Nearest-Neighbor based Metric Functions for Indoor Scene Recognition

Fatih Cakir[1], Uğur Güdükbay[1], Özgür Ulusoy[1]

[1]Bilkent University
Department of Computer Engineering
06800 Bilkent, Ankara, Turkey

e-mail: {fcakir, gudukbay, oulusoy}@cs.bilkent.edu.tr

Corresponding Author: Uğur Güdükbay

Abstract— Indoor scene recognition is a challenging problem in the classical scene recognition domain due to the severe intra-class variations and inter-class similarities of man-made indoor structures. State-of-the-art scene recognition techniques such as capturing holistic representations of a scene demonstrate low performance on indoors. Other methods that introduce intermediate steps such as identifying objects and associating them to scenes have the handicap of successfully localizing and recognizing the objects in a highly cluttered and sophisticated environment.

We propose a classification method that can handle such difficulties of the problem domain by employing a metric function based on the nearest-neighbor classification procedure using the bag-of-visual words scheme, the so-called *codebooks*. Considering the codebook construction as a Voronoi tessellation of the feature space, we have observed that, given an image, a learned weighted distance of the extracted feature vectors to the center of the Voronoi cells gives a strong indication of its category. Our method outperforms state-of-the-art approaches on an indoor scene recognition benchmark and achieves competitive results on a general scene dataset, using a single type of descriptor.

Keywords: scene classification; indoor scene recognition; nearest neighbor classifier; bag-of-visual words.

1 Introduction

Scene classification is an active research area among research scientists. Many classification methods have been proposed in the past that aim to solve different aspects of the problem such as topological localization, indoor-outdoor classification and scene categorization [1], [2], [3], [4], [5], [6], [7], [8], [9]. In scene categorization the problem is to associate a semantic label to a scene image. Although the categorization methods address the problem of categorizing any type of a scene, they usually perform well on outdoors only [10]. In contrast, classification of indoor images has remained a further challenging task due to the more difficult nature of the problem. The intra-class variations and inter-class similarities of indoor scenes are the biggest barriers for many recognition algorithms to achieve satisfactory performance on never seen images, i.e., test data. On the other hand, recognizing indoors is very important for many fields. For example, in the field of robotics, the perceptual capability of a robot for identifying its surroundings is a highly crucial ability.

Earlier works on scene recognition are based on extracting low-level features such as color, texture and shape properties of the image [1], [3], [5]. Such simple global descriptors are not powerful enough to perform well on large datasets with sophisticated environmental settings. Olivia and Torralba [4] introduce a more compact and robust global descriptor, the so-called *gist*, which captures the holistic representation of an image using spectral analysis. Their descriptor performs well on categorizing outdoor images such as forests, mountains and suburban environments but has difficulties in recognizing indoors.

Borrowing ideas from the human perceptual system, recent work on indoor scene recognition focuses on classifying images by using representations of both global and local image properties and integrating intermediate steps such as object detection [10], [11]. This is not surprising since indoors are usually characterized by the objects they contain. Consequently, indoor scene recognition can be mainly considered as a problem of identifying the objects first, and then, classifying the scene accordingly. Intuitively, this idea seems reasonable but with state-of-the-art object recognition methods [12], [13], [14], it is very unlikely to successfully localize and identify unknown number of objects in such a cluttered and sophisticated indoor image. Hence, classifying a particular scene via objects becomes yet a more challenging issue to handle.

A solution to this problem is to classify the indoor image by implicitly modeling objects with densely sampled local cues. These cues will then give indirect evidence of a presence of an object. Although this solution seems contrary to the methodology of recognizing indoors by the human visual system, i.e., explicitly identifying objects and associating them to scenes, it provides a successful alternative way by bypassing the drawbacks of trying to successfully localize objects in highly intricate environments.

The most successful and popular descriptor that captures the crucial information of an image region is the Scale-Invariant Feature Transform (SIFT) [15], [16]. [16]. This brings the idea that SIFT-like features extracted densely from images of a certain class have to be more similar in some manner than image descriptors of irrelevant classes. This similarity measure can be achieved by first defining a set of categorical words (the so-called *visual words*) for each class and then using a learned metric function to measure the distance between local cues and these visual words.

Thus, we introduce a novel non-parametric weighted metric function with a spatial extension based on the approach described in [17]. In their work, Bolman *et al.* show that a Nearest-Neighbor (NN) based classifier which computes direct Image-to-Class distances without any quantization step achieves performance rates among the top leading learning-based classifiers. We show that a NN-based classifier is also well suited for categorizing indoors since: i) It incorporates image-to-class distances which is extremely crucial for classes with high variability; ii) Considering the insufficient performance of state-of-the-art recognition algorithms on a large object dataset [12], it successfully allows classifying indoor sceneries directly from local cues without having to incorporate any intermediate steps such as categorizing via objects; iii) Given a query image, it allows ranked results and thus can be employed for a preprocessing step to successfully narrow down the size of possible categories for subsequent analyses.

Bolman *et al.* also show that a descriptor quantization step, i.e., codebook generation, severely degrades the performance of the classifier by causing information loss in the feature space. They argue that a non-parametric method such as the Nearest-Neighbor classifier has no training phase like the learning-based methods to compensate this loss of information. They evaluate their approach on Caltech101 [18] and Caltech256 datasets [19], where each image contains only one object and maintains a common position, and on the Graz-01 dataset [20], which has three classes (bikes, persons and a background class) with a basic class vs. no-class classification task. On the other hand, for a multi-category recognition task of scenes where multiple objects co-exist in a highly cluttered, varied and complicated form, we observe that our NN-based classifier with a descriptor quantization step outperforms the state-of-the-art learning-based methods. The additional quantization step allows us to incorporate spatial information of the quantized vectors, and more importantly, it significantly reduces the performance gap between our method and other learning-based approaches. It is computationally inefficient for a straightforward NN-based method without a quantization step to perform classification, considering the datasets with large amount of training images.

The rest of this paper is organized as follows: Section 2 discusses related work. In Section 3 we describe the framework of our proposed method. We present experimental results and evaluate the performance in Section 4. Section 5 gives conclusions and future work.

2 Related Work

Earlier works on scene classification are based on extracting low-level features such as color, texture and shape properties of the image. Szummer and Picard [1] use such features to determine whether an image is an outdoor or an indoor scene. Vailaya *et al.* [3] use color and edge properties for the city vs. landscape classification problem. Ulrich and Nourbakhsh [5] employ color-based histograms for mobile robot localization. Such simple global features are not discriminative enough to perform well on a difficult classification problem, such as recognizing scene images. To overcome this limitation, Olivia and Torralba [4] introduce the gist descriptor, a technique that attempts to categorize scenes by capturing the spatial structure properties, such as the degree of openness, roughness, naturalness, using spectral analysis of the image. Although a significant improvement over earlier basic descriptors, it has been shown in [10] that it performs poorly in recognizing indoor images. One other popular descriptor is SIFT [16]. Due to its strong discriminative power even under severe image transformations, noise and illumination changes, it has been the most preferred visual descriptor in many scene recognition algorithms [6], [7], [21], [22], [23].

Such local descriptors have been successfully used with the bag-of-visual words scheme for constructing codebooks. This concept has proven to provide good results in scene categorization [23]. Fei-Fei and Perona [22] represent each category with such a codebook and classify scene images by using Bayesian hierarchical models. Lazebnik *et al.* [7] use the same concept with spatial extensions. They hierarchically divide an image into sub-regions, which they call the spatial pyramid, and compute histograms based on quantized SIFT vectors over these regions. A histogram intersection kernel is then used to compute a matching score for each quantized vector. The final spatial pyramid kernel is implemented as concatenating weighted histograms of all features at all sub-regions. The traditional bag-of-visual words scheme discards any spatial information; hence many methods utilizing this concept also introduce different spatial extensions [7], [24].

Bosch *et al.* [25] present a review of the most common scene recognition methods. However, recognizing solely indoors is a more challenging task than recognizing outdoor scenes owing to severe intra-class variations and inter-class similarities of man-made indoor structures. Consequently, it has been investigated separately within the general scene classification problem. Quattoni and Torralba [10] brought attention to this challenging task by introducing a large indoor scene dataset consisting of 67 categories. They argue that together with the global structure of a scene which they capture via the gist descriptor, the presences of certain objects described by local features are strong indications of its category. Espinace *et al.* [11] suggest using objects as an intermediate step for classifying a scene. Such approaches are coherent with the human vision system since we identify and characterize scenes with the objects they contain. However, with the state-of-the-art object recognition methods [12], [13], [14], [26], it is very unlikely to successfully identify multiple objects in such a cluttered and sophisticated environmental setting. Instead of explicitly modeling the objects, we can use local cues as indirect evidence about the presence of them and thus bypass the drawbacks of having to successfully recognize it, which is a very difficult problem considering the intricate nature of indoors.

3 Nearest-Neighbor based Metric Functions (NNbMF)

3.1 Baseline Problem Formulation

The popular bag-of-visual words paradigm introduced in [27] has become commonplace in various image analysis tasks. It has proven to provide powerful image representations for image classification and object/scene detection. To summarize the procedure, consider **X** to be a set of feature descriptors in D-dimensional space, i.e., $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M]^T \in \mathbb{R}^{M \times D}$. A vector quantization or a codebook formation step involves the Voronoi tessellation of the feature space by applying k-means clustering to set **X** to minimize the cost function

$$J = \sum_{i=1}^{K} \sum_{m=1}^{M} \|\mathbf{x}_m - \mathbf{v}_\mathrm{i}\|^2 \tag{1}$$

where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2 \dots, \mathbf{v}_K]^T$ correspond to centers of the Voronoi cells, i.e. the visual words of codebook **V**, and $\|\cdot\|$ denotes the L_2-norm. After forming a codebook for each class using Equation 1, a set $\mathbf{X}_q = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T$ denoting the extracted feature descriptors from a query image can be categorized to class c by employing the Nearest-Neighbor classification function $y : \mathbb{R}^{N \times D} \rightarrow \{1, \dots, C\}$ given as

$$y(\mathbf{X}) = \underset{c=1,\dots C}{\operatorname{argmin}} \left[\underbrace{\sum_{n=1}^{N} \|\mathbf{x}_n - NN_c(\mathbf{x}_n)\|}_{h(\mathbf{X})} \right] \tag{2}$$

where $NN_c(\mathbf{x})$ denotes the nearest visual word of $\mathbf{x}_n$, i.e., nearest Voronoi cell center, in the Voronoi diagram of class c and $y_i \in \{1, \dots, C\}$ refers to class labels. It should be noted that Equation (2) does not take into account unquantized descriptors, as in [17]. There is a trade-off between information loss and computational efficiency because of the quantization of the feature space.

3.2 Incorporating Spatial Information

The classical bag-of-visual words approach does not take into account spatial information and thus loses crucial information about the spatial distribution of the feature descriptors within an image. Hence, this is an important aspect to be considered for achieving satisfactory results in a classification framework. We incorporate spatial information as follows. Given extracted descriptors in D-dimensional space, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times D}$ and their spatial locations $\mathbf{S} = [(x_1, \mathrm{y}_1), (x_2, y_2), \dots, (x_N, y_N)]$, during the codebook generation step, we also calculate their relative position with respect to the corresponding image boundaries in which they are extracted. Hence their relative locations are $\mathbf{S}' = [(x'_1, y'_1), (x'_2, y'_2), \dots, (x'_N, y'_N)] = \left[\left(\frac{x_1}{w_1}, \frac{y_1}{h_1}\right), \left(\frac{x_2}{w_2}, \frac{y_2}{h_2}\right), \dots, \left(\frac{x_N}{w_N}, \frac{y_N}{h_N}\right)\right]$ where (w_1, h_1), (w_2, h_2), … , (w_N, h_N) pairs represent the width and height values of the corresponding images. After applying clustering to the set $\mathbf{X}$, we obtain the visual word set $\mathbf{V}$ as described in the previous section. Since similar feature descriptors of $\mathbf{X}$ are expected to be assigned to the same visual word, their corresponding coordinate values described in set $\mathbf{S}'$ should have similar values. Figure 1 shows the spatial layout of the descriptors assigned to several visual words.

To incorporate this information into Equation (2), we consider the density estimation methods that are generally used for determining unknown probabilistic density functions. It should be noted that we do not consider a probabilistic model; thus obtaining and using a legitimate density function is irrelevant in our case. We can assign weights for each grid on the spatial layout of every visual word using a histogram counting technique (cf. Figure 1). Suppose we geometrically partition this spatial layout into $M \times M$ grids. Then for the f^{th} visual word of class c, $\mathbf{v}_{cf}$, the weight of a grid can be calculated as

$$\mathbf{W}^{cf} = \left[w_{ij}^{cf}\right] = \frac{k}{N} \tag{3}$$

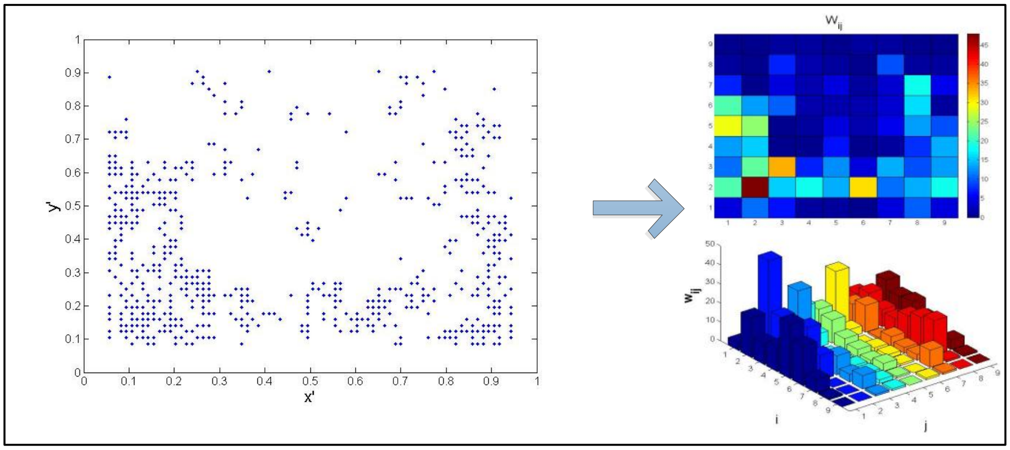
where k is the number of descriptors assigned to $\mathbf{v}_{cf}$ that fall into that particular grid and N is the total amount of descriptors assigned to $\mathbf{v}_{cf}$. During the classification of a query image, the indices i, j correspond to the respective grid location of an extracted feature descriptor. One alternative way for defining weights is to firstly considering $\mathbf{W}^{cf} = \left[w_{ij}^{cf}\right] = k$ and then scaling this matrix as

$$\mathbf{W}^{cf'} = \frac{\left[w_{ij}^{cf}\right]}{\max(\mathbf{W}^{cf})} \tag{4}$$

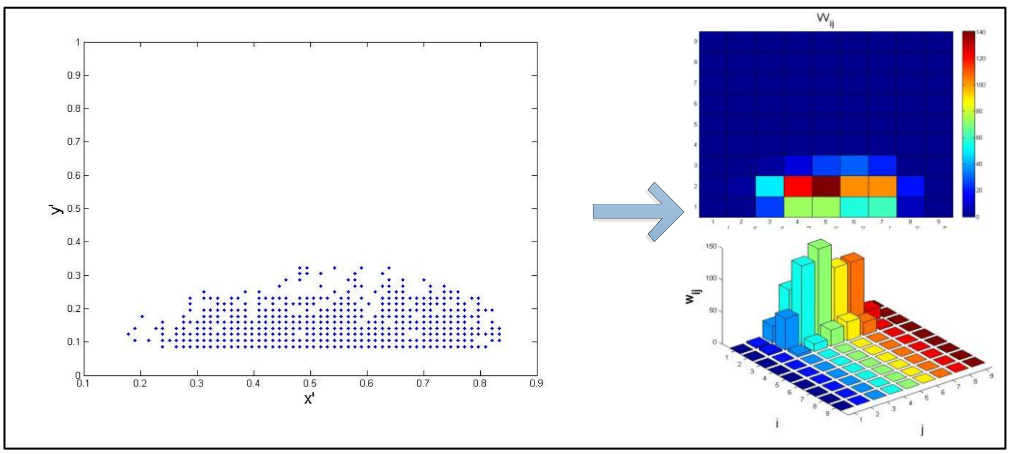
where $\max(\cdot)$ describes the largest element. Equation (4) does not provide weight consistency of the visual words throughout a codebook. It assigns larger weights to visual words that have a sparse distribution in the spatial layout while attenuates the weights of the visual words that are more spatially compact. The choice of a weight matrix assignment is directly related to the problem domain, as we have found Equation (3) more suitable for the 67-indoor benchmark and Equation (4) suitable for the 15-scenes benchmark.

We calculate the weight matrices for all visual words of every codebook. The function $h(\mathbf{X})$ described in Equation (2) now can be improved as

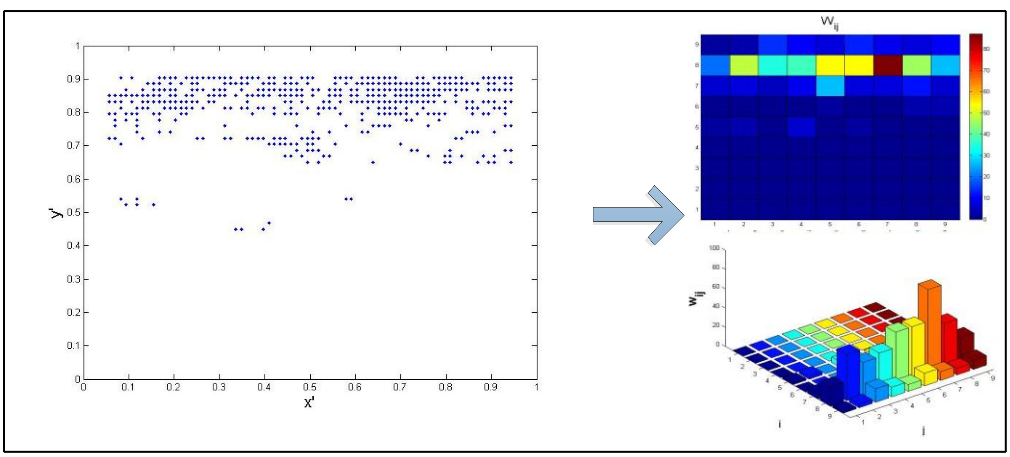
$$\sum_{m=1}^{M} \left(1 - \gamma_c \mathbf{W}_{ij}^{cf}\right) \times \|\mathbf{x}_m - NN_c(\mathbf{x}_m)\| \tag{5}$$



(a)



(b)



(c)

Figure 1: Left of (a), (b) and (c) represent the spatial layout of three different visual words that represent the relative positions of the extracted descriptors to its image boundaries. These layouts are then geometrically partitioned into $M \times M$ bins and a weight matrix $\mathbf{W}$ is computed as shown in right hand side of (a), (b) and (c).

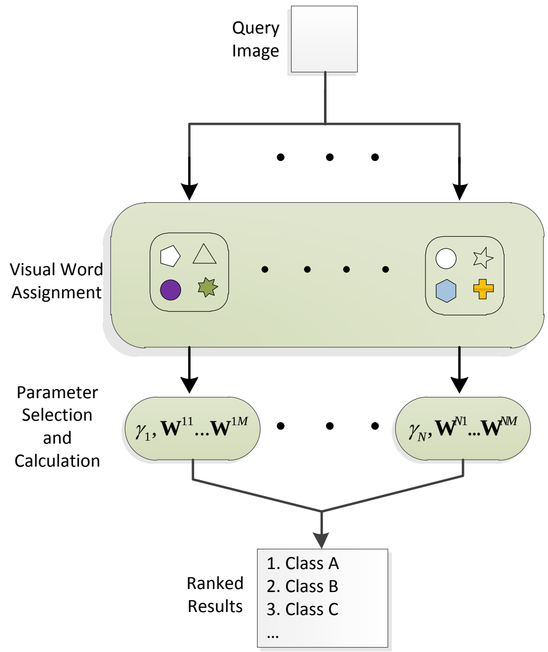


Figure 2: Flow chart of the testing phase of our method.

where $NN_c(\mathbf{x}_m) \equiv \mathbf{v}_{cf}$. Obviously γ_c operates as a scale operator for a particular class, e.g., if $\gamma_c = 0$ then the spatial location for class c is entirely omitted when classifying an image, i.e., only the sum of the descriptors Euclidean distance to their closest visual words is considered.

This scale operator can be determined manually or by using an optimization model. Now assume a vector $\mathbf{d}^c \in \mathbb{R}^M$ that holds the L_2 −norm of every extracted descriptor $\mathbf{x}$ of an image to the nearest visual word of codebook c as its elements; i.e., $d_f^c = \left\|\mathbf{x}_f - NN_c(\mathbf{x}_f)\right\|$, where $f \in [1, M]$ corresponds to the extracted descriptor indices and $NN_c(\mathbf{x}_i)$ refers to the nearest visual word to $\mathbf{x}_i$ ($NN_c(\mathbf{x}_i) \equiv \mathbf{v}_{cf}$). α_i^c denotes the corresponding spatial weights assigned to d_i^c; i.e., $\alpha_i^c = \gamma_c \mathbf{W}_{ij}^{cf}$. Referring to the vector of these spatial weights as $\boldsymbol{\alpha}^c \in \mathbb{R}^M$, Equation (5) can now be redefined as $(\mathbf{1} - \boldsymbol{\alpha}^c) \cdot \mathbf{d}^c$ and an image can be classified to class c by using the function

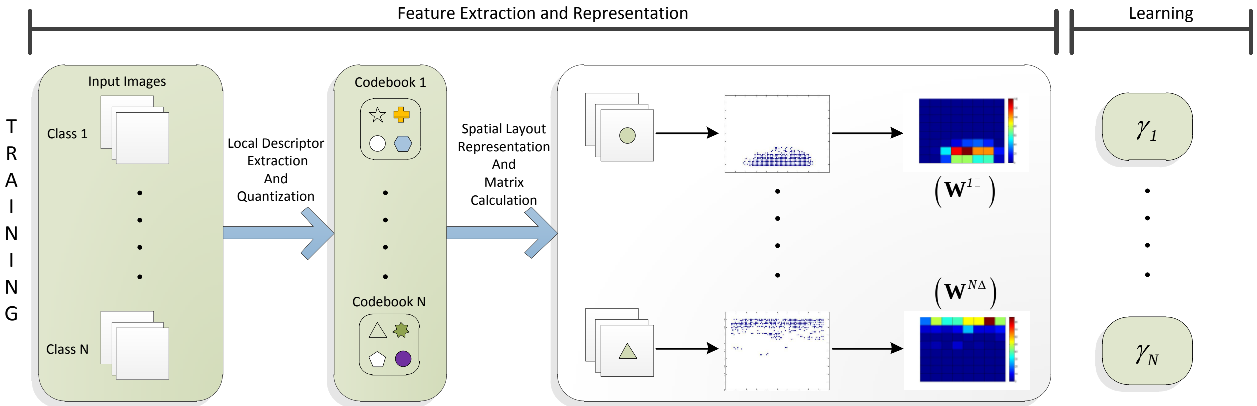


Figure 3: Flow chart of the training phase of our method.

$$y(\mathbf{X}) = \underset{c=1,\ldots C}{\operatorname{argmin}} \left[\underbrace{(\mathbf{1} - \boldsymbol{\alpha}^c) \cdot \mathbf{d}^c}_{h(\mathbf{X})} \right] \tag{6}$$

Consider an image i that belongs to class j with an irrelevant class k. We would like to satisfy the inequalities $\left(\mathbf{1} - \boldsymbol{\alpha}_i^j\right)^T \mathbf{d}_i^j < \left(\mathbf{1} - \boldsymbol{\alpha}_i^k\right)^T \mathbf{d}_i^k$. Given i training images and j classes, we specify a set of $S = i \times j \times (j-1)$ inequality constraints where $k = j - 1$. Since we will not be able to find a scale vector that satisfies all such constraints, we introduce slack variables, ξ_{ijk}, and try to minimize the sum of slacks allowed. We also aim to select a scale vector $\boldsymbol{\gamma}$ such that Equation (5) remains as close to Equation (2) as possible. Hence we minimize the L_n −norm of $\boldsymbol{\gamma}$. Consequently, finding the scale vector $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_j]$ can now modeled as an optimization problem as follows:

$$\begin{aligned} &\min && \|\boldsymbol{\gamma}\|_n + \varphi \sum_{i,j,k} \xi_{ijk} \\ &\text{subject to} && \forall (i,j,k) \in S: \\ & && \left(-\boldsymbol{\alpha}_i^j\right)^T \mathbf{d}_i^j + \left(\boldsymbol{\alpha}_i^k\right)^T \mathbf{d}_i^k < \mathbf{d}_i^k - \mathbf{d}_i^j + \xi_{ijk} \\ & && \xi_{ijk} \geq 0\,, \boldsymbol{\gamma} \geq \mathbf{0} \end{aligned} \tag{7}$$

where φ is a penalizing factor. We choose n from $\{1, 2\}$, resulting in a linear and quadratic programming problem, respectively. One may prefer the L_2 −norm, since sparsity is not desirable in our case due to the fact that sparse solutions may heavily bias categories associated with large scale weights. An alternative model is to define one weight value associated to all categories. This model is less flexible but it prevents a possible degradation in recognition performance caused by sparsity. The scale vector can also be manually chosen. Figures 2 and 3 depict the testing and training phase of the proposed method, respectively.

4 Experimental Setup and Results

4.1 Training Data and Parameter Selections

This section presents the training setup of our NN-based metric function on the 15 scenes [7] and 67 indoors datasets [10]. The 15-scenes dataset contains 4485 images spread over 15 indoor-outdoor categories containing 200 to 400 images each. We use the same experimental setup as in [7] and select randomly chosen 100 images for training, i.e., for codebook generation and learning the scale vector $\boldsymbol{\gamma}$.

The 67-indoors dataset contains images solely from indoors with very high intra-class variations and inter-class similarities. We use the same experimental setup, as in [10].

We use two different scales of SIFT descriptors for evaluation. For the 15-scenes dataset, patches with bin sizes of 6 and 12 pixels are used, and for the 67-indoors dataset, the bin sizes are selected as 8 and 16 pixels. The SIFT descriptors are sampled and concatenated at every four pixels and are constructed from 4×4 grids with eight orientation bins (256 dimension in total). The training images are first resized to speed the computation and to provide scale consistency. The aspect ratio is maintained, but all images are scaled down so their largest resolution does not exceed 500 and 300 pixels and the feature space is clustered using k-means into 500 and 800 visual words, for the 67-indoors and 15-scenes datasets, respectively.

The spatial layout of each visual word from each category is geometrically partitioned into $M \times M$ bins and a weight matrix is formed for each visual word from Equation (3) and Equation (4). Several settings are used to determine the scale vector $\boldsymbol{\gamma}$. We first consider assigning different weights to all categories ($\boldsymbol{\gamma} \in \mathbb{R}^C$). We find the optimal scale vector by setting $n = \{1, 2\}$ in Equation (7) and solving the corresponding optimization problem. We also use another setting for the optimization model where we assign the same weight to all categories ($\gamma \in \mathbb{R}$). Alternatively, we select the scale parameter $\gamma \in \mathbb{R}$ manually.

The constraints in Equation (7) are all formed as described in the previous section with 10 training images. The rest of the training set are used for codebook construction. However, we also show results when all the training images are used for codebook construction and the scale parameter γ is chosen manually.

Performance rate is calculated by the ratio of correctly classified test images within each class. The final recognition rate is the total correctly classified test images divided by the total number of test images used in the evaluation.

4.2 Results and Discussion

Table 1 shows recognition rates for both datasets with different scale vector settings. $Baseline$ and $Baseline_{full}$ refers to the method when Equation (2) is used in which no spatial information is incorporated. The difference is that $Baseline_{full}$ uses all the available training images for codebook generation while $Baseline$ leaves 10 images for scale parameter learning. Not surprisingly, in both datasets $Baseline_{full}$ shows better performance. The results where a

TABLE I

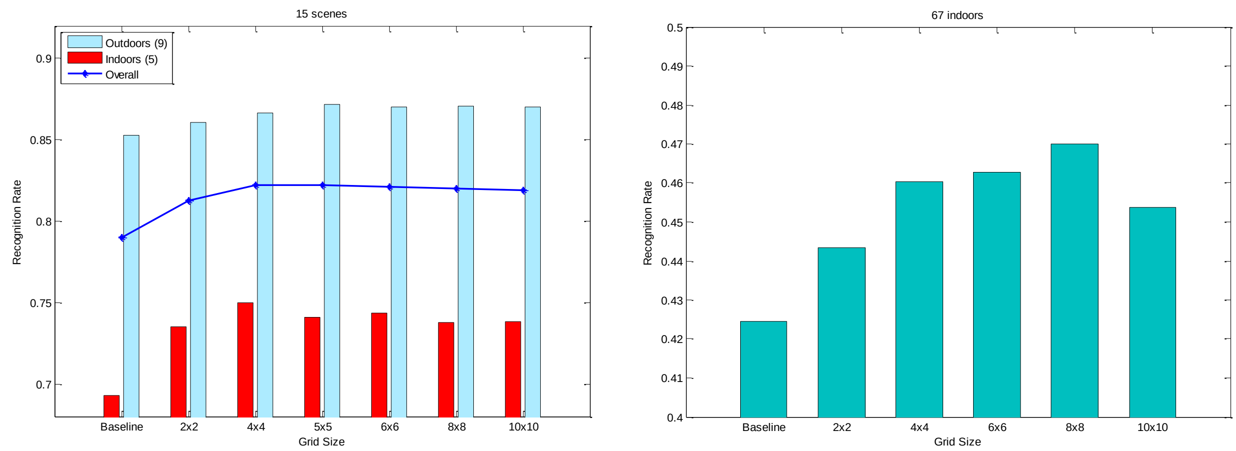
PERFORMANCE COMPARISON WITH DIFFERENT $\boldsymbol{\gamma}$ SETTINGS

	$Baseline$	$\boldsymbol{\gamma}_{LP} \in \mathbb{R}^C$	$\boldsymbol{\gamma}_{QP} \in \mathbb{R}^C$	$\gamma_{LP} \in \mathbb{R}$	$Baseline_{full}$	$\gamma \in \mathbb{R}$
67-indoors	40.75	35.15	35.15	43.13	42.46	**47.01**
15-scenes	78.93	79.60	79.83	81.17	78.99	**82.24**

C refers to the number of categories in a dataset and Baseline refers to the method when Equation (2) is used. Subscripts LP and QP stand for linear and quadratic programming, respectively. They refer to the optimization model with different n settings in Equation (7).

scale parameter is assigned to every category ($\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \dots, \gamma_C] \in \mathbb{R}^C$) are slightly better than the baseline implementation in the 15-scenes benchmark. Although an insignificant increase, we observe that setting $n = 2$ in Equation (7) gives a higher recognition rate compared to $n = 1$. This confirms our previous assertion that dense solutions increase the performance. This effect is clearly observed when we assign the same scaling parameter γ to all 15 categories.

On the other hand, assigning a different scale parameter for each category in the 67-indoors benchmark decreases the performance values for both the LP and QP programming models. In fact we observed that the solutions to these models are identical for our setting. This situation can avoided and the overall performance value can be increased by using more training images, however this results in the reduction of the amount of available training images for codebook construction and this also degrades performance. One other solution is to assign the same scale parameter to all categories. This positively affects the performance resulting in a 43% recognition rate. One can easily expect that this effect will be much stronger in a problem domain where spatial distributions of the visual words are more ordered and compact. More importantly when we use all training images for generating codebooks and thus manually select the scale parameter we observe the highest results.



(a) (b)

Figure 4: Recognition rates based on different grid size settings.

Figure 4 shows the recognition rates with different weight matrix ($\mathbf{W}$) sizes. Geometrically partitioning the spatial layout into 5×5 and 8×8 grids yields the best results for the 15-scenes and 67-indoors datasets, respectively. The 15-scenes dataset can be separated as 5 indoor and 9 outdoor categories. We ignore the *industrial* category since it contains both indoor and outdoor images. Observe that incorporating the spatial information improves the performance rate of the outdoor categories by $\sim$2% only. The performance rate for indoor categories is improved by up to 6%. This difference can be explained by the more orderly form of the descriptors extracted from indoor images. This improvement is $\sim$4,5% for the 67-indoors dataset due to further difficulty and intra-class variations.

Table 2 compares our method with state-of-the-art scene recognition algorithms. Our method achieves more than 7% improvement over the best published result in the 67-indoor benchmark [26] and shows competitive performance in the 15-scenes dataset. Figure 5 and 6 shows the confusion matrix of the 67 indoors and 15 scenes datasets, respectively.

Our method also induces rankings that could naturally be used as a pre-processing step of another recognition algorithm. As shown in Figures 7 (a) and (b), our method returns the correct

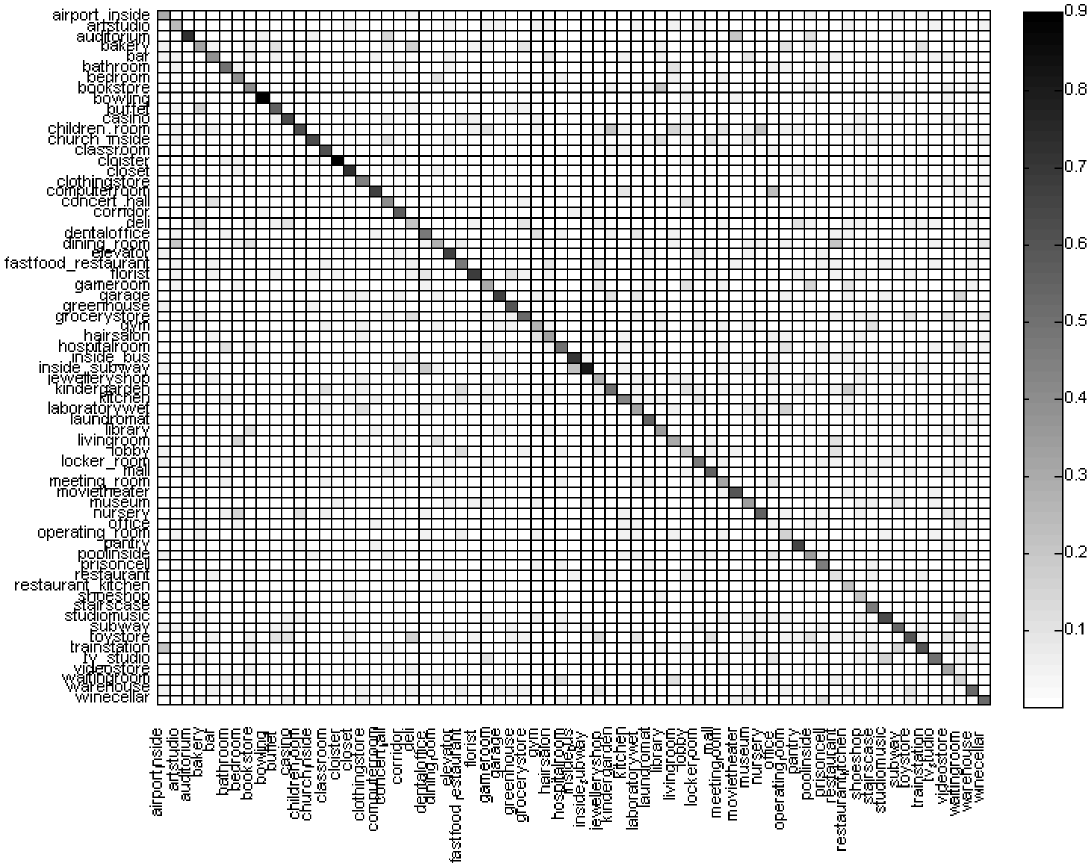


Figure 5: Confusion matrix of the 67 indoors dataset. The horizontal and vertical axes correspond to the true and predicted classes, respectively.

category within the top ten results by ranking the categories for a query image with 82% overall accuracy in the 67-indoors benchmark. This rate is near 100% considering the returned top three results in the 15-scenes dataset (cf. Figure 7 (b)). Hence one can utilize this aspect of our algorithm to narrow down category choices, consequently increasing their final recognition rate by analyzing other information channels of the query image with different complementary descriptors or classification methods. Figure 8 shows a set of classified images.

4.3 Runtime Performance

Compared to the learning-based methods such as the popular Support Vector Machines (SVM), the Nearest-Neighbor classifier has a slow classification time, especially when the data points to

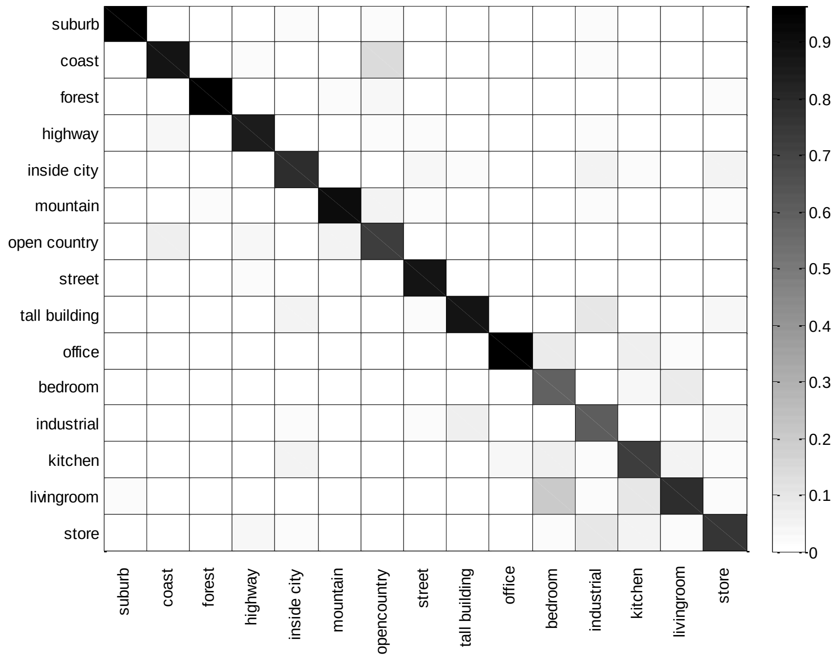
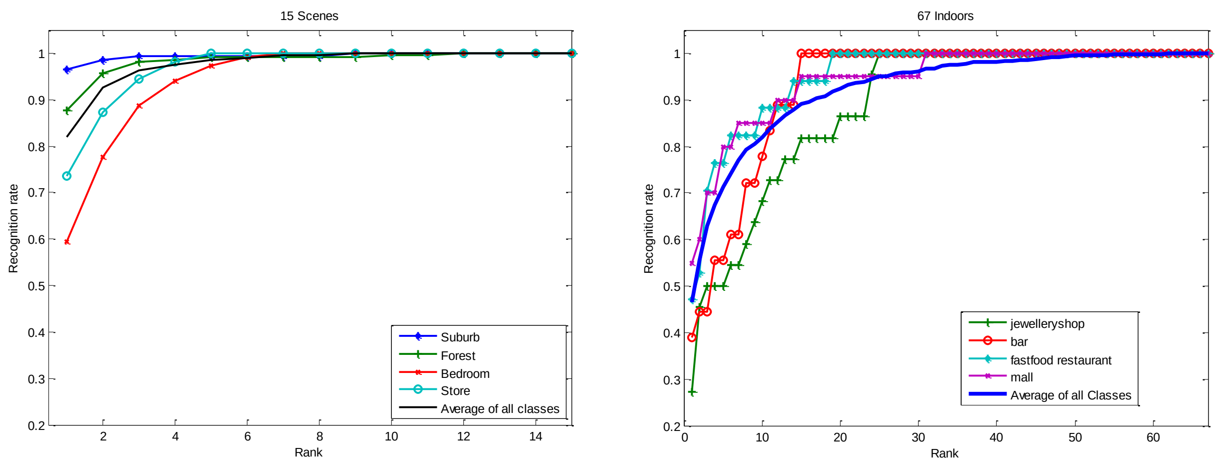


Figure 6: Confusion matrix of the 15-scenes dataset. The columns and row denote the true and predicted classes, respectively.

be considered is too large and the dimension is too high. Several approximation techniques have been proposed to increase the efficiency of this method [28], [29]. These techniques involve pre-processing the search space using data structures, such as KD-trees or BD-trees. These trees are hierarchically structured so that only a subset of the data points in the search space is considered for a query point. We utilize the Approximate Nearest Neighbors library (ANN) [28]. For the 67 indoors benchmark, it takes approximately 0.9 seconds to form a tree structure of a category codebook and about 2.0 seconds to search all query points of an image in a tree structure, using an Intel Centrino Duo 2.2 GHz CPU. Without quantizing, it takes about 100 seconds to search all the query points. For the 15-scenes benchmark, it takes about 1.5 seconds to construct a search tree and 4.0 seconds to search all query points in it. Without quantizing, it takes approximately 200 seconds to search all the query points.

TABLE II
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART METHODS

Methods	67 Indoors Classification Rate	15 Scenes Classification Rate
Morioka *et al.* [26]	39.63 ± 0.69	83.40 ± 0.58
Quattoni and Torralba [10]	~ 28	-
Zhou *et al.* [31]	-	$\mathbf{85.20}$
Yang *et al.* [13]	-	80.28 ± 0.93
Lazebnik *et al.* [7]	-	81.40 ± 0.50
NNbMF	$\mathbf{47.01}$	82.24



(a) (b)

Figure 7: Recognition rates based on rankings. Given a query image, if the true category is returned in the top k results, it is considered as a correct classification.

The CUDA implementation of the K-nearest neighbor method [30] further increases the efficiency by parallelizing the search process. we observed ~ 0.2 seconds per class to search the query points extracted from an image using a NVIDIA Geforce 310M graphics card.

5 Conclusion

We propose a simple, yet effective Nearest-Neighbor based metric function for recognizing indoor scene images. In addition, given an image our method also induces rankings of categories for a possible pre-processing step for further classification analyses. Our method also incorporates the spatial layout of the visual words formed by clustering the feature space. Experimental results show that the proposed method effectively classifies indoor scene images compared to state-of-the-art methods.

We are currently investigating to further improve the spatial extension part of our method by using other estimation techniques to better capture and model the layout of the formed visual words. We are also investigating to apply the proposed method to other problem domains such as auto-annotation of images.

References

[1] M. Szummer and R. W. Picard, "Indoor-outdoor image classification", *in Proceedings of the International Workshop on Content-Based Access of Image and Video Databases (CAIVD '98)*, p. 42, Washington, DC, USA, 1998.

[2] A. Torralba, K. Murphy, W. Freeman, and M. Rubin, "Context-based vision system for place and object recognition", *in Proceedings of the International Conference on Computer Vision*, 2003.

[3] A. Vailaya, A. Jain, and H. Zhang, "On image classification: city vs. landscapes", *Pattern Recognition*, vol. 31, pp. 1921-1935, 1998.

[4] 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, pp. 145-175, 2001.

[5] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization", *in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[6] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification using a hybrid generative/discriminative approach", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 712-727, 2008.

[7] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories", *in Proceedings of the_IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

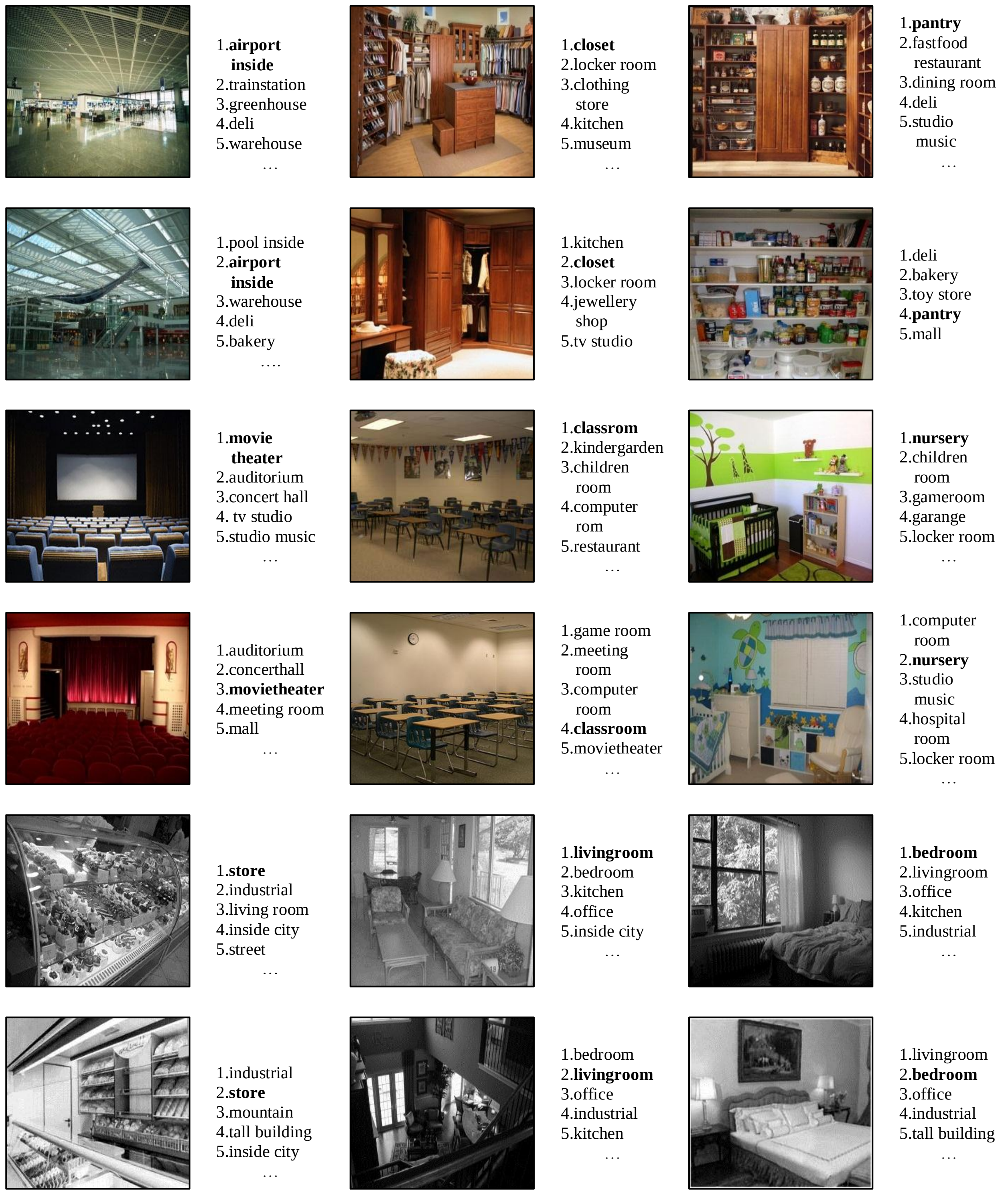


Figure 8: Classified images for a subset of indoor scene images. Images from the first five row are taken from the 67-indoors and the last two rows are from the indoor categories of the 15-scenes dataset. For every query image the list of ranked categories is shown on the right side. The bold name denotes the true category.

[8] S. Se, D. G. Lowe, and J. J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," *in Proceedings of the International Conference on Robotics and Automation*, pp. 2051–2058, 2001.

[9] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen, "A discriminative approach to robust visual place recognition", *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[10] A. Quattoni and A.Torralba, "Recognizing indoor scenes", *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[11] P. Espinace, T. Kollar, A. Soto, and N. Roy. "Indoor Scene Recognition Through Object Detection", Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2010.

[12] P. Gehler, S. Nowozin, "On feature combination for multiclass object classification", *in Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 221-228, 2009.

[13] J. Yang, K. Yu, Y. Gong, T. Huang, "Linear spatial pyramid matching using sparse coding for image classification". *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1794-1801

[14] J. Wu, J. M. Rehg, "Beyond the Euclidean distance: creating effective visual codebooks using the histogram intersection kernel", *in Proceedings of the Twelveth IEEE International Conference on Computer Vision (ICCV)*, 2009.

[15] K. Mikolajczyk, C. Schmid, "A performance evaluation of local descriptors", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615-1630, 2005

[16] D. Lowe, "Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[17] O. Boiman, E. Shechtman, M. Irani, "In defense of Nearest-Neighbor based image classification", *in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* , pp.1-8, 2008.

[18] L. Fei-Fei, R. Fergus, P. Perona, P., "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories", *in Proceedings of the CVPR Workshop on Generative-Model Based Vision*, 2004.

[19] G. Grin, A. Holub, P. Perona, "Caltech 256 Object Category Dataset", *Technical Report*, UCB/CSD-04-1366, California Institute of Technology, 2006.

[20] A. Opelt, M. Fussenegger, A. Pinz, P. Auer, “Weak hypotheses and boosting for generic object detection and recognition”, *in Proceedings of the Eighth European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science*, Vol. 2, pp. 71-84, 2004.

[21] J. Vogel, B. Schiele, “A semantic typicality measure for natural scene categorization”, *in Proceedings of 26th Pattern Recognition Symposium (DAGM), Lecture Notes in Computer Science,* vol. 3175, pp. 195–203, 2004.

[22] L. Fei-Fei and P. Perona, “A bayesian hierarchical model for learning natural scene categories,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* vol. II, pp. 524–531, 2005.

[23] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, and T. Tuytelaars, “A thousand words in a scene,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, no. 9, pp. 1575–1589, 2007.

[24] V. Viitaniemi, J. Laaksonen, “Spatial extensions to bag of visual words”, *in Proceedings of the 8th ACM International Conference on Image and Video Retrieval (CVIR)*, 2009.

[25] A. Bosch, X. Muñoz, and R. Mart, A review: which is the best way to organize/classify images by content?" *Image and Vision Computing*, vol. 25, no. 6, pp.778-791, 2007.

[26] N. Morioka, S. Satoh, “Building compact local pairwise codebook with joint feature space clustering”, *in Proceedings of the 11th European Conference on Computer Vision (ECCV)*, pp. 692-705, 2010.

[27] J. Sivic and A. Zisserman, “Video Google: a text retrieval approach to object matching in videos,” *in Proceedings of the Ninth International Conference on Computer Vision*, pp. 1470-1478, 2003.

[28] D. Mount and S. Arya, “ANN: A library for approximate nearest neighbor searching. *in Proceedings of the 2nd Annual Fall Workshop on Computational Geometry*, 1997.

[29] A. Andoni, P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions”, *in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science* (FOCS '06) , pp.459-468, 2006.

[30] V. Garcia and E. Debreuve and M. Barlaud. Fast k-nearest neighbor search using GPU. *In Proceedings of the CVPR Workshop on Computer Vision on GPU*, 2008.

[31] X. Zhou, N. Cui, Z. Li, F. Liang, T. S. Huang, “Hierarchical Gaussianization for image classification”, *in Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.