Bag of Visual Words Classifier

Computer Vision and Pattern Recognition Exam

University of Trieste (UniTS)

Marco Tallone

January 2025

Abstract

This report presents the implementation of a Bag of Visual Words (BoW) image classifier. The objective is to build a classifier for scene recognition by building a visual vocabulary from a set of images, representing them as normalized histograms of visual words, and performing multi-class classification. The visual vocabuilary is built by clustering SIFT descriptors extracted from the images, and the classification is performed comparing K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) classifiers. Different kernels have been tested for the SVM classifier, including radial basis function (RBF), χ^2 kernel and pyramid match kernel. Results show that the best performance is achieved by the SVM classifiers, in particular when implemented with a spatial pyramid feature representation to add spatial information to the classic BoW approach.

1 Introduction

The Bag of Visual Words (BoW) model is a popular computer vision technique used for image classification or retrieval. It is based on the idea of treating images as documents and representing them as normaized histograms of visual words belonging to a visual vocabulary, which is obtained by clustering local features extracted from a set of images.
The objective of this project is to implement a BoW image classifier for scene recognition by first building a visual vocabulary from a set of test images and then performing multi-class classification using K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) classifiers.
In particular, the visual vocabulary is built by clustering SIFT descriptors extracted from the test images using the K-Means algorithm. Descriptors have been computed both from keypoints detected wit the the SIFT algorithm and from dense sampling of the images with a fixed grid to compare the two approaches.
In the classification phase instead, the performance of a simple KNN classifier is compared with that of different SVM classifiers all adopting the "*one-vs-all*" strategy for multi-class classification. The SVM classifiers differ in the kernel used and in the kind of input features they are trained on.
Additionally, the use of a spatial pyramid matching for natural scene recognition proposed by *Lazebnik et al.* [TODO:add reference] is also tested. This technique is used to add spatial information to the classic BoW approach in an attempt to improve the classification performance.
The following sections are organized as follows. Section 2 presents the dataset used for the experiments and analyzes the images distribution. Section 3 explains the process of building the visual vocabulary from the descriptors computed on the test images. Section 4 shows how images can be represented as normalized histograms of visual words. In section 5 different classifiers are compared for the multi-class classification task while section **??** presents the spatial pyramid matching technique and its application to the BoW model. Finally, section 7 summarizes the resultsof this study together with some final considerations.
With the aim of maintaining a clear and concise report of the work, the following sections will only present the most noticeable results of the study, while further analysis and the detailed model implementations will be available on the authors GitHub repository [TODO:add reference].

2 Dataset Description

The dataset adopted for the implementation and the assessment of the BoW classifier in this study is the *15-Scenes* dataset from *Lazebnik et al.* [TODO:reference]. This dataset consists of a total of 4485 grayscale images, each belonging to one of 15 different scene categories (*office, kitchen, living room, bedroom, store, industrial, tall building, inside city, street, highway, coast, open country, mountain, forest, suburb*). The images come already splitted into a training set of 1500 images, with a uniform distribution of 100 images per category, and a test set of 2985 images, for which the distribution of images is not uniform as shown in Figure 2.1.

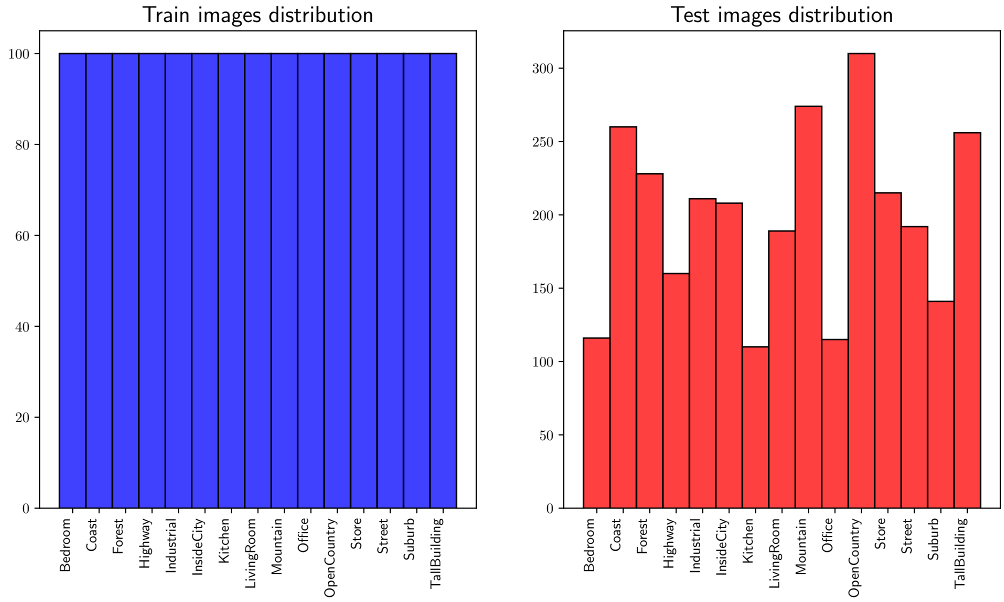


Figure 2.1: Distribution of images in train and test sets for the *15-Scenes* dataset.

Moreover, it's worth mentioning that the images in the dataset are of different sizes and aspect ratios, with the vaste majority of them being 256 × 256 pixels. This is an important aspect to consider since it directly impacts the feature extraction phase as explained in Section 3.

3 Visual Vocabulary Construction

The process of building a visual vocabulary can be split in two main steps:

1. **Feature extraction**: sampling of SIFT descriptors from train images.
2. **Clustering**: grouping of the extracted descriptors into k clusters.

3.1 Feature Extraction

The first step in building a visual vocabulary is to extract features from the images in the training set. In this case, SIFT descriptors are used as features and they have been extracted following two different approaches:

1. using the **SIFT detector**: the SIFT detector is used to detect keypoints in the images and then the corresponding SIFT descriptors are computed on these keypoints.
2. sampling on a **dense regular grid**: following the approach adopted by *Lazebnik et al.* [TOOD: add reference], SIFT descriptors can also be computed from keypoints sampled on a regular dense grid over each image. Following the approach of the paper, a regular grid with spacing of 8 pixels between each keypoint has been used.

As highlighted by *Lazebnik et al.* [TOOD: add reference] reporting the original evaluation from *Fei-Fei and Perona* [TOOD: add reference], sampling descriptors from a regular grid should intuitively work better from scene classification tasks, as this method allows to capture uniform regions such as sky, calm water, or road surfaces.

Both sampling methods have been tested and compared, with the results shown in later section 6. For the first approach, the SIFT detector has been initialized to retain only the best 500 features in each image resulting in a total of 593 006 keypoints, while for the second approach 1 482 434 keypoints have been collected.

Notice that with both approaches the exact number of keypoints extracted in each image can change from one image to another. In the first case, this is due to the content of the images and the actual features present in them[1], while in the second case this is due to the different sizes and aspect ratios of the images in the dataset. Figure 3.1 shows an example of this phenomenon, where a different number of keypoints has been detected by the SIFT detector in two different images.

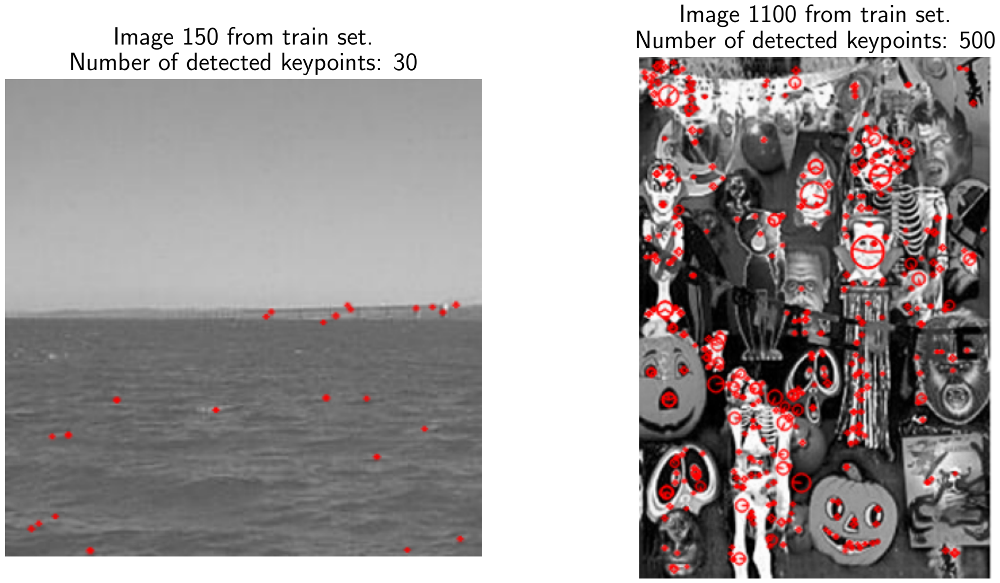


Image 150 from train set.
Number of detected keypoints: 30
Image 1100 from train set.
Number of detected keypoints: 500

Figure 3.1: Example of keypoints detected by the SIFT detector in two different images. In the left side image only 30 keypoints have been detected, while in the right side image all the desired 500 keypoints have been detected.

3.2 Clustering

Following the previous step, clustering has been performed on a random subset of extracted SIFT descriptors to build the visual vocabulary. The size of the subset has been fixed to 10 000 for descriptors sampled using SIFT detector and to 25 000 for descriptors sampled on a regular grid. This has been done to approximatively maintain the same ratio between the total number of sampled descriptors and the number of descriptors used for clustering in the two cases.

In paticular, the *k-means* algorithm has been used to cluster the descriptors into K clusters, where K is the size of the visual vocabulary.

For this purpose, different values of K have been tested, ranging from $K = 50$ to $K = 1000$. After some tests, the value of $K = 400$ has been fixed to build the visual vocabulary. This choice is motivated by two main reasons. First, since the value of the silhouette score decreases noticeably after $K = 400$ as it's possible to see in figure 3.2, this value has been chosen as a good trade-off between the quality of the clustering and the computational cost of the clustering process. Second, considering the purpose of this project, the value of $K = 400$ allows for a direct comparison with the results reported in the original paper by *Lazebnik et al.* [TOOD: add reference], where the same value has been used.

After the clustering process, the obtained centroids representation the visual words of the vocabulary will be 128-dimensional vectors.

[1] Intuitively, images with a lot of texture or with a lot of edges will have more keypoints than images with uniform regions such as sky or water.

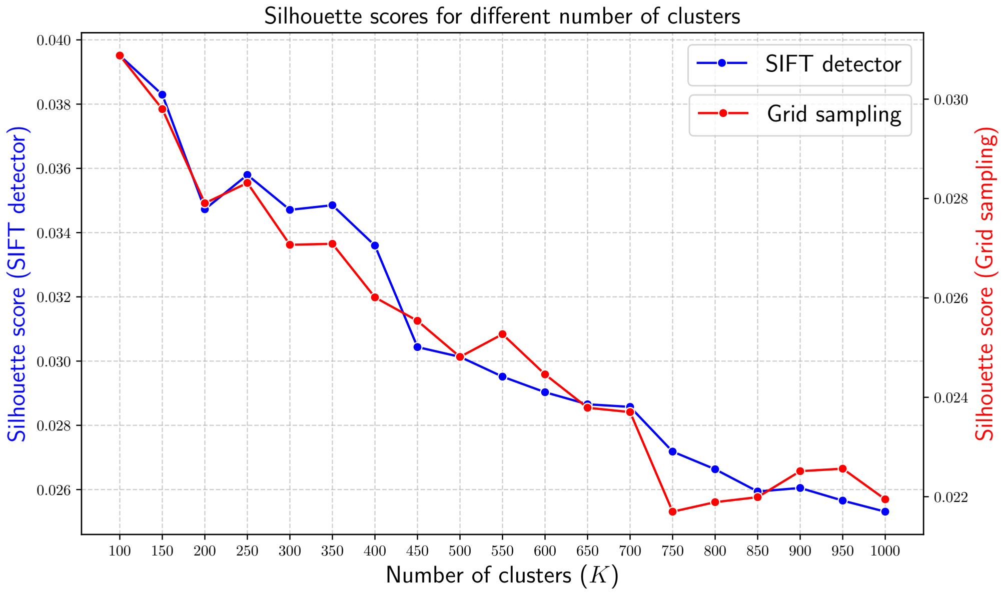


Figure 3.2: Silhouette score for different values of K resulting from clustering on SIFT descriptors sampled using the SIFT detector (blue) and on a regular grid (red).As expected the value decreases with the increase of K. Relatively significant decreases are observed after $K = 200$ and $K = 400$ for the two cases.

4 Image Representation

A important step in the Bag of Visual Words (BoW) pipeline is the representation of images as normalized histograms of visual words.
In this phase, the SIFT descriptors previously extracted from the images are assigned to the visual words in the vocabulary by means of a clustering algorithm. Then, for each image in the dataset, a histogram is built by counting the occurrences of each visual word in the given image.
The result is a fixed-length representation in which images are seen as normalized histograms having k bins, each corresponding to a visual word in the vocabulary. Such representation is used as input to the classifiers in section 5 for the scene recognition task.
Additionally, in order to account for the relevance of the visual words, the *term frequency-inverse document frequency (TF-IDF)* weighting scheme has also been implemented and compared. In this second case, the elements of the histograms representation have been computed as:

$$t_i = \frac{n_{id}}{n_d} \cdot \log\left(\frac{N}{N_i}\right) \tag{4.1}$$

where n_{id} is the number of occurrences of the i-th visual word in image d, n_d is the total number of visual words in image d, N is the total number of images in the dataset, and N_i is the number of images containing the i-th visual word.

5 Classification

In order to perform the scene recognition task, two main classes of classifiers have been implemented: K-Nearest Neighbors (KNN) and Support Vector Machines (SVM). This section describes these classifiers while the results are presented in the next section 6.

5.1 K-Nearest Neighbors (KNN)

A K-Nearest Neighbors classifier taking as input features the normalized histograms of visual words has been implemented. The classifier assigns to each image in the test set

the label of the majority class among its k^2 nearest neighbors in the training set.
In the simple case of $k = 1$, the label of the closest histogram in the training set is assigned to the test image. A slightly better result can be achieved by performing a linear search for the hyperparameter k over the range $[1, 50]$ using the average accuracy as assessment metric. For each value of k in the range, the accuracy of the corresponding KNN classifier is computed and stored. At the end, the value of k that maximizes the accuracy is selected and the performance of that classifier is evaluated.

5.2 Support Vector Machines (SVM)

A series of multi-class Support Vector Machine (SVM) classifiers have been implemented following the "*one-vs-all*" strategy. For each possible class, a single binary classifier is trained and the final prediction is obtained by selecting the class with the highest confidence score. Each binary classifier is trained taking as input features either the normalized histograms or the TF-IDF weighted histograms, and modified ground truth labels where the class of interest is labeled as $+1$ and all other classes are labeled as -1.
Different kernels for these SVM classifiers have ben tested and compared. Initially, the default radial basis function (RBF) kernel has been adopted. Subsequently, the generalized Gaussian kernel based on the χ^2 distance shown in equation 5.1 has been tested.

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \sum_i \frac{(\mathbf{x}_i - \mathbf{x}'_i)^2}{\mathbf{x}_i + \mathbf{x}'_i}\right) \tag{5.1}$$

This kernel, implemented with the parameter $\gamma = \frac{1}{2}$, is well known to be effective in histogram comparison tasks.
Finally the *spatial pyramid kernel* described in section 5.3 has also been tested.

5.3 Spatial Pyramid Matching

In an attempt to improve the classification performance of the SVM classifiers by adding spatial information to the classic BoW approach, a spatial pyramid feature representation has been implemented as described by *Lazebnik et al.* [TODO:cite paper].
The idea behind the spatial pyramid is to repeatedly subdivide each image into subsequently finer grids at different resolution levels $\ell \in \{0, \dots, L\}$, and to compute the histograms $H_\ell(i)$ of visual words for each i^{th} grid at level ℓ. More precisely, at level ℓ each image is divided into 2^ℓ cells along each dimension, and the count of visual words is computed for each cell.
It's important to notice that in this case the features are extracted from images by sampling SIFT descriptors from a regular grid with spacing of 8 pixels between keypoints. Therefore, the feature extraction using the SIFT detector has not been performed in this case both to follow the same approach of the original paper and also to avoid the possibility of having empty cells[3].
All the hstograms are then weighted according to a multiplicative weighting scheme that favors features counts computed at higher (finer resolution) levels of the pyramid while penalizing those at lower levels. In particular, if L is the number of levels in the pyramid, the weigths associated to the histograms at level ℓ are computed as:

$$w_0 = \frac{1}{2^L} \quad \text{and} \quad w_\ell = \frac{1}{2^{L-\ell+1}} \quad \text{for} \quad \ell \in \{1, \dots, L\} \tag{5.2}$$

The resulting weighted histograms are then stacked together to form, for each image, a single extended multi-level descriptor that can be used as input to the SVM classifiers. Such descriptor, not only contains the information about the visual words, but also encodes locality information at different scales which was missing in the standard BoW approach.
The SVM classifiers receiving these input feature vectors must be trained using the *spatial pyramid kernel* which correctly computes the similarity between any two extended descriptors by performing a (weighted) sum of histograms intersections. Formally, given a visual vocabulary of size K and two images X and Y represented by their extended

[2] Notice in this context k is a hyperparameter of the classifier and not to be confused with the number of clusters K used to build the visual vocabulary.

[3] Which might happen if a given cell contains a uniform region of the image.

descriptors

$$\begin{aligned}H^X &= \{H^X_{\ell.k}(i) \mid \forall \ell \in \{0,\dots,L-1\}, \forall k \in \{0,\dots,K-1\}, \forall i \in \{0,\dots,2^{2\ell}-1\}\}\\ H^Y &= \{H^Y_{\ell.k}(i) \mid \forall \ell \in \{0,\dots,L-1\}, \forall k \in \{0,\dots,K-1\}, \forall i \in \{0,\dots,2^{2\ell}-1\}\}\end{aligned}$$

then the *pyramid match kernel* betwen the two images is computed as

$$K^L(X,Y) = \sum_{\ell=0}^{L-1} \sum_{k=0}^{K-1} \sum_{i=0}^{2^{2\ell}-1} \min\left(H^X_{\ell.k}(i), H^Y_{\ell.k}(i)\right) \tag{5.3}$$

Hence, the kernel in equation 5.3 has been used to compute the Gram matrix both for the train and test sets, and the SVM classifiers have been trained and evaluated accordingly.

6 Results

The results obtained with all the classifiers, the two feature extraction methods and the different image representations are summarized in Table 6.1. The average accuracy over all the classes has been used as the main assessment metric for all the models. However, confusion matrices have also been computed[4] in order to better understand which classes are more difficult to classify and these are reported in Appendix A1.

All the expreimental results have been obtained by running the models on a machine equipped with an Intel® Core™ i7–8565U CPU @ 4.60 GHz and 8 GB of RAM.

First of all a dummy classifier, that always predict the most frequent class (*Open Country*) on the test set, has been used as baseline for comparison for all the other models. The average accuracy obtained on the test set is 10.39 %, and is of course independed from the input feature representation.

The KNN classifiers achieved their best results using the normalized histograms representation and the dense grid feature sampling both in the case of single and multiple neighbors. However, the best accuracies obtained do not exceed 32.36 % for the single neighbor case and 38.76 % for the multiple neighbors case (obtained with $k = 20$). The results are surely better than the baseline classifier but still far from being satisfactory as a lot of confusion between classes is present as highlighted by the relative confusion matrices in figures A.1 and A.2.

The results collected with the SVM classifiers have instead been more promising. In this case the best performances have been obtained from the features extracted by the SIFT detector. The SVMs trained with the default RBF kernel achieved a top accuracy of 50.89 %, while the ones using the χ^2 kernel reached 51.49 %. Hence, a marginal improvement has been obtained by using the specialized kernel, but the results are still comparable. From the confusion matrices shown in A.3 and A.4 a noticeable (but understandable) confusion arises between the *Open Country* and *Forest* classes. The *Living Room*, *Bedroom* and *Kitchen* classes are also often misclassified among each other. The *Industrial* class, instead, is the one with the lowest overall accuracy.

By a large margin, the best accuracy has been obtained by the SVM classifiers using the spatial pyramid matching kernel. In this case, by sampling features from a regular grid and adopting the extended weighted histogram representation for the images, an accuracy of 75.54 % has been achieved. This result is significantly better than the other classifiers and is a clear indication that the spatial information is crucial for the scene recognition task. The superiority of this approach is also confirmed by the almost completely diagonal confusion matrix (Figure A.5), in which only a few misclassifications are present, mainly between the *Living Room* and *Bedroom* classes. Although higher than the other classifiers, the lowest accuracy is still measured for the *Industrial* class.

Finally, from Table 6.1 we can also notice that, when the SIFT detector has been adopted as feature extraction method, the performances of the classifiers when using either the normalized histogram or the TF-IDF representation for the input features are very similar. Instead, this has not been the case when the dense grid sampling has been used. In this case, the classifiers have achieved better results using the normalized histograms representation and significantly lower accuracies have been measured when the TF-IDF weighting scheme has been applied. However, such result is not completely unexpected and might be explained by the fact that grid sampling doesn't necessarily guarantee to

[4] Only for the *dense grid* sampling method and the normalized histogram representation for conistency reasons as explained in Appendix A1.

extract characteristic and significative features from the images as the SIFT detector does. For this reason, the TF-IDF weighting scheme might fail in properly enhancing the importance of the most discriminative visual words in each image.

Classifier	**Feature Sampling**	**Image Representation**	
		Histogram	*TF-IDF*
Dummy Classifier	*SIFT detector*	10.39 %	10.39 %
	Dense grid	10.39 %	10.39 %
1-NN	*SIFT detector*	31.49 %	31.83 %
	Dense grid	32.36 %	23.48 %
k-NN	*SIFT detector*	37.19 %	36.42 %
	Dense grid	38.76 %	28.31 %
SVM (RBF)	*SIFT detector*	50.65 %	50.89 %
	Dense grid	50.08 %	42.01 %
SVM (χ^2)	*SIFT detector*	51.49 %	50.95 %
	Dense grid	50.12 %	42.81 %
SVM (Pyramid Matching Kernel)	*Dense grid*	75.54 %	

Table 6.1: Average accuracies for all the implemented classifiers comparing feature extraction methods and image representations. Notice that the final result for the SVM with the pyramid matching kernel has been obtain by adopting the extended weighted histogram representation as presented in Section 5.3.

7 Conclusions

In conclusion, this study has shown the implementation of a Bag of Visual Words (BoW) classifier for the scene recognition task from the construction of a visual vocabulary by perfoming clustering on SIFT descriptors to the classification of images using classic machine learning technques as K-Nearest Neighbors (KNN) and Support Vector Machines (SVM).

From the results obtained, it clearly emerges that the SVM classifiers have outperformed the KNN classifiers in terms of accuracy, but the latter at least surpassed the dummy classifier performance used as baseline.

The usage of the specialized χ^2 kernel for the SVM classifiers has also shown to be marginally beneficial with respect to the default RBF kernel, even if the performace difference is not significant.

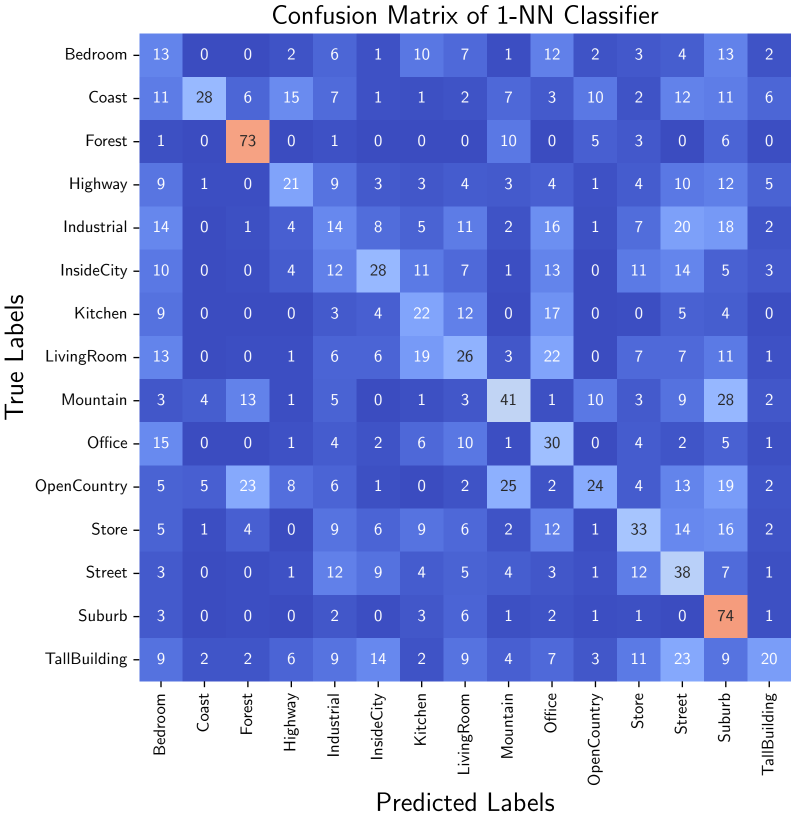
Moreover, from the results no significant differece emerges between the usage of the normalized histograms or the TF-IDF representation for the input features when the SIFT detector has been used for feature extraction. Instead, an overall decrease in performace has been observed when using a combination of dense grid sampling and TF-IDF representation.

Ultimately, the best accuracy has been achieved by the SVM classifiers using the spatial pyramid approach proposed by *Lazebnik et al.* Altough the final accuracy of 75.54 % doesn't quite match the one obtained by the authors of the original paper, the results are by far better than the other classifiers and this confirms the importance of adding spatial information to the classic BoW approach for the scene recognition task. The main differences in the results of this study from the ones obtined by *Lazebnik et al.* might be due either to the placement of the sampling grid of keypoints in the images or to the custering parameters used to build the visual vocabulary (e.g. the size of the random sample used or the features selected for clustering). Nevertheless, the results obtained are still satisfactory and confirm the validity of the BoW approach for the scene recognition task.

A Appendix

A1 Confusion Matrices

Here the confusion matrices for all the classifiers are reported. These matrices refers to the case in which features have been extracted by a SIFT descriptor from keypoints sampled on a dense grid on the images and the input representation is the normalized histograms of visual words. This has been done to maintain the results consistent with the ones obtained using the spatial pyramid matching approach and to be able to compare the performances of the different classifiers. However, similar matrices can be computed in the other cases as well.



	Bedroom	Coast	Forest	Highway	Industrial	InsideCity	Kitchen	LivingRoom	Mountain	Office	OpenCountry	Store	Street	Suburb	TallBuilding
Bedroom	13	0	0	2	6	1	10	7	1	12	2	3	4	13	2
Coast	11	28	6	15	7	1	1	2	7	3	10	2	12	11	6
Forest	1	0	73	0	1	0	0	0	10	0	5	3	0	6	0
Highway	9	1	0	21	9	3	3	4	3	4	1	4	10	12	5
Industrial	14	0	1	4	14	8	5	11	2	16	1	7	20	18	2
InsideCity	10	0	0	4	12	28	11	7	1	13	0	11	14	5	3
Kitchen	9	0	0	0	3	4	22	12	0	17	0	0	5	4	0
LivingRoom	13	0	0	1	6	6	19	26	3	22	0	7	7	11	1
Mountain	3	4	13	1	5	0	1	3	41	1	10	3	9	28	2
Office	15	0	0	1	4	2	6	10	1	30	0	4	2	5	1
OpenCountry	5	5	23	8	6	1	0	2	25	2	24	4	13	19	2
Store	5	1	4	0	9	6	9	6	2	12	1	33	14	16	2
Street	3	0	0	1	12	9	4	5	4	3	1	12	38	7	1
Suburb	3	0	0	0	2	0	3	6	1	2	1	1	0	74	1
TallBuilding	9	2	2	6	9	14	2	9	4	7	3	11	23	9	20

Figure A.1: Confusion matrix for the 1-NN classifier.

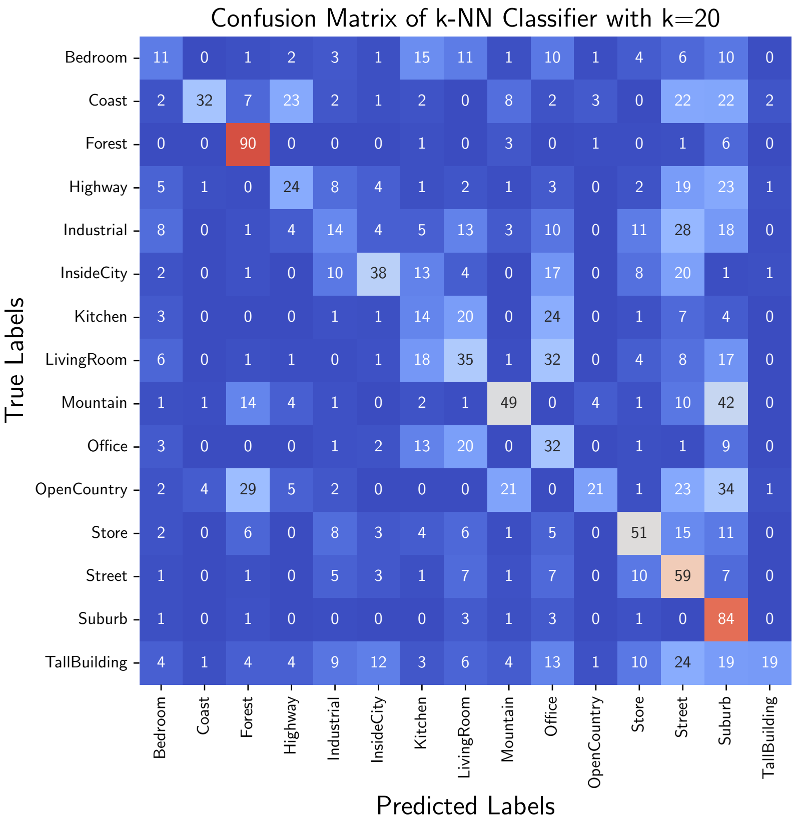


Figure A.2: Confusion matrix for the k-NN classifier with $k = 20$.

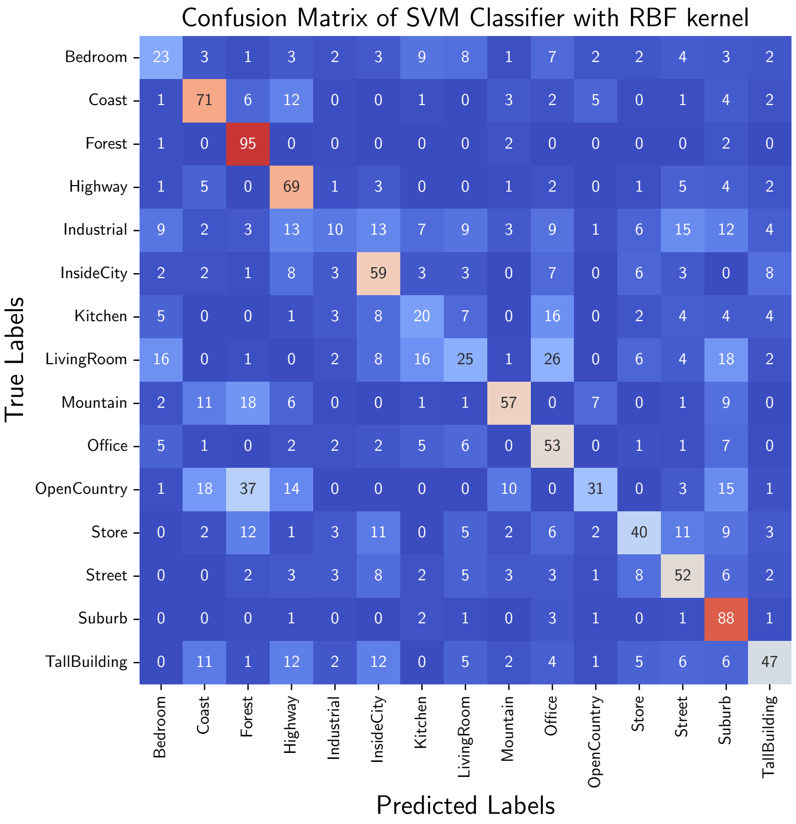
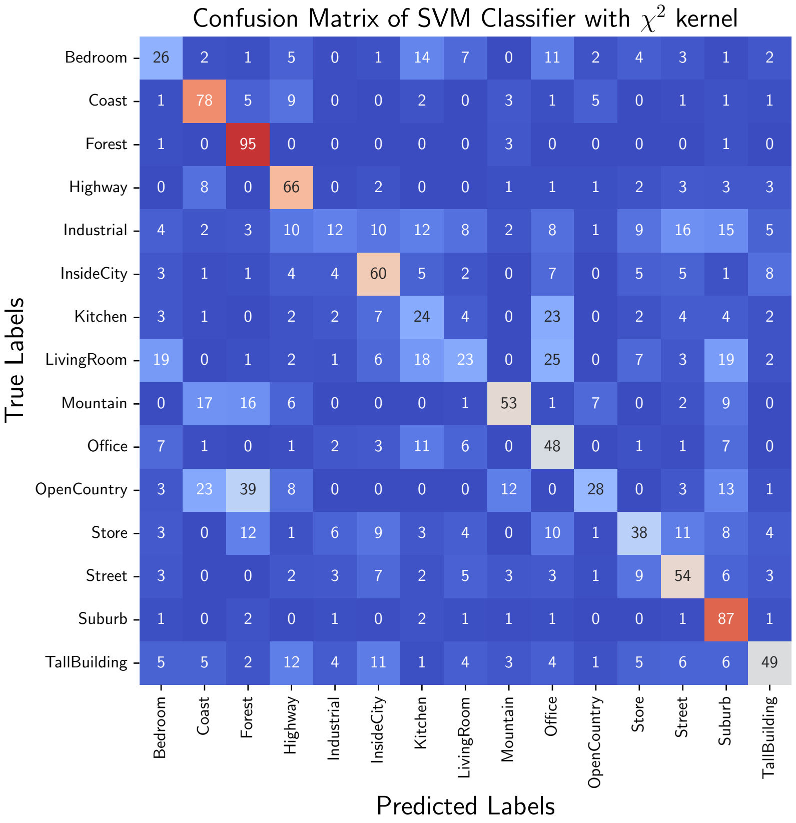


Figure A.3: Confusion matrix for the SVM classifier with RBF kernel.



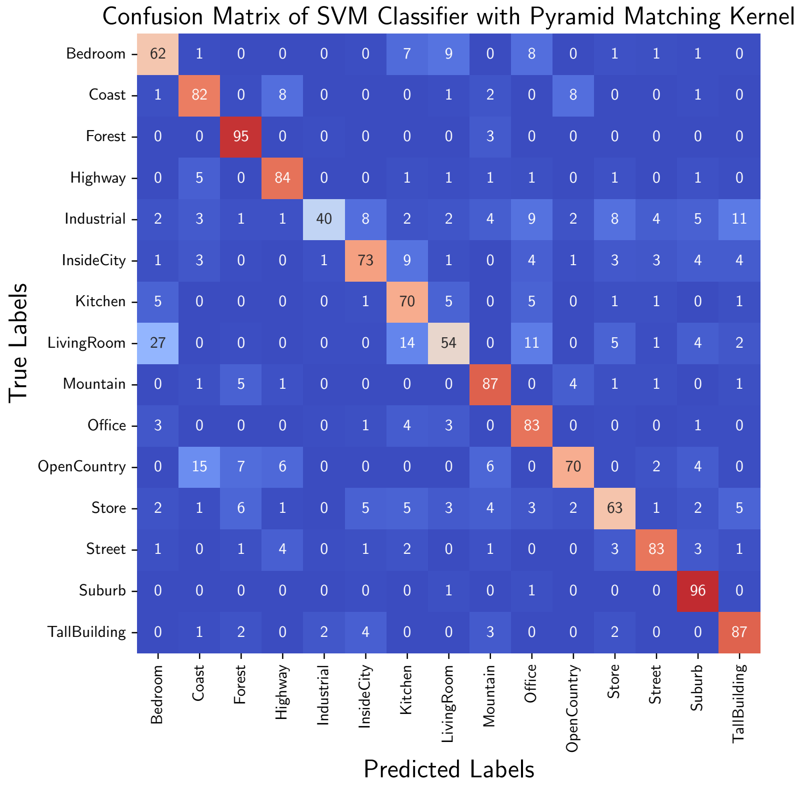
Confusion Matrix of SVM Classifier with χ^2 kernel

True Labels

Predicted Labels

	Bedroom	Coast	Forest	Highway	Industrial	InsideCity	Kitchen	LivingRoom	Mountain	Office	OpenCountry	Store	Street	Suburb	TallBuilding
Bedroom	26	2	1	5	0	1	14	7	0	11	2	4	3	1	2
Coast	1	78	5	9	0	0	2	0	3	1	5	0	1	1	1
Forest	1	0	95	0	0	0	0	0	3	0	0	0	0	1	0
Highway	0	8	0	66	0	2	0	0	1	1	1	2	3	3	3
Industrial	4	2	3	10	12	10	12	8	2	8	1	9	16	15	5
InsideCity	3	1	1	4	4	60	5	2	0	7	0	5	5	1	8
Kitchen	3	1	0	2	2	7	24	4	0	23	0	2	4	4	2
LivingRoom	19	0	1	2	1	6	18	23	0	25	0	7	3	19	2
Mountain	0	17	16	6	0	0	0	1	53	1	7	0	2	9	0
Office	7	1	0	1	2	3	11	6	0	48	0	1	1	7	0
OpenCountry	3	23	39	8	0	0	0	0	12	0	28	0	3	13	1
Store	3	0	12	1	6	9	3	4	0	10	1	38	11	8	4
Street	3	0	0	2	3	7	2	5	3	3	1	9	54	6	3
Suburb	1	0	2	0	1	0	2	1	1	1	0	0	1	87	1
TallBuilding	5	5	2	12	4	11	1	4	3	4	1	5	6	6	49

Figure A.4: Confusion matrix for the SVM classifier with χ^2 kernel.



Confusion Matrix of SVM Classifier with Pyramid Matching Kernel

True Labels

Predicted Labels

	Bedroom	Coast	Forest	Highway	Industrial	InsideCity	Kitchen	LivingRoom	Mountain	Office	OpenCountry	Store	Street	Suburb	TallBuilding
Bedroom	62	1	0	0	0	0	7	9	0	8	0	1	1	1	0
Coast	1	82	0	8	0	0	0	1	2	0	8	0	0	1	0
Forest	0	0	95	0	0	0	0	0	3	0	0	0	0	0	0
Highway	0	5	0	84	0	0	1	1	1	1	0	1	0	1	0
Industrial	2	3	1	1	40	8	2	2	4	9	2	8	4	5	11
InsideCity	1	3	0	0	1	73	9	1	0	4	1	3	3	4	4
Kitchen	5	0	0	0	0	1	70	5	0	5	0	1	1	0	1
LivingRoom	27	0	0	0	0	0	14	54	0	11	0	5	1	4	2
Mountain	0	1	5	1	0	0	0	0	87	0	4	1	1	0	1
Office	3	0	0	0	0	1	4	3	0	83	0	0	0	1	0
OpenCountry	0	15	7	6	0	0	0	0	6	0	70	0	2	4	0
Store	2	1	6	1	0	5	5	3	4	3	2	63	1	2	5
Street	1	0	1	4	0	1	2	0	1	0	0	3	83	3	1
Suburb	0	0	0	0	0	0	0	1	0	1	0	0	0	96	0
TallBuilding	0	1	2	0	2	4	0	0	3	0	0	2	0	0	87

Figure A.5: Confusion matrix for the SVM classifier with spatial pyramid matching kernel.