In Defense of Nearest-Neighbor Based Image Classification

Oren Boiman
The Weizmann Institute of Science
Rehovot, ISRAEL

Eli Shechtman
Adobe Systems Inc. &
University of Washington

Michal Irani
The Weizmann Institute of Science
Rehovot, ISRAEL

Abstract

State-of-the-art image classification methods require an intensive learning/training stage (using SVM, Boosting, etc.) In contrast, non-parametric Nearest-Neighbor (NN) based image classifiers require no training time and have other favorable properties. However, the large performance gap between these two families of approaches rendered NN-based image classifiers useless.

We claim that the effectiveness of non-parametric NN-based image classification has been considerably undervalued. We argue that two practices commonly used in image classification methods, have led to the inferior performance of NN-based image classifiers: (i) Quantization of local image descriptors (used to generate "bags-of-words", codebooks). (ii) Computation of 'Image-to-Image' distance, instead of 'Image-to-Class' distance.

We propose a trivial NN-based classifier – NBNN, (Naive-Bayes Nearest-Neighbor), which employs NN-distances in the space of the local image descriptors (and not in the space of images). NBNN computes direct 'Image-to-Class' distances without descriptor quantization. We further show that under the Naive-Bayes assumption, the theoretically optimal image classifier can be accurately approximated by NBNN.

Although NBNN is extremely simple, efficient, and requires no learning/training phase, its performance ranks among the top leading learning-based image classifiers. Empirical comparisons are shown on several challenging databases (Caltech-101,Caltech-256 and Graz-01).

1. Introduction

The problem of image classification has drawn considerable attention in the Computer Vision community. The concentrated effort of the research community in the last few years resulted in many novel approaches for image classification, that progressed the field quickly in a few years. For instance, in a course of three years, the classification rate on the Caltech-101 database climbed from under 20% in 2004 [8] to almost 90% in 2007 [27].

Image classification methods can be roughly divided into two broad families of approaches: (i) **Learning-based classifiers**, that require an intensive learning/training phase of the classifier parameters (e.g., parameters of SVM [6, 12, 13, 15, 16, 18, 20, 26, 27, 31], Boosting [24], parametric generative models [8, 10, 29], decision trees [5], fragments and object parts [2, 9], etc.) These methods are also known as *parametric methods*. The leading image classifiers, to date, are learning-based classifiers, in particular SVM-based methods (e.g., [6, 20, 27]). (ii) **Non-parametric classifiers**, that base their classification decision directly on the data, and require *no learning/training* of parameters. The most common non-parametric methods rely on Nearest-Neighbor (NN) distance estimation, referred to here as "NN-based classifiers". A special case of these is the "Nearest-Neighbor-Image" classifier (in short - "NN-Image"), which classifies an image by the class of its nearest (most similar) image in the database. Although this is the most popular among the NN-based image classifiers, it provides inferior performance relative to learning-based methods [27].

Non-parametric classifiers have several very important advantages that are not shared by most learning-based approaches: (i) Can naturally handle a huge number of classes. (ii) Avoid overfitting of parameters, which is a central issue in learning based approaches. (iii) Require no learning/training phase. Although training is often viewed as a one-time preprocessing step, retraining of parameters in large dynamic databases may take days, whereas changing classes/training-sets is instantaneous in non-parametric classifiers.

Despite these advantages, the large performance gap between non-parametric NN-image classifiers and state-of-the-art learning-based approaches led to the perception that non-parametric image classification (in particular NN-based image classification) is not useful. We claim that the capabilities of non-parametric image classification have been considerably under-valued. We argue that two practices, that are commonly used in image classification, lead to significant degradation in the performance of non-parametric image classifiers:

(i) *Descriptor quantization:* Images are often represented by the collection of their local image descriptors (e.g., SIFT [19], Geometric-Blur (GB) [30], image patches, etc.) These are often quantized to generate relatively small

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

"codebooks" (or "bags-of-words"), for obtaining compact image representations. Quantization gives rise to a significant dimensionality reduction, but also to significant degradation in the discriminative power of descriptors. Such dimensionality reduction is essential for many learning-based classifiers (for computational tractability, and for avoiding overfitting). However, it is *unnecessary and especially harmful* in the case of non-parametric classification, that has no training phase to compensate for this loss of information. (ii) *'Image-to-Image' distance* is essential to Kernel methods (e.g., SVM). When used in NN-Image classifiers, it provides good image classification only when the query image is similar to one of the database images, but does not generalize much beyond the labelled images. This limitation is especially severe for classes with large diversity.

In this paper we propose a remarkably simple non-parametric NN-based classifier, which requires no descriptor quantization, and employs a direct "Image-to-Class" distance. We show that under the Naive-Bayes assumption[1], *the theoretically optimal image classifier* can be accurately approximated by this simple algorithm. For brevity, we refer to this classifier as **"NBNN"**, which stands for "Naive-Bayes Nearest-Neighbor".

NBNN is embarrassingly simple: Given a query image, compute all its local image descriptors $d_1, ..., d_n$. Search for the class C which minimizes the sum $\sum_{i=1}^{n} \| d_i - \mathrm{NN}_C(d_i) \|^2$ (where $\mathrm{NN}_C(d_i)$ is the NN-descriptor of d_i in class C). Although NBNN is extremely simple and requires no learning/training, its performance ranks among the top leading learning-based image classifiers. Empirical comparisons are shown on several challenging databases (Caltech-101,Caltech-256 and Graz-01).

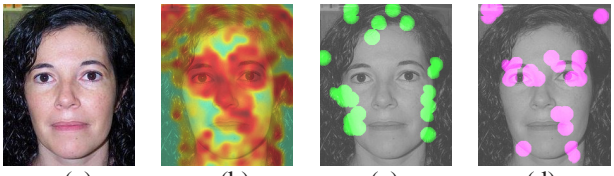
The paper is organized as follows: Sec. 2 discusses the causes for the inferior performance of standard NN-based image classifiers. Sec. 3 provides the probabilistic formulation and the derivation of the optimal Naive-Bayes image classifier. In Sec. 4 we show how the optimal Naive-Bayes classifier can be accurately approximated with a very simple NN-based classifier (NBNN). Finally, Sec. 5 provides empirical evaluation and comparison to other methods.

2. What degrades NN-image classification?

Two practices commonly used in image classification lead to significant degradation in the performance of non-parametric image classifiers.

2.1. Quantization damages non-parametric classifiers

Descriptor quantization is often used to generate codebooks (or "bags-of-words") for obtaining compact image representations (e.g., compact histograms of quantized descriptors). A large set of features/descriptors taken from the data (typically hundreds of thousands of descriptors extracted from the training images), is quantized to a rather small codebook (typically into $200 - 1000$ representative descriptors). Lazebnik et al. [16] further proposed to add rough quantized location information to the histogram representation. Such coarsely-quantized descriptor codebooks are necessary for practical use of SVM-based methods for image classification [6, 16, 27]. Such quantized codebooks were also used in the NN-image classification methods compared to in [27].



(a) (b) (c) (d)

Figure 1. **Effects of descriptor quantization – Informative descriptors have low database frequency, leading to high quantization error.** *(a) An image from the Face class in Caltech101. (b) Quantization error of densely computed image descriptors (SIFT) using a large codebook (size $6,000$) of Caltech-101 (generated using [14]). Red = high error; Blue = low error. The most informative descriptors (eye, nose, etc.) have the highest quantization error. (c) Green marks the 8% of the descriptors in the image that are most frequent in the database (simple edges). (d) Magenta marks the 8% of the descriptors in the image that are least frequent in the database (mostly facial features).*

However, the simplicity and compactness of such a quantized codebook representation comes with a high cost: As will be shown next, the amount of discriminative information is considerably reduced due to the rough quantization. Learning-based algorithms can compensate for some of this information loss by their learning phase, leading to good classification results. This, however, is not the case for simple non-parametric algorithms, since they have no training phase to "undo" the quantization damage.

It is well known that highly frequent descriptors have low quantization error, while rare descriptors have high quantization error. However, the most frequent descriptors in a large database of images (e.g., Caltech-101) comprise of *simple edges and corners* that appear abundantly in all the classes within the database, and therefore are least informative for classification (provide very low class discriminativity). In contrast, the most informative descriptors for classification are the ones found in one (or few) class, but are rare in other classes. These *discriminative descriptors tend to be rare* in the database, hence get high quantization error. This problem is exemplified in Fig. 1 on a face image from Caltech-101, even when using a relatively large codebook of quantized descriptors.

As noted before [14, 26], when densely sampled image descriptors are divided into fine bins, the bin-density follows a power-law (also known as long-tail or heavy-tail distributions). This implies that most descriptors are infrequent (i.e., found in low-density regions in the descriptor

[1] Naive Bayes assumption: Image descriptors are i.i.d. given image class.

space), therefore rather isolated. In other words, there are almost no 'clusters' in the descriptor space. Consequently, any clustering to a small number of clusters (even thousands) will inevitably incur a very high quantization error in most database descriptors. Thus, *such long-tail descriptor distribution is inherently inappropriate for quantization*.

High quantization error leads to a drop in the discriminative power of descriptors. Moreover, *the more informative (discriminative) a descriptor is, the more severe the degradation in its discriminativity*. This is shown quantitatively in Fig. 2. The graph provides an evidence to the severe drop in the discriminativity (informativeness) of the (SIFT) descriptors in Caltech-101 as result of quantization. The descriptor discriminativity measure of [2, 26] was used: $p(d|C)/p(d|\overline{C})$, which measures how well a descriptor d discriminates between its class C and all other classes $\overline{C}$. We compare the average discriminativity of all descriptors in all Caltech-101 classes after quantization: $p(d_{quant}|C)/p(d_{quant}|\overline{C})$, to their discriminativity before quantization.

Alternative methods have been proposed for generating compact codebooks via informative feature selection [26, 2]. These approaches, however, discard all but a small set of highly discriminative descriptors/features. In particular, they discard all descriptors with *low-discriminativity*. Although individually such descriptors offer little discriminative power, there is a *huge number* of such descriptors [26]. When considered in unison, they offer significant discriminative power (this is like having a huge ensemble of 'very weak classifiers'). This discriminative power is not exploited when using sparse informative feature selection.

In other words, both quantization and informative feature selection on a long-tail distribution will incur a large information loss. In contrast, we propose (Sec. 4) an alternative approach to efficiently approximate the descriptor distribution, without resorting to quantization or feature selection. This is achieved by using NN-distances in descriptor space, and is shown to be highly suitable for long-tail distributions. Our NBNN algorithm, which employs this approximation, *can exploit the discriminative power of both (few) high and (many) low informative descriptors.*

In Sec. 5 we empirically show that quantization is one of the major sources for inferior performance in non-parametric image classification. Unlike most reported NN-image classifiers, the one reported in Berg et al. [30] refrained from descriptor quantization and used the raw unquantized image descriptors (Geometric Blur). Nevertheless, their NN-Image classifier still provided low performance relative to their SVM-KNN method. We suggest that the main reason for this gap is the use of "Image-to-Image" distances, as explained next.

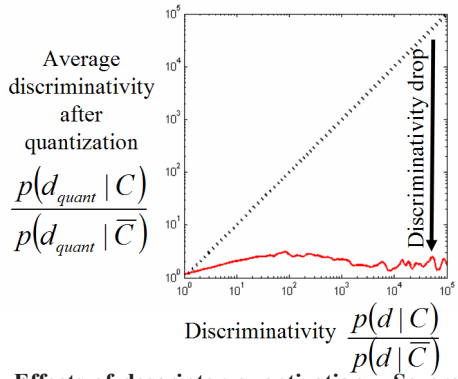


Figure 2. **Effects of descriptor quantization – Severe drop in descriptor discriminative power.** *We generated a scatter plot of descriptor discriminative power before and after quantization (for a very large sample set of SIFT descriptors d in Caltech-101, each for its respective class C). We then averaged this scatter plot along the y-axis. This yields the "Average discriminative power after quantization" (the RED graph). The display is in logarithmic scale in both axes. NOTE: The more informative (discriminative) a descriptor d is, the larger the drop in its discriminative power.*

2.2. Image-to-Image vs. Image-to-Class Distance

In this section we argue that "Image-to-Image" distance, which is fundamental to Kernel methods (e.g., SVM, RVM), significantly limits the generalization capabilities of non-parametric image classifiers when the number of labelled ('training') images is small.

NN-image classifiers provide good image classification when the query image is similar to one of the labelled images in its class. Indeed, NN-image classifiers have proved to be highly competitive in restricted image classification domains (e.g., OCR and Texture Classification [30]), where the number of labelled database images is very high relative to the class complexity. From a theoretical point of view, *NN classification tends to the Bayes optimal classifier as the sample size tends to infinity* [7].

However, NN-image classifiers cannot generalize much beyond the labelled image set. In many practical cases, the number of "samples" (the number of training/labelled images) is very small relative to the class complexity (e.g., $10-30$ per class). When there are only few labelled images for classes with large variability in object shape and appearance (as in the Ballet example of Fig. 3), bad classification is obtained.

When images are represented by "bag-of-features" histograms, "Image-to-image" distance becomes the 'distance' between two descriptor distributions of the two images (which can be measured via histogram intersection, Chi-square, or KL-divergence). "Image-to-Image" KL-distance (divergence) involves measuring the average log-likelihood of each descriptor $d \in I_1$ given the descriptor distribution in I_2 [28]. Consequently, NN-Image classifiers employ the descriptor distribution of each individual image $I \in C$ *separately*. If, instead, we used the descriptor distribution of

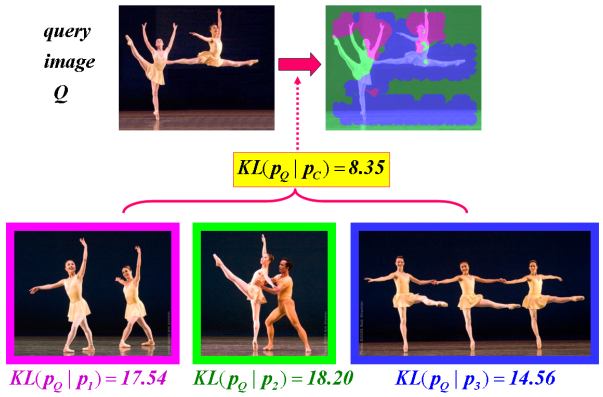


Figure 3. **"Image-to-Image" vs. "Image-to-Class" distance.** *A Ballet class with large variability and small number (three) of 'labelled' images (bottom row). Even though the "Query-to-Image" distance is large to each individual 'labelled' image, the "Query-to-Class" distance is small.* **Top right image:** *For each descriptor at each point in Q we show (in color) the 'labelled' image which gave it the highest descriptor likelihood. It is evident that the new query configuration is more likely given the three images, than each individual image seperately. (Images taken from [4].)*

the *entire* class C (using all images $I \in C$), we would get better generalization capabilities than by employing individual "Image-to-Image" measurements. Such a direct "Image-to-Class" distance can be obtained by computing the KL-distance between the descriptor distributions of Q and C. As can be seen in Fig. 3, even though the "Query-to-Image" KL-distance is large for all the 'labelled' images in the Ballet class, the "Query-to-Class" KL-distance may still be small, enabling correct classification. Inferring new image configurations by "composing pieces" from a set of other images was previously shown useful in [17, 4].

We prove (Sec. 3) that under the Naive-Bayes assumption, the *optimal* distance to use in image classification is the KL "Image-to-Class" distance, and not the commonly used "Image-to-Image" distribution distances (KL, χ^2, etc.)

3. Probabilistic Formulation

In this section we derive the optimal Naive-Bayes image classifier, which is approximated by NBNN (Sec. 4). Given a new query (test) image Q, we want to find its class C. It is well known [7] that maximum-a-posteriori (MAP) classifier *minimizes* the average classification error: $\hat{C} = arg\max_C p(C|Q)$. When the class prior $p(C)$ is uniform, the MAP classifier reduces to the Maximum-Likelihood (ML) classifier:

$$\hat{C} = arg\max_C p(C|Q) = arg\max_C p(Q|C).$$

Let $d_1, ..., d_n$ denote all the descriptors of the query image Q. We assume the simplest (generative) probabilistic model, which is the Naive-Bayes assumption (that the descriptors $d_1, ..., d_n$ of Q are i.i.d. given its class C), namely:

$$p(Q|C) = p(d_1, .., d_n|C) = \prod_{i=1}^{n} p(d_i|C)$$

Taking the log probability of the ML decision rule we get:

$$\hat{C} = arg\max_C \log(p(C|Q)) = arg\max_C \frac{1}{n}\sum_{i=1}^{n} \log p(d_i|C) \quad (1)$$

The simple classifier implied by Eq. (1) is the *optimal classification algorithm* under the Naive-Bayes assumption. In Sec 4 we show how this simple classifier can be accurately approximated using a non-parametric NN-based algorithm (without descriptor quantization).

Naive-Bayes classifier $\Leftrightarrow$ Minimum "Image-to-Class" KL-Distance: In Sec. 2.2 we discussed the generalization benefits of using an "Image-to-Class" distance. We next show that the above MAP classifier of Eq. (1) is equivalent to minimizing "Query-to-Class" KL-distances.

Eq. (1) can be rewritten as:

$$\hat{C} = arg\max_C \sum_d p(d|Q)\log p(d|C)$$

where we sum over all possible descriptors d. We can subtract a constant term independent of C from the right hand side of the above equation, without affecting $\hat{C}$. By subtracting $\sum_d p(d|Q)\log p(d|Q)$, we get:

$$\begin{aligned} \hat{C} &= arg\max_C\Big(\sum_{d\in D} p(d|Q)\log\frac{p(d|C)}{p(d|Q)}\Big) \\ &= arg\min_C\big(KL(p(d|Q)\|p(d|C))\big) \end{aligned} \quad (2)$$

where $KL(\cdot\|\cdot)$ is the KL-distance (divergence) between two probability distributions. In other words, under the Naive-Bayes assumption, *the optimal MAP classifier* minimizes a "Query-to-Class" KL-distance between the descriptor distributions of the query Q and the class C.

A similar relation between Naive-Bayes classification and KL-distance was used in [28] for texture classification, yet between pairs of images (i.e., "Image-to-Image" distances and not "Image-to-Class" distances). Distances between descriptor distributions for the purpose of classification have also been used by others [6, 16, 20, 27, 30], but again – between *pairs of images*.

4. The Approximation Algorithm Using NN

In this section we present the "NBNN" classifier, which accurately approximates the optimal MAP Naive-Bayes image classifier of Sec. 3.

Non-Parametric Descriptor Density Estimation:
The optimal MAP Naive-Bayes image classifier of Eq. (1) requires computing the probability density $p(d|C)$ of descriptor d in a class C. Because the number of local descriptors in an image database is *huge* (on the order of the number of pixels in the database), a Parzen density estimation

provides an accurate non-parametric approximation of the continuous descriptor probability density $p(d|C)$ [7]. Let $d_1^C, .., d_L^C$ denote all the descriptors obtained from all the images contained in class C. Then the Parzen likelihood estimation $\hat{p}(d|C)$ is:

$$\hat{p}(d|C) = \frac{1}{L}\sum_{j=1}^{L} K(d - d_j^C) \quad (3)$$

where $K(.)$ is the Parzen kernel function (which is non-negative and integrates to 1; typically a Gaussian: $K(d - d_j^C) = \exp(-\frac{1}{2\sigma^2} \parallel d - d_j^C \parallel^2)$). As L approaches infinity, and σ (the width of $K(.)$) reduces accordingly, $\hat{p}$ converges to the true density $p(d|C)$ [7].

In principle, to obtain high accuracy, *all the database descriptors* should be used in the density estimation of Eq. (3). While feasible, this is computationally time-consuming (since it requires computing the distance $(d - d_j^C)$ for all descriptor d_j^C $(j = 1..L)$ in each class). We next show an efficient and accurate nearest-neighbor approximation of this Parzen estimator.

The NBNN Algorithm:

Due to the long-tail characteristic of descriptor distributions, almost all of the descriptors are rather isolated in the descriptor space, therefore very far from most descriptors in the database. Consequently, all of the terms in the summation of Eq. (3), except for a few, will be negligible (K exponentially decreases with distance) . Thus we can accurately approximate the summation in Eq. (3) using the (few) r largest elements in the sum. These r largest elements correspond to the r *nearest neighbors* of a descriptor $d \in Q$ within the class descriptors $d_1^C, .., d_L^C \in C$:

$$p_{\text{NN}}(d|C) = \frac{1}{L}\sum_{d=1}^{r} K(d - d_{\text{NN}_j}^C) \quad (4)$$

Note that the approximation of Eq. (4) always *bounds from below* the complete Parzen window estimate of Eq. (3).

Fig. 4 shows the accuracy of such NN approximation of the distribution $p(d|C)$. Even when using a very small number of nearest neighbors (as small as $r = 1$, a *single* nearest neighbor descriptor for each d in each class C), a very accurate approximation $p_{\text{NN}}(d|C)$ of the complete Parzen window estimate is obtained (see Fig. 4.a). Moreover, NN descriptor approximation *hardly reduces the discriminative power of descriptors* (see Fig. 4.b). This is in contrast to the severe drop in discriminativity of descriptors due to descriptor quantization.

We have indeed found very small differences in the actual classification results when changing r from 1 to 1000 nearest neighbors. The case of $r = 1$ is especially convenient to use, since $\log p(d|C)$ obtains a very simple form: $logP(Q|C) \propto -\sum_{i=1}^{n} \parallel d_i - \text{NN}_C(d_i) \parallel^2$ and there is no longer a dependence on the variance of the Gaussian kernel K. This simple form of the classifier was used in all the experimental results reported in Sec. 5.

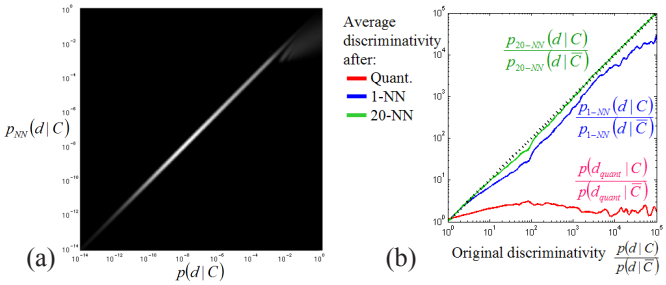


Figure 4. **NN descriptor estimation preserves descriptor density distribution and discriminativity.** **(a)** *A scatter plot of the 1-NN probability density distribution $p_{NN}(d|C)$ vs. the true distribution $p(d|C)$. Brightness corresponds to the concentration of points in the scatter plot. The plot shows that 1-NN distribution provides a very accurate approximation of the true distribution.* **(b)** *20-NN descriptor approximation (Green graph) and 1-NN descriptor approximation (Blue graph) preserve quite well the discriminative power of descriptors. In contrast, descriptor quantization (Red graph) severely reduces discriminative power of descriptors.* **Displays are in logarithmic scale in all axes.**

The resulting Naive-Bayes NN image classifier (NBNN) can therefore be summarized as follows:

The NBNN Algorithm:

1. Compute descriptors $d_1, ..., d_n$ of the query image Q.
2. $\forall d_i \; \forall C$ compute the NN of d_i in C: $\text{NN}_C(d_i)$.
3. $\hat{C} = arg\min_C \sum_{i=1}^{n} \parallel d_i - \text{NN}_C(d_i) \parallel^2$.

Despite its simplicity, this algorithm accurately approximates the theoretically optimal Naive-Bayes classifier, requires no learning/training, and is efficient.

Combining Several Types of Descriptors: Recent approaches to image classification [5, 6, 20, 27] have demonstrated that combining several types of descriptors in a single classifier can significantly boost the classification performance. In our case, when multiple (t) descriptor types are used, we represent each point in each image using t descriptors. Using a Naive Bayes assumption on all the descriptors of all types yields a very simple extension of the NBNN algorithm above. The decision rule linearly combines the contribution of each of the t descriptor types. Namely, Step (3) in the above single-descriptor-type NBNN is replaced by: $\hat{C} = arg\min_C \sum_{j=1}^{t} w_j \cdot \sum_{i=1}^{n} \parallel d_i^j - \text{NN}_C(d_i^j) \parallel^2$, where d_i^j is the i-th query descriptor of type j, and w_j are determined by the variance of the Parzen Gaussian kernel K_j corresponding to descriptor type j. Unlike [5, 6, 20, 27], who learn weights w_j per descriptor-type per class, our w_j are fixed and shared by all classes.

Computational Complexity & Runtime: We use the efficient approximate-r-nearest-neighbors algorithm and KD-tree implementation of [23]. The expected time for a NN-search is logarithmic in the number of elements stored in the KD-tree [1]. Note that the KD-tree data structure is

used only for efficiency of NN-search. It requires no learning/training of parameters. This pre-processing step has a low complexity ($O(N \log N)$ in the number of elements N) and has a low runtime (e.g., a total of a few seconds for constructing all the KD-trees for all the classes in Caltech-101).

Let n_{label} be the number of labelled ('training') images per class, n_C the number of classes and n_D the number of descriptors per image. Each KD-tree contains $n_{label} \cdot n_D$ descriptors. Each of the n_D query descriptors searches within n_C KD-trees. Thus, the time complexity for one query image is $O(n_C \cdot n_D \cdot \log(n_{label} \cdot n_D)) = O(n_C \cdot n_D \cdot \log(n_D))$ (since usually $n_{label} \ll n_D$). There is no training time in our case, except for the fast preprocessing of the KD-tree.

For example, the run time of NBNN on Caltech-101 for classifying an image with densely sampled SIFT descriptors and $n_{label} = 30$, takes 1.6 sec. per class.

5. Results and Experiments

In this section we experimented with NBNN, and compared its performance to other classifiers (learning-based and NN-based). Implementation details are provided in Sec. 5.1. Sec. 5.2 provides performance comparisons on Caltech-101, Caltech-256 and Graz-01. It shows that although our NBNN classifier is extremely simple and requires no learning/training, its performance ranks among the top leading learning-based image classifiers. Sec 5.3 further demonstrates experimentally the damaging effects of using descriptor-quantization or "image-to-image" distances in a non-parametric classifier.

5.1. Implementation

We tested our NBNN algorithm with a *single descriptor-type* (SIFT), and with a combination of 5 *descriptor-types*:
1. The SIFT descriptor ([19]).
2 + 3. Simple Luminance & Color Descriptors: We use log-polar sampling of raw image patches, and take the *luminance part* (L* from a CIELAB color space) as a luminance descriptor, and the *chromatic part* (a*b*) as a color descriptor. Both are normalized to unit length.
4. Shape descriptor: We extended the Shape-Context descriptor [22] to contain *edge-orientation histograms* in its log-polar bins. This descriptor is applied to texture-invariant edge maps [21], and is normalized to unit length.
5. The Self-Similarity descriptor of [25].

The descriptors are densely computed for each image, at five different spatial scales, enabling some scale invariance. To further utilize rough spatial position (similar to [30, 16]), we augment each descriptor d with its location l in the image: $\tilde{d} = (d, \alpha l)$. The resulting L_2 distance between descriptors, $\|\tilde{d}_1 - \tilde{d}_2\|^2 = \|d_1 - d_2\|^2 + \alpha^2 \|l_1 - l_2\|^2$, combines descriptor distance and location distance. (α was manually set in our experiments. The same fixed α was used for Caltech-101 and Caltech-256, and $\alpha = 0$ for Graz-01.)

NN-based method	Performance
SPM NN Image [27]	$42.1 \pm 0.81\%$
GBDist NN Image [27]	$45.2 \pm 0.96\%$
GB Vote NN [3]	52%
SVM-KNN [30]	$59.1 \pm 0.56\%$
NBNN (1 Desc)	**$65.0 \pm 1.14\%$**
NBNN (5 Desc)	**$72.8 \pm 0.39\%$**

Table 1. *Comparing the performance of non-parametric NN-based approaches on the Caltech-101 dataset ($n_{label} = 15$). All the listed methods do not require a learning phase.*

5.2. Experiments

Following common benchmarking procedures, we split each class to randomly chosen disjoint sets of 'training images' and 'test images'. In our NBNN algorithm, since there is *no training*, we use the term 'labelled images' instead of 'training images'. In learning-based methods, the training images are fed to a learning process generating a classifier for the test phase. In our case, there is no such learning phase and the classifier is fixed for all image sets.

We denote by n_{label} the number of 'labelled images' per class. We use commonly used numbers of labelled (training) and test images: On Caltech-101 we randomly selected $n_{label} = 1, 5, 15, 30$ images per class and tested on 20 images per class. On Caltech-256 we randomly selected $n_{label} = 1, 5, 10, 20, 30$ images per class and tested on 25 images per class. The entire procedure was repeated several times (randomly selecting labelled and test sets) and each time performance is computed as the mean recognition rate per class. The benchmark procedure for Graz-01 is somewhat different, and will be described later.

Caltech-101: This database [8] has 101 classes (animals, furniture, vehicles, flowers, etc.) with high intra-class appearance and shape variability. We show three types of comparisons on Caltech-101: (i) Comparing performance of NBNN to other NN-based methods (Table 1). (ii) Comparing NBNN with a single descriptor-type to other single-descriptor-type image classifiers (both learning-based and NN-based) (Fig. 5.a). (iii) Comparing NBNN with multiple descriptor-types to other multi-descriptor-type image classifiers (learning-based methods) (Fig. 5.b).

Table 1 shows a performance comparison on Caltech-101 for several NN-based methods. For this experiment we used 15 labelled images, in order to compare to numbers reported in the other works. Our single descriptor NBNN algorithm (using SIFT) outperforms by a large gap all NN-image methods. Moreover, it outperforms 'SVM-KNN' [30] (a hybrid between NN-based and SVM-based, which was considered state-of-the-art until recently).

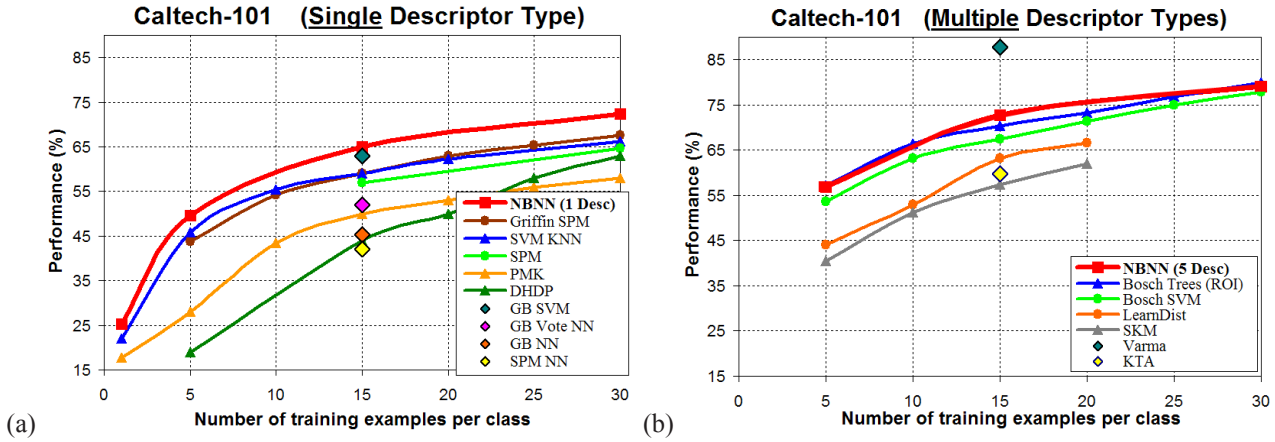


Figure 5. **Performance comparison on Caltech-101. (a) Single descriptor type methods:** *'NBNN (1 Desc)', 'Griffin SPM' [13], 'SVM KNN' [30], 'SPM' [16], 'PMK' [12], 'DHDP' [29], 'GB SVM' (SVM with Geometric Blur) [27], 'GB Vote NN' [3], 'GB NN' (NN-Image with Geometric Blur) [27], 'SPM NN' (NN-Image with Spatial Pyramids Match) [27].* **(b) Multiple descriptor type methods:** *'NBNN (5 Desc)', 'Bosch Trees' (with ROI Optimization) [5], 'Bosch SVM' [6], 'LearnDist' [11], 'SKM' [15], 'Varma' [27], 'KTA' [18].*

Our multi-descriptor NBNN algorithm performs even better (72.8% on 15 labelled images). 'GB Vote NN' [3] uses an image-to-class NN-based voting scheme (without descriptor quantization), but each descriptor votes only to a *single* (nearest) class, hence the inferior performance.

Fig. 5.a further shows that for a single descriptor-type, our NBNN algorithm *outperforms all previously reported learning-based methods*. Note that results obtained by 'GB SVM' (Varma et al. [27] using SVM with a single Geometric Blur (GB) kernel), obtained results similar (slightly worse) than our single-descriptor NBNN (but better than all others). We suggest that the reason for their improved performance relative to other methods is due to the fact that they used *unquantized* (GB) descriptors. Note that NBNN obtained better performance *without training*, and with a *significantly lower runtime*.

Fig. 5.b shows that when combining multiple descriptor-types, NBNN compares in performance to 'Bosch Trees (ROI)' [5], and performs better than all other previously reported learning-based methods, with the exception of 'Varma' [27]. Note that unlike the other methods, we do not learn class-adaptive combinations of descriptor types.

Caltech-256: This database [13] contains 256 categories with higher intra-class variability than Caltech-101, and higher object location variability within the image. Comparison of NBNN to other methods are displayed on Fig. 6. The large positional variability of objects in images of Caltech-256 was effectively addressed by the ROI (Region Of Interest) optimization of [5], leading to better performance of 'Bosch Trees (ROI)' relative to NBNN for a small number of training (labelled) images. However, due to the generalization capabilities of NBNN (resulting from using "image-to-class" distances – see Sec. 2.2), the gap is closed when the number of training (labelled) images reaches 30.

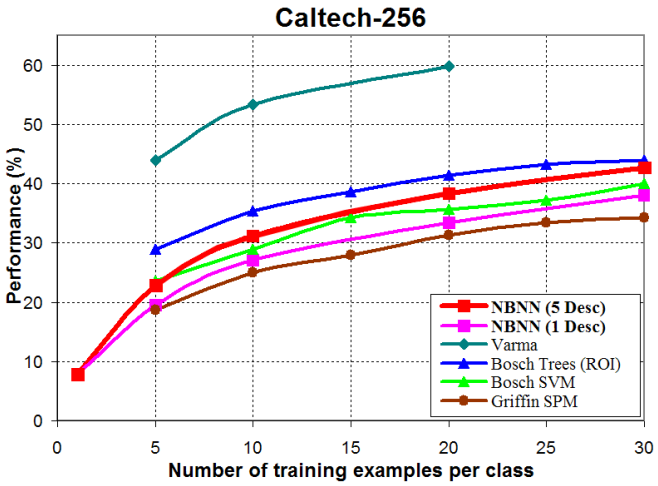


Figure 6. **Performance comparison on the Caltech-256 dataset.** *Single descriptor-type: 'NBNN (1 Desc)', 'Griffin SPM' [13]. Multi-descriptor-type: 'NBNN (5 Desc)', 'Bosch SVM' [6], 'Bosch Trees' (with ROI Optimization) [5], 'Varma' [27]*

Graz-01: The Graz-01 database [24] has two object-classes (bikes and persons), and one background-class. It is characterized by very high intra-class variations of scale, 3D orientations and location of the objects within the image, and there is much more background clutter than in the Caltech databases. The classification task in Graz-01 is Class vs. No-Class. We follow the experimental setup of [16, 24, 31]: For each object (persons/bikes) we randomly sample 100 negative examples (of which 50 images are drawn from the background), and 100 positive examples. The test set is similarly distributed. Table 2 reports the ROC equal error rate averaged over five runs. We compare performance of NBNN against [16, 24, 31]. Although NBNN is a non-parametric (no-learning) method, its per-

Class	Opelt [24]	Zhang [31]	Lazebnik [16]	**NBNN (1 Desc)**	**NBNN (5 Desc)**
Bikes	86.5	92.0	86.3 $\pm$2.5	**89.2** $\pm$4.7	**90.0** $\pm$4.3
People	80.8	88.0	82.3 $\pm$3.1	**86.0** $\pm$5.0	**87.0** $\pm$4.6

Table 2. **Results on Graz-01**

	No Quant.	**With Quant.**
"Image-to-Class"	**70.4%**	50.4% (-28.4%)
"Image-to-Image"	58.4% (-17%)	-

Table 3. *Impact of introducing descriptor quantization or "Image-to-Image" distance into NBNN (using SIFT descriptor on Caltech-101, $n_{label} = 30$).*

formance is better than the learning-based classifiers of [16] (SVM-based) and [24] (Boosting based). NBNN performs only slightly worse than the SVM-based classifier of [31].

5.3. Impact of Quantization & Image-to-Image Dist.

In Sec. 2 we have argued that descriptor quantization and "Image-to-Image" distance degrade the performance of non-parametric image classifiers. Table 3 displays the results of introducing either of them into NBNN (tested on Caltech-101 with $n_{label} = 30$). The baseline performance of NBNN (1-Desc) with a SIFT descriptor is 70.4%. If we replace the "Image-to-Class" KL-distance in NBNN with an "Image-to-Image" KL-distance, the performance drops to 58.4% (i.e., 17% drop in performance). To check the effect of quantization, the SIFT descriptors are quantized to a codebook of 1000 words. This reduces the performance of NBNN to 50.4% (i.e., 28.4% drop in performance).

The spatial pyramid match kernel of [16] measures distances between histograms of *quantized* SIFT descriptors, but within an SVM classifier. Their SVM learning phase compensates for some of the information loss due to quantization, raising classification performance up to 64.6%. However, comparison to the baseline performance of NBNN (70.4%) implies that *the information loss incurred by the descriptor quantization was larger than the gain obtained by using SVM*.

Acknowledgment: The authors would like to thank Lena Gorelick for her many wise and valuable comments. This work was funded in part by the Israel Science Foundation and the Israeli Ministry of Science.

References

[1] S. Arya and H.-Y. A. Fu. Expected-case complexity of approximate nearest neighbor searching. In *Symposium on Discrete Algorithms*, 2000.
[2] E. Bart and S. Ullman. Class-based matching of object parts. In *CVPR Workshop on Image and Video Registration*, 2004.
[3] A. Berg. Shape matching and object recognition. In *Ph.D. Thesis, Computer Science Division, Berkeley, 2005*.
[4] O. Boiman and M. Irani. Similarity by composition. In *NIPS*, 2006.
[5] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *ICCV*, 2007.
[6] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007.
[7] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, New York, 2001.
[8] R. Fei-Fei, L.and Fergus and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative-Model Based Vision*, 2004.
[9] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61, 2005.
[10] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR'03*.
[11] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *ICCV*, 2007.
[12] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005.
[13] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, CalTech, 2007.
[14] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *ICCV*, 2005.
[15] A. Kumar and C. Sminchisescu. Support kernel machines for object recognition. In *ICCV*, 2007.
[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.
[17] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV Workshop on Statistical Learning in CV*, 2004.
[18] Y. Lin, T. Liu, and C. Fuh. Local ensemble kernel learning for object category recognition. In *CVPR*, 2007.
[19] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004.
[20] M. Marszałek, C. Schmid, H. Harzallah, and J. van de Weijer. Learning object representations for visual object class recognition. In *Visual Recognition Challange*, 2007.
[21] C. M. J. Martin, D.R.; Fowlkes. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5), 2004.
[22] G. Mori, S. Belongie, and J. Malik. Efficient shape matching using shape contexts. *PAMI*, 27(11), 2005.
[23] D. Mount and S. Arya. Ann: A library for approximate nearest neighbor searching. In *CGC 2nd Annual Workshop on Comp. Geometry*, 1997.
[24] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *ECCV*, 2004.
[25] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007.
[26] T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. In *ICCV*, 2007.
[27] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, 2007.
[28] M. Varma and A. Zisserman. Unifying statistical texture classification frameworks. *IVC*, 22(14), 2004.
[29] G. Wang, Y. Zhang, and L. Fei-Fei. Using dependent regions for object categorization in a generative framework. In *CVPR*, 2006.
[30] H. Zhang, A. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.
[31] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 73(2), 2007.