Computer Vision project

Michele Sanna

February 14, 2024

Abstract

In this work will be implemented the second problem proposed in the project assignment. The assignment requires to implement a bag-of-words approach to solve an image classification task. Every section of this report will cover the choices concerning the corresponding step in the assignment.

1 Building a visual vocabulary

For each image in the dataset 50 SIFT descriptors have been sampled using the OpenCV detector, obtaining approximately 75k descriptors for the whole training dataset. Then the descriptors have been clustered using a K-means algorithm. The initial strategy to determine the optimal number of centroids for the K-mean clustering was to look at the curve of the average silhouette score, but it had a decreasing trend without any significant maxima. The number of centroids that it's used is therefore chosen based on the best performance in terms of accuracy.

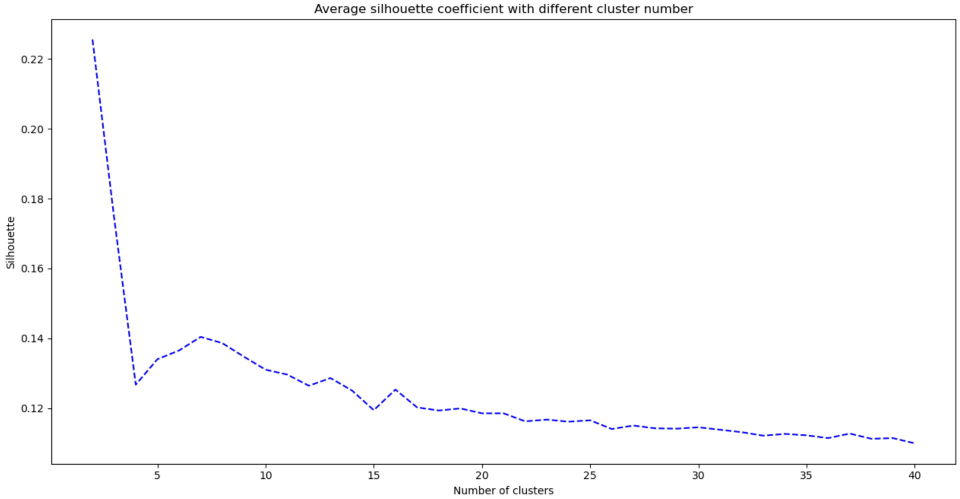


Figure 1: Average silhouette score for each cluster

If we look at the plot in Figure 2 we can clearly see that the accuracy increases as the number of clusters increases, and the curve has a logarithm-like shape. Therefore when we reach an high number of centroids, increasing the number of centroids leads to a negligible increase in accuracy, but leads to a substantial increase of execution time, especially when using custom SVM kernels.

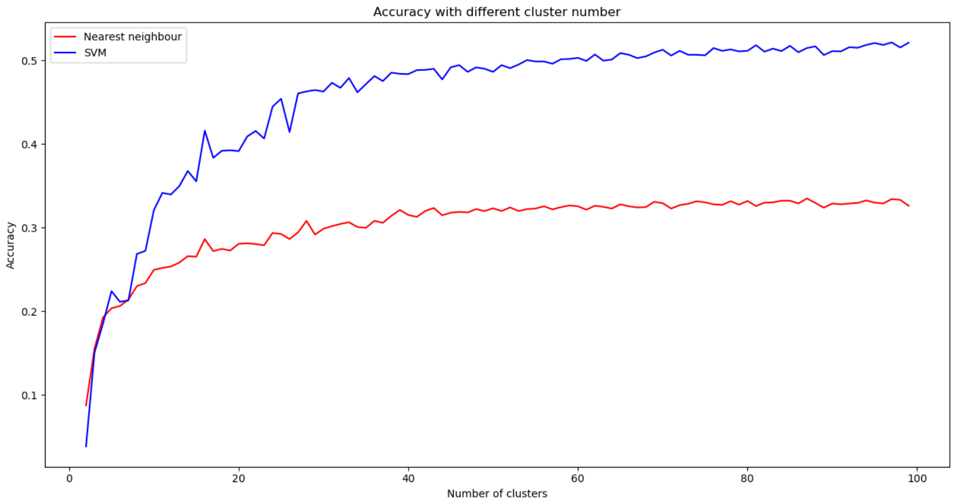


Figure 2: Accuracy of a gaussian kernel SVM for each number of clusters

For this reason the number of centroids that was chosen is 50, in order to have a good compromise between accuracy and execution time.

2 Representing each image as an histogram

To represent each image as an histogram an high number of SIFT descriptors is computed (up to 800). Then in order to represent each image of the training set as a normalized histogram having k bins three different methods have been tested:

- Regular voting: for each SIFT descriptor, the value of the bin corresponding to the closest centroid is increased by 1. Once that all the SIFT descriptors have "voted" the vector is normalized.
- Soft voting: for each SIFT descriptor, the value of each bin is increased by the reciprocal of the distance between the descriptor and the centroid corresponding to the bin.
- Alternative soft voting: for each SIFT descriptor, the value of each bin is increased by e^{-d}, where d is the distance between the descriptor and the centroid corresponding to the bin.

For all the methods, in order to measure the distance between the centroid and the descriptor, the euclidean distance has been used. Also the normalization was the same for all the methods (l2 normalization). To determine which was the best method I measured the accuracy (assessed with a 10-fold cross validation) of the linear SVM classifier using each one of those methods.

Fitting a bag-of-words dataset created with the regular voting method led to an accuracy of 0.407, while the soft voting methods led to an accuracy of 0.351 for the first soft voting method and an accuracy of 0.424 for the alternative soft voting method. Therefore the alternative soft voting method has been chosen

3 Nearest Neighbour Classifier

In addition to the nearest neighbour classifier (that considered only the closest datapoint to determine the class) has been also trained a Knn classifier that considers the closest 5 datapoints and weights them by the inverse of their distance.

The entries of the confusion matrices below have been normalized for the number of samples of each class. The overall test accuracies are reported in Table 1

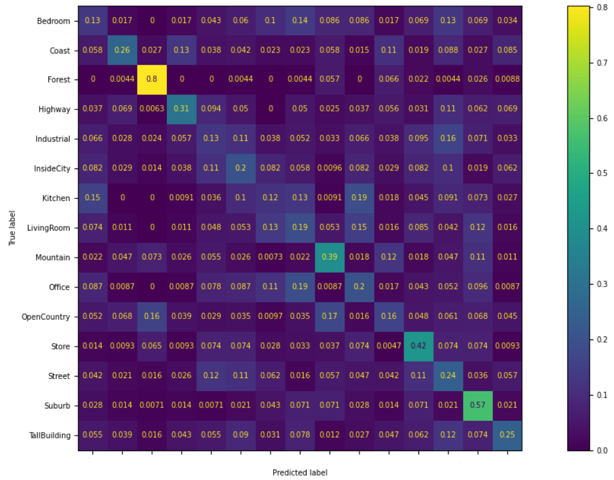


Figure 3: Confusion matrix for the nearest neighbour classifier

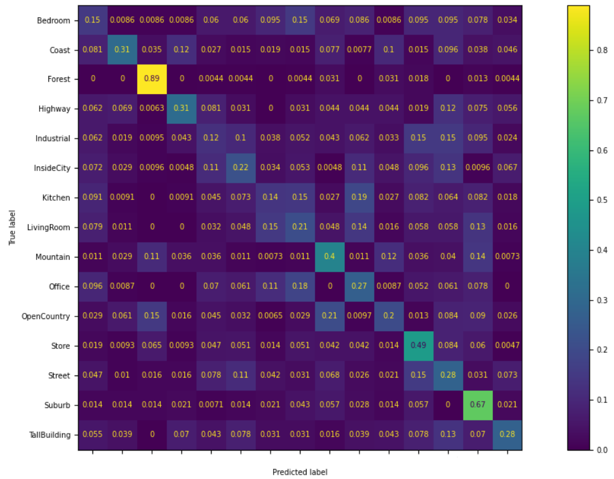


Figure 4: Confusion matrix for the K nearest neighbours classifier

In those plots the accuracy of both models for the class 'forest' stands out. As we will see in the next sections, this happens also with SVM classifiers. I had manually checked for duplicates between test set and training set, but it appears there were none. Probably the features of the images of this class (maybe the leaves?) are well captured by the SIFT descriptors.

	Nearest neighbour	K nearest neighbours
Accuracy	0.299	0.337

Table 1: Test accuracy for the nearest neighbours classifiers

4 One Vs Rest SVM Classifier

4.1 Kernels

In order to obtain the best accuracy, a variety of kernel have been tested:

- Linear kernel
- Gaussian kernel with euclidean distance
- Generalized Gaussian kernel with chi-squared distance
- Generalized Gaussian kernel with earth mover's distance

The plain Gaussian kernel and the chi-squared Gaussian kernel are those that performed better, on the other hand the linear kernel led to low accuracy scores and the earth mover's distance kernel has been discarded for its computing speed. Indeed, even if for this task the earth's mover distance kernel might be the most adequate kernel, my implementation of it (that uses a scipy linear programming solver, so a C backend) requires 25 second to compute each row of the distance matrix on my laptop, so the training of each of the 15 classifiers of the one-vs-rest SVM would require a total of 156 hours.

4.2 Assessing

To assess the best gamma parameter (i.e. 2.8) for the chi-squared generalized Gaussian kernel a 10 fold cross validation has been used, while for the euclidean distance Gaussian kernel the gamma parameter has been set to:

$$\frac{n\_features}{\sigma^2}$$

. The cross-validation accuracy for each of those kernels is reported in Table 2

	Linear kernel	Gaussian kernel	χ^2 gaussian kernel	EMD gaussian kernel
Accuracy	0.417	0.441	0.456	

Table 2: Cross-validation accuracy for each kernel

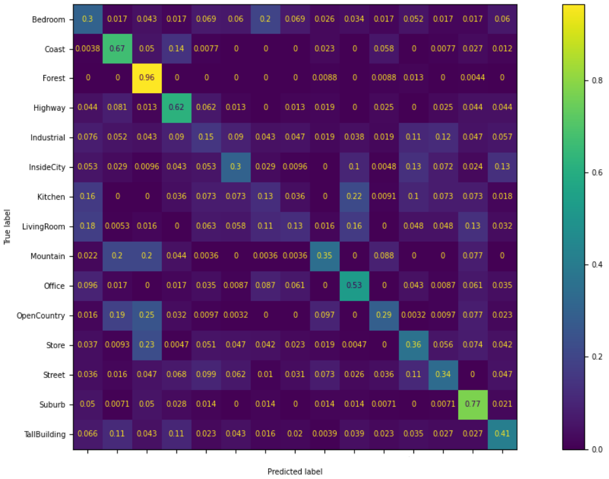
5 SVM evaluation

This section contains the confusion matrix of the one vs rest SVM tested on the test set. The entries of the matrices have been normalized for the number of samples of each class. The overall accuracies are reported in Table 3.

	Linear kernel	Gaussian kernel	χ^2 gaussian kernel	EMD gaussian kernel
Accuracy	0.424	0.465	0.495	

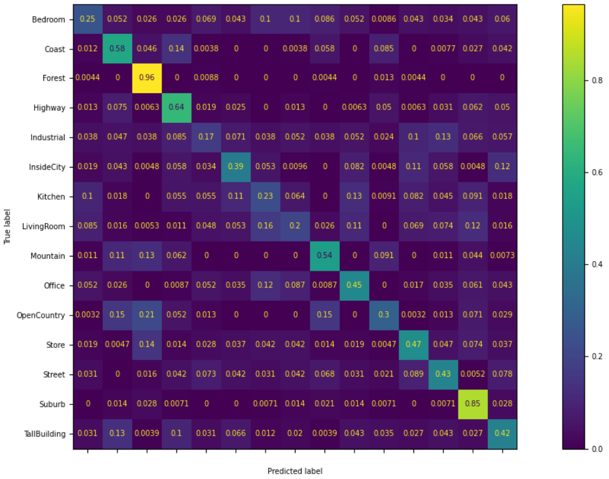
Table 3: Test set accuracy for each kernel

Is noteworthy that the accuracy in the test set is noticeably higher than the cross-validation accuracy. I think that this is due to the fact that, for coincidence, some classes for which the models perform better (like 'forest') are overrepresented in the test dataset.



Bedroom	0.3	0.017	0.043	0.017	0.069	0.06	0.2	0.069	0.026	0.034	0.017	0.052	0.017	0.017	0.06
Coast	0.0038	0.67	0.05	0.14	0.0077	0	0	0	0.023	0	0.058	0	0.0077	0.027	0.012
Forest	0	0	0.96	0	0	0	0	0	0.0088	0	0.0088	0.013	0	0.0044	0
Highway	0.044	0.081	0.013	0.62	0.062	0.013	0	0.013	0.019	0	0.025	0	0.025	0.044	0.044
Industrial	0.076	0.052	0.043	0.09	0.15	0.09	0.043	0.047	0.019	0.038	0.019	0.11	0.12	0.047	0.057
InsideCity	0.053	0.029	0.0096	0.043	0.053	0.3	0.029	0.0096	0	0.1	0.0048	0.13	0.072	0.024	0.13
Kitchen	0.16	0	0	0.036	0.073	0.073	0.13	0.036	0	0.22	0.0091	0.1	0.073	0.073	0.018
LivingRoom	0.18	0.0053	0.016	0	0.063	0.058	0.11	0.13	0.016	0.16	0	0.048	0.048	0.13	0.032
Mountain	0.022	0.2	0.2	0.044	0.0036	0	0.0036	0.0036	0.35	0	0.088	0	0	0.077	0
Office	0.096	0.017	0	0.017	0.035	0.0087	0.087	0.061	0	0.53	0	0.043	0.0087	0.061	0.035
OpenCountry	0.016	0.19	0.25	0.032	0.0097	0.0032	0	0	0.097	0	0.29	0.0032	0.0097	0.077	0.023
Store	0.037	0.0093	0.23	0.0047	0.051	0.047	0.042	0.023	0.019	0.0047	0	0.36	0.056	0.074	0.042
Street	0.036	0.016	0.047	0.068	0.099	0.062	0.01	0.031	0.073	0.026	0.036	0.11	0.34	0	0.047
Suburb	0.05	0.0071	0.05	0.028	0.014	0	0.014	0	0.014	0.014	0.0071	0	0.0071	0.77	0.021
TallBuilding	0.066	0.11	0.043	0.11	0.023	0.043	0.016	0.02	0.0039	0.039	0.023	0.035	0.027	0.027	0.41

Figure 5: Confusion matrix for one-vs-rest SVM with a linear kernel



Bedroom	0.25	0.052	0.026	0.026	0.069	0.043	0.1	0.1	0.086	0.052	0.0086	0.043	0.034	0.043	0.06
Coast	0.012	0.58	0.046	0.14	0.0038	0	0	0.0038	0.058	0	0.085	0	0.0077	0.027	0.042
Forest	0.0044	0	0.96	0	0.0088	0	0	0	0.0044	0	0.013	0.0044	0	0	0
Highway	0.013	0.075	0.0063	0.64	0.019	0.025	0	0.013	0	0.0063	0.05	0.0063	0.031	0.062	0.05
Industrial	0.038	0.047	0.038	0.085	0.17	0.071	0.038	0.052	0.038	0.052	0.024	0.1	0.13	0.066	0.057
InsideCity	0.019	0.043	0.0048	0.058	0.034	0.39	0.053	0.0096	0	0.082	0.0048	0.11	0.058	0.0048	0.12
Kitchen	0.1	0.018	0	0.055	0.055	0.11	0.23	0.064	0	0.13	0.0091	0.082	0.045	0.091	0.018
LivingRoom	0.085	0.016	0.0053	0.011	0.048	0.053	0.16	0.2	0.026	0.11	0	0.069	0.074	0.12	0.016
Mountain	0.011	0.11	0.13	0.062	0	0	0	0	0.54	0	0.091	0	0.011	0.044	0.0073
Office	0.052	0.026	0	0.0087	0.052	0.035	0.12	0.087	0.0087	0.45	0	0.017	0.035	0.061	0.043
OpenCountry	0.0032	0.15	0.21	0.052	0.013	0	0	0	0.15	0	0.3	0.0032	0.013	0.071	0.029
Store	0.019	0.0047	0.14	0.014	0.028	0.037	0.042	0.042	0.014	0.019	0.0047	0.47	0.047	0.074	0.037
Street	0.031	0	0.016	0.042	0.073	0.042	0.031	0.042	0.068	0.031	0.021	0.089	0.43	0.0052	0.078
Suburb	0	0.014	0.028	0.0071	0	0	0.0071	0.014	0.021	0.014	0.0071	0	0.0071	0.85	0.028
TallBuilding	0.031	0.13	0.0039	0.1	0.031	0.066	0.012	0.02	0.0039	0.043	0.035	0.027	0.043	0.027	0.42

Figure 6: Confusion matrix for one-vs-rest SVM with a gaussian kernel

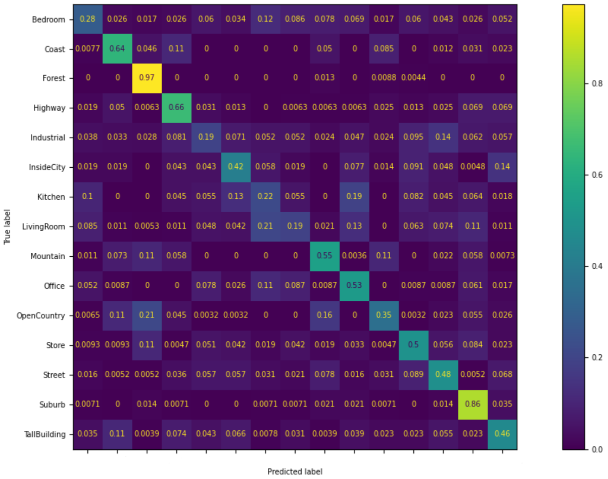


Figure 7: Confusion matrix for one-vs-rest SVM with a χ^2 gaussian kernel