Scene Recognition with Naive Bayes Non-linear Learning

Marco Fornoni
Idiap Research Institute, Martigny, Switzerland
Ecole Polytechnique Fédérale (EPFL) Lausanne, Switzerland
Email: marco.fornoni@idiap.ch

Barbara Caputo
Idiap Research Institute, Martigny, Switzerland
Sapienza - Università di Roma, Italy
Email: barbara.caputo@idiap.ch

***Abstract*—A crucial feature of a good scene recognition algorithm is its ability to generalize. Scene categories, especially those related to human made indoor places or to human activities like sports, do present a high degree of intra-class variability, which in turn requires high robustness and generalization properties. Such features are amongst the distinctive characteristics of the Naive Bayes Nearest Neighbor (NBNN) approach [1], an image classification framework that since its introduction in 2008 has been gaining momentum in the visual recognition community. In this paper we show how with a straightforward modification of the original NBNN scoring function it is possible to use a recently introduced latent locally linear SVM algorithm to discriminatively learn a set of prototype local features for each class. The resulting classification algorithm, that we call Naive Bayes Non-linear Learning (NBNL) preserves the generality and robustness properties of the original approach, while greatly reducing its memory requirements during testing, and significantly improving its performance. To the best of our knowledge this is the first work to exploit the structure of the local features through the use of a latent locally linear discriminative learning method. Experiments over three different public scene recognition datasets show the effectiveness of the proposed algorithm, which outperforms several existing NBNN-based methods and is competitive with standard Bag-of-Words plus SVM approaches.**

I. Introduction

The dominating trend over the last decade in visual recognition has been the use of Bag of Words representations (BoW, [2]), combined with state of the art machine learning classifiers, ranging from max-margin algorithms [3] to Bayesian frameworks [4]. This general approach is crucially based on the assumptions that: 1) it is possible to determine the class of an image by computing image-to-image distances; 2) the representations based on vector quantization, or other forms of encoding are sufficient to describe the images. Since the seminal work of Boiman et al. in 2008, these two assumptions have been challenged with the introduction of the Naive Bayes Nearest Neighbor (NBNN) algorithm [1]. The NBNN classifier drops the vector quantization and the image-to-image distance computation in favor of an image-to-class approach. Hence, classes are directly represented by unordered sets of local features extracted from the training images, and a query image is classified by directly comparing its local features with those contained in each class-specific set of local descriptors. This results in a classification method that is competitive, performance-wise, with more established learning methods using BoW representations, while at the same time promising a high degree of robustness and generality when applied to categorization problems. This last feature of NBNN, and of NBNN-based methods (for a review of NBNN related works we refer the reader to Section II) is very appealing for scene recognition problems. Indeed, one of the greatest challenges in developing strong scene recognition algorithms lies in the intrinsic variability that scene images present, especially when moving from outdoor scenes [5] to scene of human-made, indoor environment [6], or scenes of human events like sports [7]. The community of researchers working on NBNN-based algorithms has acknowledged the potential of these methods for this specific application, by using more and more often several of the existing public databases for scene classification, as benchmarks to evaluate their approaches [8], [9], [10], [11]. The original NBNN algorithm does not perform any learning during training, as it simply stores all the available local features for all classes. While this makes the method attractively simple, it also leads to potential memory problems and scalability issues during testing. In this paper we propose a method for tackling these issues, while also improving the recognition performance of the algorithm. We build on a very recently introduced latent formulation of locally linear SVM [12], [13], and we show that with a moderate modification of the original NBNN scoring function it is possible to use this algorithm to learn an extremely compact set of prototypical local descriptors for each class. This new representation results in a greatly reduced memory footprint and computation time during testing, while also significantly increasing the predictive performance w.r.t. the original NBNN algorithm. We call the resulting algorithm Naive Bayes Non-linear Learning (NBNL). To assess our method, we perform experiments on three Scene Recognition datasets (Sports [7], 15-Scenes [3] and Indoor Scene Recognition [6]), comparing it with previously proposed NBNN-based algorithms and a BoW+SVM approach with a form of Spatially Local Coding [14]. Experiments show that NBNL significantly outperforms NBNN on all the datasets, while also achieving competitive or better performance than the BoW+SVM baseline and previously proposed NBNN-based algorithms. This shows the promise of our approach.

The rest of the paper is organized as follows: in Section II we review the most relevant previous works on NBNN. In section III we present our approach, directly deriving it from the NBNN algorithm. In Section IV we present the experimental results, before moving to conclusions in Section V.

II. Related works

Since its introduction, many authors have pointed out how the success of the NBNN algorithm [1] relies heavily on the large number of local descriptors in the training set, limiting its scalability to real-world applications [8], [15], [9], [16], [17]. Moreover, the somehow flat structure imposed on the space of local descriptors limits the expressiveness of the model, which tends to underperform methods based on learning [8], [9], [17]. Based on these observations, many authors have tried to exploit the structure of the local descriptors to improve the recognition performance, reduce the testing time, or the memory footprint of the algorithm. In Local NBNN [18], the class-conditional probability estimates for a given query descriptor are performed by restricting the search only to the training descriptors and classes present in its strict neighborhood. By ignoring the probability estimates for classes which do not lie in a neighborhood of the sample, the authors show an increase in the recognition performances and an improved scalability w.r.t. the number of classes. In [11], the authors propose to apply unsupervised learning (PCA) to the SIFT descriptors. This simple idea allows to compress the data and speed up the distance computation, while preserving or increasing the predictive performances. In LI2C [9], the Euclidean distance is replaced by a Mahalanobis distance and a supervised distance learning procedure is performed to learn a set of class-specific metrics. This results in an improved recognition performance, with good results obtained by using only five to ten percent of the training data. In the same work, the authors also proposed to apply the idea of spatial pyramid restriction (SPM) and force the features from a certain area of a query image to be matched only with training descriptors extracted from the same area. Instead of learning a metric for each class, the authors of [16] introduced a method to construct a kernel from the vanilla NBNN probability estimates and proposed to use it to train a SVM classifier. The main advantage of this approach is that it allows to integrate NBNN with existing kernel-based methods, for example by combining it with kernels based on BoW models. In a more recent work [19], each class is partitioned into several clusters and, for a given query image, an NBNN image-to-cluster distance is computed for each cluster. These NBNN image-to-cluster distances are then used to construct a richer NBNN kernel, resulting in improved performances w.r.t. [16]. Adopting a rather different approach, [10] proposes to learn a set of prototype descriptors for each class by training a class specific codebook. During prediction the NBNN image-to-class distances are then computed by using the learned codebooks instead of the complete training set. The method, coupled with a new spatial encoding, proved to be able to achieve very competitive performance, with reduced testing times and memory requirements. Finally, [17] modifies the NBNN scoring function, replacing the 1-NN patch-to-class distance computation with a k-NN approach (with $k > 1$), coupled with LLC encoding, Sparse Coding, or Collaborative Coding.

III. The NBNL approach

While many of the methods presented in Section II result in a performance increase w.r.t. the original NBNN algorithm, often also with reduced time and space complexities, only [10] produces a significantly more compact representation of the training data. In this paper we show how with a little modification of the NBNN scoring function it is possible to make use of a recently introduced local learning algorithm [12] to directly learn an extremely compact set of descriptor prototypes for each class, in a supervised discriminative fashion. By effectively exploiting the structure of the training patches, the proposed method greatly reduces the memory necessary to represent the training set, while also significantly increasing the classification accuracy and the testing speed.

A. The NBNN algorithm

Our method is based on the NBNN classifier. In the NBNN approach [1], the class of an image is estimated by a MAP approach. Let $\boldsymbol{X}_i = [\boldsymbol{x}_{i1} \quad \boldsymbol{x}_{i2} \quad \dots \quad \boldsymbol{x}_{in}]^\top \in \mathbb{R}^{n\times d}$ be a query image containing a set of n local descriptors $\boldsymbol{x}_{ij} \in \mathbb{R}^d$ and $\mathcal{Y} \triangleq \{1,\dots,c\}$ be a set of possible classes. If we assume that the class priors are uniform and that the local descriptors are conditionally independent given the class (Naive-Bayes assumption), the MAP estimate of the class of image $\boldsymbol{X}_i$ can be written as

$$\hat{y}_i = \operatorname*{arg\,min}_{y\in\mathcal{Y}} \; -\sum_{j=1}^{n} \log p(\boldsymbol{x}_{ij}|y). \tag{1}$$

$p(\boldsymbol{x}_{ij}|y)$ can be estimated using a kernel density estimator

$$\hat{p}(\boldsymbol{x}_{ij}|y) = \frac{1}{L_y h^d} \sum_{l_y=1}^{L_y} K\left(\frac{\boldsymbol{x}_{ij} - \boldsymbol{x}_{l_y}}{h}\right), \tag{2}$$

where $\boldsymbol{x}_{l_y}$ is the l-th local descriptor from class y, L_y is the total number of local descriptors in y, $K(\boldsymbol{x}) = (2\pi)^{-\frac{d}{2}} \exp\left(-\|\boldsymbol{x}\|^2\right)$ and h is the bandwidth parameter. This quantity is difficult to compute, because the number of local descriptors in a class y is huge. Nonetheless it can be reliably approximated by using only the single Nearest Neighbor $NN_y(\boldsymbol{x}_{ij})$ of $\boldsymbol{x}_{ij}$ in class y [1]. The NBNN classification rule thus becomes

$$\hat{y}_i \approx \operatorname*{arg\,min}_{y\in\mathcal{Y}} \sum_{j=1}^{n} \|\boldsymbol{x}_{ij} - NN_y(\boldsymbol{x}_{ij})\|^2. \tag{3}$$

The resulting classification algorithm is very simple and can achieve classification performance close to the one obtained by more complex BoW models [1]. However, as anticipated before, one main disadvantage of this algorithm is that it requires to store all the local descriptors of the training set, while its expected prediction complexity grows either linearly, or logarithmically (if the exact NN search is replaced with an approximated one [20]) w.r.t. the size of the training set.

B. The NBNL decision rule

Let L_y be the number of local descriptors in the training set of class y. In order to reduce the the memory requirements and the search space of the NBNN classifier it would be desirable to preselect a set of $m \ll L_y$ representative prototypes for each class y. Though a difficult task at first glance, the goal is achievable by making use of a recently introduced learning algorithm [12]. Let us call $\boldsymbol{W}_y$ the matrix $[\boldsymbol{w}_{y,1} \quad \boldsymbol{w}_{y,2} \quad \dots \quad \boldsymbol{w}_{y,m}]^\top \in \mathbb{R}^{m\times d}$ containing the set of prototype descriptors from class y and let us assume also that all the descriptors and prototypes are normalized to one (e.g.

SIFT descriptors are normalized by design). For a given testing sample $\boldsymbol{X}_i$, the NBNN prediction rule can be decomposed as

$$\hat{y}_i = \arg\max_{y\in\mathcal{Y}} s(\boldsymbol{X}_i, y) \tag{4a}$$

$$s(\boldsymbol{X}_i, y) = \sum_{j=1}^{n} f(\boldsymbol{x}_{ij}, y), \tag{4b}$$

$$f(\boldsymbol{x}_{ij}, y) = -\|\boldsymbol{x}_{ij} - NN_{\boldsymbol{W}_y}(\boldsymbol{x}_{ij})\|^2, \tag{4c}$$

where, once again, $NN_{\boldsymbol{W}_y}(\boldsymbol{x}_{ij})$ indicates the nearest neighbor of $\boldsymbol{x}_{ij}$ in $\boldsymbol{W}_y$. Using this notation we can rewrite $f(\boldsymbol{x}_{ij}, y)$ as

$$\begin{aligned} f(\boldsymbol{x}_{ij}, y) &= -\min_{k=\{1,\ldots,m\}} \|\boldsymbol{x}_{ij} - \boldsymbol{w}_{y,k}\|^2 \\ &= -\min_{k=\{1,\ldots,m\}} \|\boldsymbol{x}_{ij}\|^2 + \|\boldsymbol{w}_{y,k}\|^2 - 2\boldsymbol{w}_{y,k}^\top \boldsymbol{x}_{ij} \\ &= \max_{k=\{1,\ldots,m\}} 2\left(\boldsymbol{w}_{y,k}^\top \boldsymbol{x}_{ij} - 1\right), \end{aligned} \tag{5}$$

where for the last equality we have used the assumption that all the descriptors and prototypes are normalized to 1 (with this assumption it is also possible to see that $0 \leq \|\boldsymbol{x}_{ij} - \boldsymbol{w}_{y,k}\|^2 \leq 4, \forall i, j, y, k$). Removing the constants, the NBNN prediction rule can thus be equivalently written as

$$\hat{y}_i = \arg\max_{y\in\mathcal{Y}} \sum_{j=1}^{n} \max_{k=\{1,\ldots,m\}} \boldsymbol{w}_{y,k}^\top \boldsymbol{x}_{ij}. \tag{6}$$

As suggested also in [17], and especially since we assume to be using a matrix $\boldsymbol{W}_y$ with a highly reduced set of prototype descriptors, it would be advisable to use more than just the single closest prototype, when computing the score for a given descriptor. Taking inspiration from the scoring functions introduced in [12], we thus propose to search for a linear combination of all the prototypes in $\boldsymbol{W}_y$, maximizing the alignment of the combination with the considered patch. This idea can be formalized by the following objective function

$$f(\boldsymbol{x}_{ij}, y) = \max_{\boldsymbol{\beta}\geq 0,\, \|\boldsymbol{\beta}\|_p \leq 1} \boldsymbol{\beta}^\top \boldsymbol{W}_y \boldsymbol{x}_{ij}, \tag{7}$$

where $\boldsymbol{\beta}$ is an m-dimensional vector of coefficients that weights how the different prototypes in the matrix $\boldsymbol{W}_y$ are combined to compute $f(\boldsymbol{x}_{ij}, y)$ and p is a parameter controlling the sparsity of the combination (and, consequently, the smoothness of the classifier [12]). The first constraint in (7) is necessary to avoid that the vector $\boldsymbol{\beta}$ inverts the similarities between $\boldsymbol{x}_{ij}$ and the prototypes $\boldsymbol{w}_{y,k}$, while without the second constraint the maximization problem would be unbounded. As it can be seen, for each sample, (7) finds a local linear combination of the class-prototypes maximizing the alignment of the combination with the sample. An important property of (7) is that it has an analytical solution [12] that allows to efficiently compute the score for any given query sample and class. For example, it is easy to show that when $p = 1$, an optimal solution of (7) is of the form $\boldsymbol{\beta} = [0 \;\ldots\; 0 \;1\; 0 \;\ldots\; 0]$, where the only 1 is in the k-th position providing the maximum positive value for $\boldsymbol{w}_{y,k}^\top \boldsymbol{x}_{ij}$ [12]. Except for a constant factor, this is equivalent to the solve (5) with the additional requirement that, instead of searching amongst all the m prototypes of class y, we restrict the search to the closest ones (such that $\|\boldsymbol{x}_{ij} - \boldsymbol{w}_{y,k}\|^2 \leq 2$). If we instead allow p to vary in $(1, \infty)$, multiple $\boldsymbol{w}_{y,k}$ could take part in the linear combination. For example, with $p = 2$ the weight assigned to each prototype would be directly proportional to its similarity to $\boldsymbol{x}_{ij}$ [12]. Plugging (7) into (4) our decision rule is finally defined as

$$\hat{y}_i = \arg\max_{y\in\mathcal{Y}} s(\boldsymbol{X}_i, y) \tag{8a}$$

$$s(\boldsymbol{X}_i, y) = \sum_{j=1}^{n} f(\boldsymbol{x}_{ij}, y) \tag{8b}$$

$$f(\boldsymbol{x}_{ij}, y) = \max_{\boldsymbol{\beta}\geq 0,\, \|\boldsymbol{\beta}\|_p \leq 1} \boldsymbol{\beta}^\top \boldsymbol{W}_y \boldsymbol{x}_{ij}. \tag{8c}$$

C. Learning the NBNL prototypes

As anticipated before, in order to be able to efficiently represent the training set and efficiently predict the class of a query image, we need to learn the matrix $\boldsymbol{W}_y$ for each class. This can actually be achieved by making use of the recently introduced Multiclass Latent Locally Linear SVM (ML3) algorithm [12], which is a multi-class local classifier based on a latent SVM formulation [21]. The aim of ML3 is to learn a smooth non-linear classifier as a local linear combination of linear ones. For a query instance, the linear sub-models of each class are locally combined according to their confidence on the sample. This choice is motivated by the intuition that, if locally trained, the most confident sub-models are the most useful in predicting the label of a testing sample. A main advantage of this approach is that it allows to efficiently train and test powerful non-linear classifiers, without the computational complexity and memory requirements of kernels, or the computational burden and architectural complexity of multi-layer architectures.

Let $\boldsymbol{W}^\top \triangleq [\boldsymbol{W}_1 \quad \boldsymbol{W}_2 \quad \ldots \quad \boldsymbol{W}_c] \in \mathbb{R}^{mc\times d}$, where $\boldsymbol{W}_y$ contains the prototypes for class y. The prediction of the ML3 algorithm is defined as $\hat{y}_i \triangleq \arg\max_{y\in\mathcal{Y}} f_{\boldsymbol{W}}(\boldsymbol{x}_i, y)$, where

$$f_{\boldsymbol{W}}(\boldsymbol{x}_i, y) \triangleq \max_{\boldsymbol{\beta}\geq 0,\, \|\boldsymbol{\beta}\|_p \leq 1} \boldsymbol{\beta}^\top \boldsymbol{W}_y \boldsymbol{x}_i. \tag{9}$$

The multiclass objective function of ML3 is then defined as

$$\min_{\boldsymbol{W},\boldsymbol{\xi}} \frac{\lambda}{2}\|\boldsymbol{W}\|_F^2 + \sum_{i=1}^{n} \xi_i \tag{10a}$$

$$\text{s.t. } 1 - \left(f_{\boldsymbol{W}}(\boldsymbol{x}_i, y_i) - \max_{y\neq y_i} f_{\boldsymbol{W}}(\boldsymbol{x}_i, y)\right) \leq \xi_i, \tag{10b}$$

$$\xi_i \geq 0,\, i = 1, \ldots, n \tag{10c}$$

where $\|\boldsymbol{W}\|_F$ is the Frobenius norm of $\boldsymbol{W}$. The non-convex learning problem (10) is solved using a CCCP procedure, coupled with a stochastic gradient descent approach [12].

As it can be seen, equation (9) has exactly the same form of (7), so that we can directly make use of the scores provided by the ML3 algorithm into the NBNN-like scoring function (8). More importantly, we can also make use of the ML3 algorithm to learn the matrices $\boldsymbol{W}_y$. The task assigned to the ML3 algorithm is thus to learn how to predict the class of any single local descriptor, by discriminatively training the matrices $\boldsymbol{W}_y$ on the local descriptors collected from all the training images of each class. Though this is a very hard task, it does not need to be solved exactly, as during the prediction phase the Naive Bayes classifier running on top of the ML3 algorithm can correct the mistakes made by the latter.

Since we still make use of the Naive Bayes assumption while have replaced the Nearest-Neighbor distance with the score provided by a non-linear learning algorithm, we call our approach Naive Bayes Non-linear Learning (NBNL). The proposed algorithm adopts the promising image-to-class distance paradigm and combines it with a discriminative training phase to produce a compact representation of the training data. This results in a remarkable reduction of the memory requirements during prediction and a significative improvement in the classification accuracy, as it will be demonstrated in the next section.

IV. Experiments

In this section we report the results obtained by NBNL and we compare it against BoW with χ^2 or intersection kernel and 512 visual words, NBNN [1], NBNN applied to PCA-SIFT [11], and with results reported in the NBNN literature. We also compare against a simple One-vs-All linear SVM trained on the local descriptors, while we do not attempt to use a canonical kernel-SVM to classify the patches, as it would not scale to the millions of descriptors that we are dealing with.

We perform experiments on three widely used scene recognition datasets (sample images in Figure 2): 1) the 8-Sports dataset [7], collecting scenes from eight different sports, with 137 to 250 images per category; 2) the 15-Scenes [3] dataset, containing 4485 low-resolution images from fifteen indoor and outdoor categories; 3) the Indoor Scene Recognition (ISR) dataset [6], consisting of 15620 images collected from the web and belonging to 67 different indoor categories, with a minimum of 100 images per category. The standard benchmarking procedure for the Sports dataset consists in selecting 70 images per class for training and 60 for testing. For the 15-Scenes dataset the default benchmark consists in randomly selecting 100 training images per class and using the remaining ones for testing. Finally, for ISR the procedure requires to select 100 images per category and split them into 80 images for training and 20 for testing.

For all our experiments (and for all the algorithms) we use a common feature extraction procedure. We initially rescale all the images so that their smallest dimension is equal to 200px (keeping the original aspect ratio), in order to enforce scale consistency. We favor SIFT descriptors over other ones (e.g. NIMBLE [22], as suggested by [17]) to fairly compare with the wide majority of NBNN and BoW methods. We thus use VLFeat [23] to densely extract SIFT features every 8px, using four different patch sizes: 16, 24, 36 and 54 pixels. As in [1], [19], [17], [16], [11], we also augment the features by concatenating to each descriptor the coordinates of its relative position in the image and we finally normalize each descriptor to one. Using these features in a standard BoW model results in an approach close to the recently introduced Spatially Local Coding [14], in which the patches in the dictionary include also an expected location. We thus name our Bag-of-Words baseline *SLC-BoW* to underline its difference w.r.t. the vanilla BoW model, lacking any spatial information.

We perform two sets of experiments. In the first one we make use of a reduced feature set and analyze the performance of the NBNL algorithm while varying the number of prototypes, the parameter p and the number of descriptors in the training set. We then perform a second set of experiments by using the full feature set on a fixed configuration of the NBNL algorithm.

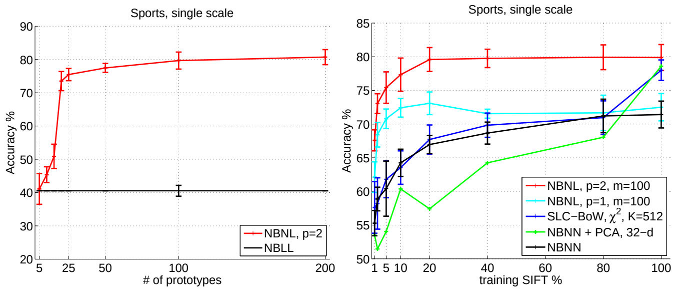

Fig. 1. Left: performance of our method varying the number of prototypes, w.r.t. the NBLL baseline (using a One-Vs-All linear SVM). Right: performances of our method and several other baselines with an asymmetric sampling strategy (sub-sampling the patches only from the training images). All the results are obtained on the Sports dataset, using single-scale descriptors.

A. Single-scale experiments

In our first set of experiments we analyze the performance of our method when using single-resolution SIFT features, as obtained by employing only descriptors with a patch size of 16px. With this configuration the total number of training descriptors for the Sports dataset is around 500,000. In this scenario the amount of information provided by the features is relatively limited and a good classifier is fundamental to achieve reasonable performance. In Figure 1-Left we use the Sports dataset to compare our approach with a simple Naive Bayes Linear Learning (NBLL) algorithm, in which we replace ML3 with a One-Vs-All linear SVM. Each experiment is repeated five times, on five different random splits, while the regularization parameter is tuned using 5-fold cross-validation. The average accuracy is plotted together with the standard deviation. As it can be noted, the non-linearity introduced by the ML3 classifier results in an impressive $+39\%$ absolute improvement w.r.t. the linear classifier. Intuitively, learning only a single prototype per class is not sufficient to accurately represent the complexity of the local descriptors. On the other hand, by learning a set of m prototypes per class and predicting with a sample-specific linear combination of them, each NBNL class model can represent a wide range of descriptors, resulting in greatly improved performances. With as few as 20 prototypes our method already achieves competitive results, while 100 prototypes are sufficient to obtain maximal performance.

In Figure 1-Right we evaluate the robustness of our approach against several other methods, when applying an asymmetric sampling strategy, as advocated in [17]. We randomly sub-sample the training descriptors by keeping only a given percentage of descriptors/image and we run experiments with each setting. We plot the results of our method with $p \in \{1, 2\}$, together with the results of SLC-BoW, NBNN and NBNN + PCA. Following the observations presented in Figure 1-Left, we fix the number of NBNL prototypes to $m = 100$. As before, each experiment is repeated five times and the regularization parameter is tuned using cross-validation. The average accuracy is then plotted, together with the standard deviation. As it is possible to see, amongst the considered methods the proposed NBNL approach results to be the most robust w.r.t. sub-sampling the training patches. By using as little as 2% of the training samples NBNL can already reach the performance level of the NBNN algorithm using the full training data, while superior performances can be achieved using only 10% of the training data. Moreover, with just 20%

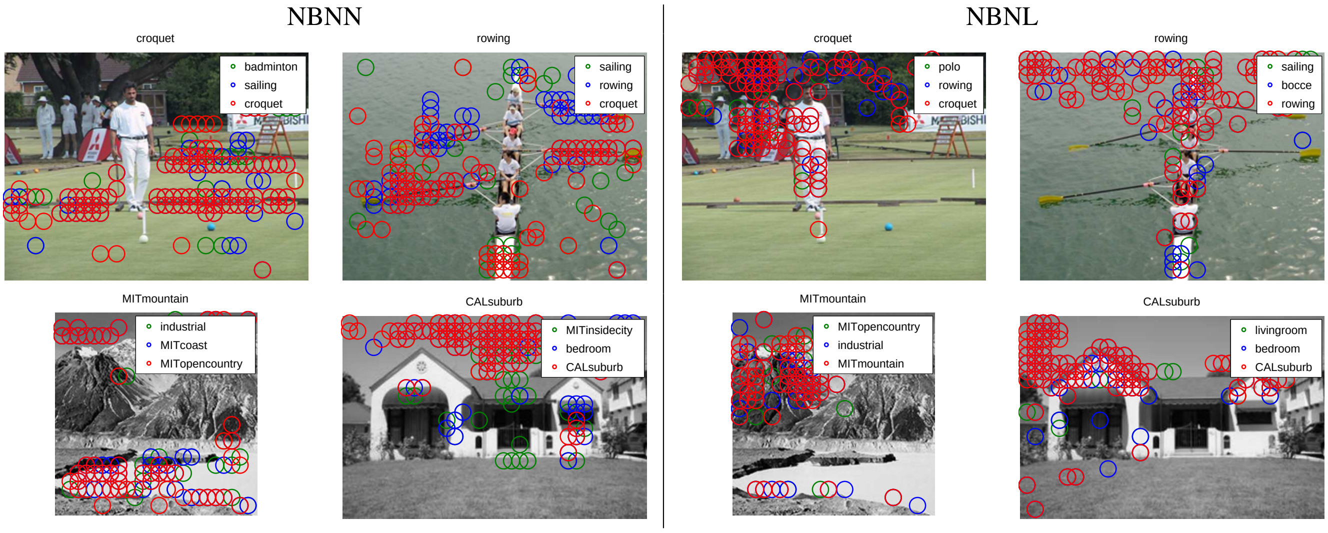

Fig. 2. Visualization of the classification results on images of the Sports (top) and the 15-Scenes (bottom) datasets. Each image is titled with its ground-truth label, while in green, blue and red are visualized the top-scoring SIFT patches for the three top-scoring classes (from lowest to highest) of each image.

of the training descriptors our method is already able to reach its maximal performance. When all the training descriptors are preserved, the performance of NBNL with $p = 2$ is similar to that of the SLC-BoW and the NBNN + PCA baselines, significantly outperforming both NBNL with $p = 1$ and NBNN. We also note that, while with $p = 1$ the NBNL algorithm still outperforms the original NBNN, setting $p = 2$ (and thus allowing for multiple prototypes to take part in the prediction) significantly improves the results. Finally, while [12] advocates for setting $p = 1.5$, we did not observe any empirical advantage over $p = 2$ on our problem (we omit the curve for clarity of presentation). We thus opt for $p = 2$, as it slightly speeds up the training procedure.

For visualization purposes, in Figure 2 we also plot the top-scoring patches selected by the original NBNN algorithm and the proposed NBNL approach on example images of the Sports and the 15-Scenes datasets. As it can be noted, NBNL favors patches lying in the most discriminative areas, correcting some of the mistakes made by the vanilla NBNN algorithm. For example, water is a more discriminative cue than paddles, as they are easily confused with field delimitation rods and mallets used in croquet games. We also note that on the Sports dataset our algorithm has learned a spatial bias towards the patches lying on the top of the scene (rich of contextual data).

B. Multi-scale experiments

TABLE I. RESULTS OF NBNL USING MULTI-SCALE SIFT DESCRIPTORS, COMPARED TO NBNN BASELINES ON THE SAME FEATURE SET, NBNN RESULTS REPORTED IN THE LITERATURE (WITH CITATION) AND TWO SLC-BOW ALGORITHMS ON THE SAME FEATURE SET (TOP).

	Sports	**15-Scenes**	**ISR**
SLC-BoW (Intersection, K=512)	84.58±2.61	79.99±0.55	37.54±2.24
SLC-BoW (χ^2, K=512)	86.54±2.27	81.31±0.39	40.92±0.89
NBNN [9]	67.60±1.1	72.8±0.7	-
NBNN + NIMBLE [17]	-	74.2±1.0	-
NBNN [16]	-	75±3	-
NBNN [11]	81.48	-	-
NB-INN + NIMBLE [17]	-	78.2±1.0	-
NNbMF [10]	-	78.99	42.46
NBNN + PCA (32-d) [11]	84.67	79.0	**48.84**
NBNN-kernel [16]	-	79±2	-
Pooled NBNN + NIMBLE [19]	-	79.7±1.5	-
NB-INN + G-KDES + PCA [17]	-	79.8±1	-
NBNN + LI2C [9]	82.07±1.2	80.07±0.4	-
NBNN	80.08±1.94	77.25±0.74	38.67±1.58
NBNN + PCA (32-d)	85.50±1.73	80.53±0.56	45.76±2.33
NBNL	**85.54**±2.81	**82.42**±0.63	42.15±1.60

For our final set of experiments we benchmark our algorithm on all the three scene recognition datasets, against all the considered baselines, using all the training descriptors with the full multi-scale setup. Using this configuration the total number of training SIFT descriptors amounts to about 1,950,000 for the Sports dataset, 3,690,000 for 15-Scenes and more than 16,000,000 for ISR. Training a multi-class classifier on such a large number of samples can be a challenge. For the ISR dataset we thus train the NBNL algorithm in single-precision and using a One-vs-One approach, which decomposes the problem into a number of very small binary problems, allowing for massive parallelization. For the other two datasets the original multi-class training procedure is used. Following the results presented in section IV-A all the NBNL experiments are performed using $p = 2$ and $m = 100$. Each experiment is repeated five times, while the regularization parameter is tuned using 5-fold cross-validation. In Table I we report the average accuracy and the standard deviation of the algorithms implemented in our benchmark, together with the results reported in the NBNN-related literature. We focus on results that do not make use of spatial pyramid, or other types of spatial coding that could be combined with the methods (NBNN, BoW and NBNL) used in our benchmark. As it has been repeatedly reported in the past [15], [10], [9], [24], any enhanced spatial encoding can further improve the performance of image classification algorithms, and assessing what is the best way to encode the spatial information in the feature representation goes beyond the scope of this work.

As it can be seen, even when using a rich multi-scale representation our approach outperforms all the other NBNN algorithms on two out of three datasets, while being also competitive with the SLC-BoW baselines. Despite its simplicity, the NBNN + PCA approach seems to be a very good performer on the ISR dataset, though the difference is less marked using our feature set. We note also that in [11] the performance of the original NBNN is not reported for the ISR dataset, making it difficult to properly evaluate the impact of the raw features on their final results. It is important to underline that our approach also produces an extremely compact representation of the original training set. For example, for the Sports dataset (with the multi-scale setup) we have measured a memory footprint of less than 830 kilobytes for our model, while the original training data requires around 1.9 gigabytes to be stored in double precision, or about 475 megabytes in a PCA compressed format. This amounts to a three orders of magnitude compression w.r.t. the original feature set and more than two orders of magnitude w.r.t. the PCA-compressed representation. The reason for this compression lies in the fact that our representation contains $8 \times 100 = 800$ prototypes in total (100 prototypes per class), instead of the almost two millions in the original set. This is acheived at the cost of a training procedure that for this dataset takes 3 hours on average (on a single thread of an Intel(R) Core(TM) i7-2600K with 16GB of RAM). Despite this relatively expensive training procedure, another advantage of our approach w.r.t. the original NBNN algorithm lies in a highly reduced testing time. For example, with the multi-scale setup the average time necessary to evaluate our algorithm on all the testing images of the Sports dataset is of 51 seconds, while the NBNN algorithm implemented using a fast approximated nearest neighbor approach [25] requires more than 20 minutes on average (17 minutes with PCA). This corresponds to a reduction of more than one order of magnitue in the testing time.

V. Conclusions

In this paper we presented the first NBNN-based algorithm that combines the image-to-class distance approach with the power of local discriminative training. We achieve this by proposing a modified version of the scoring function of the original NBNN classifier, and by using a latent locally linear SVM formulation to learn a set of prototypical local features for each class. By effectively harnessing the training data in a discriminative framework, the proposed approach provides two main advantages: 1) the memory footprint and computation time during prediction are reduced by more than one order of magnitude; 2) the recognition performance is significantly increased. Experiments on three public scene recognition databases show the potential of the proposed method.

The main limitation of the approach in its current form is that it presents a computationally intensive training procedure, due to the relatively slow convergence of the CCCP optimization. In the future we will try to attack this issue by exploring the possibility to use other non-latent locally linear methods, such as [13], or other efficient non-linear learning methods, such as explicit approximation of the Gaussian Kernel using Random Features [26]. Another promising direction could be to combine our approach with methods that directly enforce a spatial structure on the problem, as suggested in [9]. Particularly for the scene classification task, adding this feature to the algorithm could lead to an increase of performance, as shown in [24]. Future work will explore these research avenues.

Acknowledgment

This work was partially supported by the vision@home –SIVI project (B.C.)

References

[1] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in Proc. of CVPR. IEEE, 2008, pp. 1–8.

[2] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in Workshop on statistical learning in computer vision, ECCV, vol. 1, 2004, p. 22.

[3] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in Proc. of CVPR, vol. 2, 2006.

[4] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in Proc. of CVPR, vol. 2. IEEE, 2005, pp. 524–531.

[5] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," International journal of computer vision, vol. 42, no. 3, pp. 145–175, 2001.

[6] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in In Proc. CVPR. IEEE, 2009.

[7] L. Li and L. Fei-Fei, "What, where and who? classifying events by scene and object recognition," in Proc. of ICCV. IEEE, 2007.

[8] R. Behmo, P. Marcombes, A. Dalalyan, and V. Prinet, "Towards optimal naive bayes nearest neighbor," in Proc. of ECCV. Springer, 2010, pp. 171–184.

[9] Z. Wang, Y. Hu, and L.-T. Chia, "Improved learning of i2c distance and accelerating the neighborhood search for image classification," Pattern Recognition, vol. 44, no. 10, pp. 2384–2394, 2011.

[10] F. Cakir, U. Güdükbay, and Ö. Ulusoy, "Nearest-neighbor based metric functions for indoor scene recognition," Computer Vision and Image Understanding, vol. 115, no. 11, pp. 1483–1492, 2011.

[11] S. N. Vitaladevuni, P. Natarajan, S. Wu, X. Zhuang, R. Prasad, and P. Natarajan, "Scene image categorization and video event detection using naive bayes nearest neighbor," in Proc. of WACV. IEEE, 2013, pp. 140–147.

[12] M. Fornoni, B. Caputo, and F. Orabona, "Multiclass latent locally linear support vector machines," in JMLR W&CP, Volume 29: ACML, C. S. Ong and T.-B. Ho, Eds., 2013, pp. 229–244.

[13] L. Ladicky and P. H. S. Torr, "Locally linear support vector machines," in Proc. of ICML, 2011, pp. 985–992.

[14] S. McCann and D. G. Lowe, "Spatially local coding for object recognition," in Proc. of ACCV. Springer, 2012, pp. 204–217.

[15] Z. Wang, Y. Hu, and L.-T. Chia, "Image-to-class distance metric learning for image classification," in Proc. of ECCV. Springer, 2010, pp. 706–719.

[16] T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell, "The nbnn kernel," in Proc. of ICCV. IEEE, 2011, pp. 1824–1831.

[17] R. Timofte, T. Tuytelaars, and L. Van Gool, "Naive bayes image classification: beyond nearest neighbors," in Proc. of ACCV. Springer, 2012, pp. 689–703.

[18] S. McCann and D. G. Lowe, "Local naive bayes nearest neighbor for image classification," in Proc. of CVPR. IEEE, 2012, pp. 3650–3656.

[19] K. Rematas, M. Fritz, and T. Tuytelaars, "The pooled nbnn kernel: beyond image-to-class and image-to-image," in Proc. of ACCV. Springer, 2012, pp. 176–189.

[20] S. Arya and H.-Y. A. Fu, "Expected-case complexity of approximate nearest neighbor searching," SIAM Journal on Computing, vol. 32, no. 3, pp. 793–815, 2003.

[21] C.-N. J. Yu and T. Joachims, "Learning structural svms with latent variables," in Proc. of ICML. New York, NY, USA: ACM, 2009, pp. 1169–1176.

[22] C. Kanan and G. Cottrell, "Robust classification of objects, faces, and flowers using natural image statistics," in Proc. of CVPR. IEEE, 2010, pp. 2472–2479.

[23] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[24] M. Fornoni and B. Caputo, "Indoor scene recognition using task and saliency-driven feature pooling," in Proc. of BMVC, 2012.

[25] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in Proc. of VISSAPP. INSTICC Press, 2009, pp. 331–340.

[26] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in Advances in neural information processing systems, 2007, pp. 1177–1184.