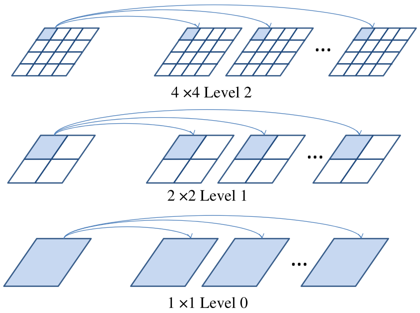

Fig. 2. The left parallelogram denotes an image, and the right parallelograms denote images in a class. We adopt the idea of spatial pyramid by restricting each feature descriptor in the image to only find its NN in the same subregion from a class at each level.

Spatial pyramid match (SPM) is proposed by Lazebnik *et al.* [9] which makes use of spatial correspondence, and the idea of pyramid match is adapted from Grauman *et al.* [8]. This method recursively divides the image into subregions at increasingly fine resolutions. We adopt this idea in our NN search by limiting each feature point in the image to find its NN only in the same subregion from a candidate class at each level. So the feature searching set in the candidate class is reduced from the whole image (top level, or level 0) to only the corresponding subregion (finer level), see Figure 2 for details. This spatial restriction enhances the robustness of NN search by reducing the effect of noise due to wrong matches from other subregions. Then the learned distances from all levels are merged together as pyramid combination.

In addition, we find in our experiments that a single level spatial restriction at a finer resolution makes better recognition accuracy compared to the top level especially for those images with geometric scene structure, although the accuracy is slightly lower than the pyramid combination of all levels. Since the candidate searching set is smaller in a finer level, which requires less computation cost for the NN search, we can use just a single level spatial restriction of the learned I2C distance to speed up the classification for test images. Compared to the top level, a finer level spatial restriction not only reduces the computation cost, but also improves the recognition accuracy in most datasets. For some images without geometric scene structure, this single level can still preserve the recognition performance due to sufficient features in the candidate class.

We also use the method of learning I2C distance function proposed in [17] to combine with the learned Mahalanobis I2C distance. The idea of learning local distance function is originally proposed by Frome *et al.* and used for image classification and retrieval in [6, 5]. Their method learns a weighted distance function for measuring I2I distance, which is achieved by also using a large margin framework to learn the weight associated with each local feature. Wang *et al.* [17] have used this idea to learn a weighted I2C distance function from each image to a candidate class, and we find our distance metric learning method can be combined with this distance function learning approach. For each class, its weighted I2C distance is multiplied with our learned Per-Class matrix to generate a more discriminative weighted Mahalanobis I2C distance. Details of this local distance function for learning weight can be found in [6, 17].

3 Experiment

3.1 Datasets and Setup

We evaluate our proposed method on four popular datasets: Scene-15, Sports, Corel and Caltech 101 dataset. We describe them briefly as follows:

- **Scene-15** Scene dataset consists of 15 scene categories, among which 8 were originally collected by Oliva *et al.* [15], 5 added by Li *et al.* [4] and 2 from Lazebnik *et al.* [9]. Each class has about 200 to 400 images, and the average image size is around 300 × 250 pixels. Following [9], we randomly select 100 images per class for training and test on the rest. The mean per-class recognition rate is reported as accuracy.
- **Sports** Sports event dataset is firstly introduced in [10], consisting of 8 sports event categories. The number of images in each class ranges from 137 to 250, so we follow [10] to select 70 and 60 images per class for training and test respectively. Since images in this dataset are usually very large, they are first resized such that the largest x/y dimension is 500.
- **Corel** Corel dataset contains 10 scene categories published from Corel Corporation. Each class contains 100 images, and we follow [14] to separate them randomly into two subsets of equal size to form the training and test set. All the images are of the size 384 × 256 or 256 × 384.
- **Caltech 101** Caltech 101 dataset is a large scale dataset containing 101 categories [3]. The number of images in each class varies from about 30 to

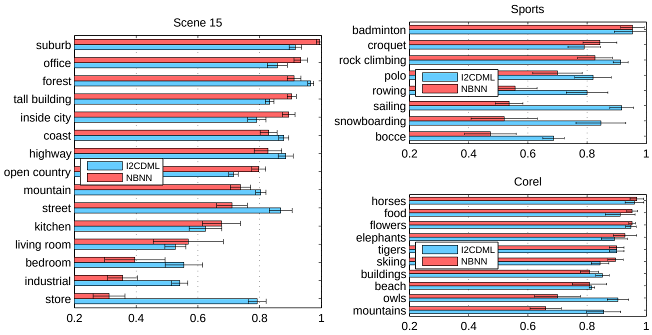

Fig. 3. Per-category recognition accuracy for comparison of I2CDML with NBNN.

800. This dataset is more challenging due to the large number of classes and intra-class variances. Following the widely used measurement by the community we randomly select 15 images per class for training. For test set, we also select 15 images from each class and report the mean accuracy.

Since the training and test set are selected randomly, we repeat the experiment for 5 times in each dataset and report the average result. For feature extraction, we use dense sampling strategy and SIFT features [12] as our descriptor, which are computed on a 16 × 16 patches over a grid with spacing of 8 pixels for all datasets. This is a simplified method compared to some papers that use densely sampled and multi-scale patches to extract large number of features, which helps in the performance results but increases the computational complexity. We name our method as I2CDML, short for Image-To-Class distance metric learning.

3.2 Results on Scene-15, Sports and Corel Datasets

Table 1. Comparing I2CDML to NBNN for recognition accuracy (%)

Method	Scene-15	Sports	Corel
I2CDML	**77.0 ± 0.60**	**78.5 ± 1.63**	**88.8 ± 0.93**
NBNN [1]	72.3 ± 0.93	67.6 ± 1.10	85.7 ± 1.20

We first compare our proposed I2CDML method with NBNN [1] on Scene-15, Sports and Corel datasets to evaluate our learned metrics. Table 1 shows the recognition accuracy averaged of all classes for the three datasets. We can see that our method significantly outperforms NBNN in every dataset, especially in

Sports dataset where the improvement is above 10%. Then we investigate the details by comparing the classification accuracy for each class in Figure 3. For those easily classified categories, our method is comparable to NBNN. Moreover, for those challenging categories that NBNN performs poorly (for example the worst three categories in Scene-15, the worst four in Sports, and the worst two in Corel, as indicated in Figure 3), our method can improve the accuracy substantially. Therefore our method improves the average accuracy by emphasizing the classification on challenging categories and yet maintains the performance for the easily classified categories.

Table 2. Comparing I2CDML to its integration for recognition accuracy (%)

Method	**Scene-15**	**Sports**	**Corel**
I2CDML	77.0 ± 0.60	78.5 ± 1.63	88.8 ± 0.93
I2CDML+SPM	81.2 ± 0.52	79.7 ± 1.83	89.8 ± 1.16
I2CDML+Weight	78.5 ± 0.74	81.3 ± 1.46	90.1 ± 0.94
I2CDML+ SPM+Weight	**83.7 ± 0.49**	**84.3 ± 1.52**	**91.4 ± 0.88**

Then we show in Table 2 the improved I2C distance through spatial pyramid restriction from the idea of spatial pyramid match in [9] and learning weight associated with each local feature in [17]. Both strategies are able to augment the classification accuracy for every dataset, and we find that SPM provides additional improvement than learning weight in Scene-15 dataset but less improvement in the other two datasets. This is likely due to the geometric structure of Scene-15 that matches with the spatial equally divided subregions very well, while in the other two datasets discriminative local features for generating the weighted I2C distance have a more important role for classification. Nevertheless, by using both strategies we get the best results in all the three datasets.

Since a spatial single level at a finer resolution will reduce the computation cost required for feature NN search, we also compare its performance with spatial pyramid combining multiple levels as well as the original size without using spatial restriction. As shown in Figure 4, this spatial single level is able to improve the accuracy compared to no spatial restriction, not only on scene constraint datasets (Scene-15 and Corel) but also on Sports event dataset that does not have geometric scene structure. Though the performance is slightly lower than the pyramid combining all levels, it saves the computation cost for both feature NN search and distance metric learning. So this spatial single level strategy will be very useful for improving the efficiency.

We compare our best results with recently published result for every dataset. All the results are listed in Table 3. In Scene-15 dataset, many researchers reported their recent results and most of them also incorporate SPM to improve the accuracy. Lazebnik *et al.* [9] first proposed SPM and combined with the Bag-of-Word (BoW) strategy, achieving 81.4%. Bosch *et al.* [2] also incorporated SPM

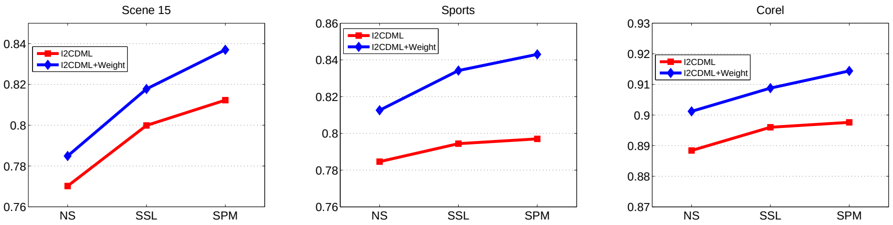

Fig. 4. Comparing the performance of no spatial restriction (NS), spatial single level restriction (SSL) and spatial pyramid match (SPM) for both I2CDML and I2CDML+Weight in all the three datasets. With only spatial single level, it achieves better performance than without spatial restriction, although slightly lower than spatial pyramid combination of multiple levels. But it requires much less computation cost for feature NN search.

Table 3. Comparing to recently published results for recognition accuracy (%)

Scene-15		Sports		Corel	
Ours	**83.7**	Ours	**84.3**	Ours	**91.4**
Lazebnik *et al.* [9]	81.4	Li *et al.* [10]	73.4	Lu *et al.* [13]	77.9
Liu *et al.* [11]	83.3	Wu *et al.* [21]	78.5	Lu *et al.* [14]	90.0
Bosch *et al.* [2]	83.7	Wu *et al.* [22]	84.2		
Yang *et al.* [24]	80.3				
Wu *et al.* [22]	84.1				

in pLSA to achieve 83.7%. The best result so far as we know is 84.1% by Wu *et al.* [22], who replace Euclidean distance by histogram intersection in BoW combined with SPM. Although our result is slightly lower than their results, we notice they have used multi-scale and denser grid to extract feature as well as combining additional Sobel gradient, while our feature extraction is very simple but still comparable to theirs. When using the same configuration, their approach is worse than ours, as either indicated in [22] as well as implemented by us using their published LibHIK[1] code, which is 81.36 ± 0.54 using CENTRIST [23] and 78.66 ± 0.44 using SIFT [12] in our implementation. For Sports dataset, Li *et al.* [10] who published them reported 73.4%, and Wu *et al.* improved it to 78.5% and 84.2% in [21] and [22] respectively, where the later one used the same configuration as their experiment in Scene-15. Nevertheless, our result is still comparable to theirs. For Corel dataset, our result is again better than the previous published results [14, 13] even without using color information. All these results show that our method can achieve state-of-the-art performance on these datasets using relatively small feature set.

[1] http://www.cc.gatech.edu/cpl/projects/libHIK/

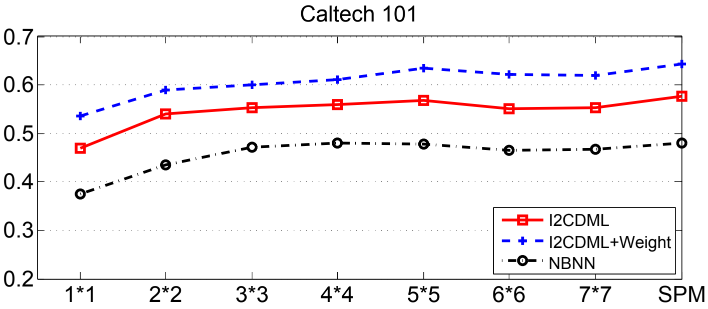

Fig. 5. Comparing the performances of I2CDML, I2CDML+Weight and NBNN from spatial division of 1× to 7×7 and spatial pyramid combination (SPM) on Caltech 101.

3.3 Results on Caltech 101 Dataset

We also evaluate our method on Caltech 101 to illustrate its scalability on datasets with more classes. In our experiment we only select 15 images per class for training, same as most previous studies. In [1], they extracted SIFT features using multi-scale patches densely sampled from each image, which result in much redundant features on the training set (about 15000 to 20000 features per image). So the NN search for I2C distance calculation takes expensive computation cost. Even using KD-Tree for acceleration, it takes about 1.6 seconds per class for the NN search of each test image [1] and thus around 160 seconds for 101 classes to classify only one test image. This is unacceptable during the testing phase and makes it difficult for real-world application. In our experiment, we only generate less than 1000 features per image on average using our feature extraction strategy, which are about 1/20 compared to the size of feature set in [1]. We also use single level spatial restriction to constrain the NN search for acceleration. For each image, we divide it from 2×2 to 7×7 subregions and test the performance of I2CDML, NBNN and I2CDML+Weight. Experiment results of without using spatial restriction (1×1 region) as well as spatial pyramid combining all levels is also reported.

From Figure 5 , we can see that without using spatial restriction, the performance of NBNN is surprisingly bad. The reason that NBNN performs excellently as reported in [1] using complex features while poorly in our small feature set implies that the performance of NBNN relies heavily on the large redundant of training feature set, which needs expensive computation cost for the NN search. For comparison, our I2CDML augments the performance of I2C distance significantly, while combining the learned weight further improves the performance. Compared to the other three datasets, this dataset contains more classes and less training images per class, which makes it more challenging. So the large improvement over NBNN indicates that our learning procedure plays an important role to maintain an excellent performance using small number of features under such challenging situation, which also requires much less computation cost in the testing phase.

From 2×2 to 7×7 subregions of spatial restriction, due to the regular geometric object layout structure of images in this dataset, the performance is further improved for all methods compared to without using spatial restriction. Though the results on spatial division from 3×3 to 7×7 do not change much, the computation cost for NN search continues decreasing with finer spatial division. For 7×7 spatial division, the size of feature set in the candidate class is 1/49 of the original image without using spatial restriction. The best result on single spatial restriction is 63.4% by I2CDML+Weight on 5×5 spatial division, which is close to the result of spatial pyramid combining all levels (64.4%) but is more efficient. NBNN can also benefit from this spatial restriction, but its result is still unacceptable for classification task using such small feature set.

Our best result is also comparable to the best reported result of NBNN (65%) in [1], which uses large number of densely sampled multi-scale local features as well as pixel location information for their I2C distances to achieve such state-of-the-art performance. The size of candidate feature set they used is about 20 times more than ours using the whole image and nearly 1000 times compared our spatial restriction of 7×7 subregions. So our implementation needs much less computation cost for the NN search during the on-line testing phase with the additional off-line training phase, whilst the result is comparable to theirs. Although we cannot reproduce their reported result in our implementation, we believe our comparison should be fair as we use the same feature set for all methods and the experiment has shown that our method achieves significant improvement on such large-scale dataset with much efficient implementation.

4 Conclusion

Image-To-Class distance relies heavily on the large number of local features in the training set and test images, which need heavy computation cost for the NN search in the testing phase. However, using small number of features results in poor performance. In this paper, we tried to improve the performance of this distance and speed up the testing phase. We added a training phase by proposing a distance metric learning method to learn the I2C distance. A large margin framework has been formulated to learn the Per-Class Mahalanobis distance metrics, with a gradient descent method to efficiently solve this optimization problem. We also discussed the method of enhancing the discrimination of the learned I2C distance for performance improvement. These efforts made the I2C distance perform excellent even using small feature set. For further accelerating the NN search in the testing phase, we adopted single level spatial restriction, which can speed up the NN search significantly while preserving the classification accuracy. The experiment results on four datasets of Scene-15, Sports, Corel and Caltech 101 verified that our I2CDML method can significantly outperform the original NBNN especially for those challenging categories, and such I2C distance achieved state-of-the-art performance on most datasets.

References

1. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR (2008)
2. Bosch, A., Zisserman, A., Munoz, X.: Scene classification using a hybrid generative/discriminative approach. TPAMI (2008)
3. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. CVPR Workshop on Generative-Model Based Vision (2004)
4. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR (2005)
5. Frome, A., Singer, Y., Malik, J.: Image retrieval and classification using local distance functions. In: NIPS (2006)
6. Frome, A., Singer, Y., Sha, F., Malik, J.: Learning globally-consistent local distance functions for shape-based image retrieval and classification. In: ICCV (2007)
7. van Gemert, J.C., Geusebroek, J.M., Veenman, C.J.: Kernel codebooks for scene categorization. In: ECCV (2008)
8. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: ICCV (2005)
9. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
10. Li, J., Fei-Fei, L.: What, where and who? Classifying events by scene and object recognition. In: ICCV (2007)
11. Liu, J., Shah, M.: Scene modeling using co-clustering. In: ICCV (2007)
12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2) (2004)
13. Lu, Z., Ip, H.H.: Image categorization by learning with context and consistency. In: CVPR (2009)
14. Lu, Z., Ip, H.H.: Image categorization with spatial mismatch kernels. In: CVPR (2009)
15. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV 42(3) (2001)
16. Rasiwasia, N., Vasconcelos, N.: Scene classification with low-dimensional semantic spaces and weak supervision. In: CVPR (2008)
17. Wang, Z., Hu, Y., Chia, L.T.: Learning instance-to-class distance for human action recognition. In: ICIP (2009)
18. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. In: NIPS (2005)
19. Weinberger, K.Q., Saul, L.K.: Fast solvers and efficient implementations for distance metric learning. In: ICML (2008)
20. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research 10, 207–244 (June 2009)
21. Wu, J., Rehg, J.M.: Where am I: Place instance and category recognition using spatial PACT. In: CVPR (2008)
22. Wu, J., Rehg, J.M.: Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In: ICCV (2009)
23. Wu, J., Rehg, J.M.: CENTRIST: A visual descriptor for scene categorization. Technical Report GIT-GVU-09-05, GVU Center, Georgia Institute of Technology (2009)
24. Yang, J., KaiYu, Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR (2009)