CS 423

Object Classification

Burak Deniz S010031

Introduction:

In this assignment, there are a total of 160 pictures in the train data set and 400 pictures in the test data set. These pictures consist of 4 different classes faces, motorbikes, airplanes, and cars. The goal of this assignment is to train the algorithm in the train data set and predict the right classification for the pictures in the test data set. Technologies available for completing these tasks are Python 3.x, OpenCV, sklearn, and matplotlib.

Problem Statement:

There are a total of 560 pictures of 4 different classes. In each class pictures also differ from one another in subjects like shape size background. The goal is to train the algorithm with the 160 pictures in the train-dataset and predict the classes of the pictures in the test data set with the most accuracy in different experimental settings. There are a total of 12 experiment settings for this assignment. Experimental settings are like this:

Feature Extraction	Feature Descriptor	Dictionary Computation	Feature Quantization	Classifier
Grid - 1	SIFT	Mean Shift	Nearest Neighbor	SVM
Grid - 1	SIFT	K-Means k = 50	Nearest Neighbor	SVM
Grid - 1	SIFT	K-Means k = 250	Nearest Neighbor	SVM
Grid - 1	SIFT	K-Means k = 500	Nearest Neighbor	SVM
Grid - 2	SIFT	Mean Shift	Nearest Neighbor	SVM
Grid - 2	SIFT	K-Means k = 50	Nearest Neighbor	SVM
Grid - 2	SIFT	K-Means k = 250	Nearest Neighbor	SVM

Grid - 2	SIFT	K-Means k = 500	Nearest Neighbor	SVM
Keypoint	SIFT	Mean Shift	Nearest Neighbor	SVM
Keypoint	SIFT	K-Means k = 50	Nearest Neighbor	SVM
Keypoint	SIFT	K-Means k = 250	Nearest Neighbor	SVM
Keypoint	SIFT	K-Means k = 500	Nearest Neighbor	SVM

Proposed Solution:

To complete all the experiments above I used several functions of the OpenCV, sklearn, numpy, scipy along with my own functions in this section I will explain what each function does from Mathematical point of view.

The first step was creating the three different feature extraction methods for the algorithm. Grid - 1 and Grid - 2 is created by my create_grid function. This function takes the image and the size from the user and traverses over the picture creating key points in each size'th pixel and returning all key points in an array. Grid - 1 consists of key points that are created in each 20th pixel and Grid - 2 consists of key points that are created in each 10th pixel. For the Keypoint feature extraction experiments keypoints are created by the sift.detect function. This function uses the Difference of Gaussians which is obtained as the difference of Gaussian blurring of an image with two different σ. Once DoG are found, images are searched for local extreme over scale and space to determine the key points.

When the key points are created I used sift.compute() function to compute the descriptors from the key points I have found. This function does the following: A 16X16 neighborhood around the key point is taken. It is divided into 16 - sub-blocks of 4X4 size. For each sub-block, 8 bin orientation histogram is created. So there are a total of 128 bin values are available. It is represented as a vector to form a keypoint descriptor. This process is done for all the key points in the image.

After finding the keypoints and the descriptors I stack the descriptors vertically and covert the data inside the descriptors to float to make the data available for k-means and mean shift function. When this process is complete I put the three different

descriptors to k-means function with three different k values from the experimental setups.

The k-means algorithm adjusts the classification of the observations into clusters and updates the cluster centroids until the position of the centroids is stable over successive iterations. The stability of the centroids is determined by comparing the absolute value of the change in the average Euclidean distance between the observations and their corresponding centroids against a threshold.

After running the k-means function we start to create the histogram. In this histogram, the algorithm calculates the frequencies of different k-means clusters in an image. When this process is done the algorithm kinda knows which image class is constructed by which key points. After calculating the histogram the algorithm represents them as vectors.

When the vectorization process is done algorithm starts to Standardize features by removing the mean and scaling to unit variance. In a way, the algorithm normalizes the vectors that are coming from the histogram.

For the last step of the training algorithm, the algorithm uses LinearSVC to fit the features and the class into the SVM model also for the mean-shift experiments the algorithm clusters with the MeanShift.fit function. When this process is done the algorithm dumps the necessary outputs to the corresponding pkl files. For example bovw_grid1.pkl consists of the LinearSVC model of grid 1, training pictures classes, standardized features from the feature histogram, the number of clusters used for k-means, and the VOC from the k-means. Whit this process the algorithm completes the training part.

For the testing part of the algorithm, uses some of the steps until the last step with the training algorithm which are:

1: Create the grid and keypoints

2: Use SIFT to find the descriptors.

3: Stack the descriptors vertically in a numpy array.

4: Create the histogram and calculate the frequency.

These 4 steps are done in the training algorithm in the same way they are done in the testing algorithm the only difference is that instead of using the images from the training data set we use images from the testing data set which consists of 400 pictures. When this process is done we use the SVM model that comes from the corresponding pkl file to predict the classification of the test image. This prediction is done on the test images and you can see all the results of the experiments in the result section.

RESULT

For K = 50:

prediction and accuracy of key-points that is found by sift detect algorithm

accuracy = 0.6425

Confusian matrix

Test Pictures / Predictions	Airplanes	Cars	Faces	Motorbikes
Airplanes (100 images)	63	15	13	9
Cars (100 images)	18	71	5	6
Faces (100 images)	14	16	66	4
Motorbikes(100 images	7	22	14	57

prediction and accuracy of grid-1 (Key points are created in each 10th pixel)

accuracy = 0.9625

Confusion Matrix

Test Pictures / Predictions	Airplanes	Cars	Faces	Motorbikes
Airplanes (100 images)	95	2	0	3
Cars (100 images)	0	100	0	0
Faces (100 images)	0	1	91	8
Motorbikes(100 images	0	1	0	99

prediction and accuracy of grid-2 (Key points are created in each 20th pixel)

accuracy = 0.9275

Confusion Matrix

Test Pictures / Predictions	Airplanes	Cars	Faces	Motorbikes
Airplanes (100 images)	88	5	3	4
Cars (100 images)	2	97	0	1
Faces (100 images)	1	0	93	6
Motorbikes(100 images	2	4	1	93

For K = 250 :

prediction and accuracy of key-points that is found by sift detect algorithm

accuracy = 0.7075

Confusion Matrix

Test Pictures / Predictions	Airplanes	Cars	Faces	Motorbikes
Airplanes (100 images)	73	15	9	3
Cars (100 images)	18	74	5	3
Faces (100 images)	9	12	75	4
Motorbikes(100 images	11	16	12	61

prediction and accuracy of grid-1 (Key points are created in each 10th pixel)

accuracy = 0.99

Confusion Matrix

Test Pictures / Predictions	Airplanes	Cars	Faces	Motorbikes
Airplanes (100 images)	98	2	0	0
Cars (100 images)	0	100	0	0
Faces (100 images)	0	0	98	2
Motorbikes(100 images	0	0	0	100

prediction and accuracy of grid-2 (Key points are created in each 20th pixel)

accuracy = 0.985

Confusion Matrix

Test Pictures / Predictions	Airplanes	Cars	Faces	Motorbikes
Airplanes (100 images)	96	2	1	11
Cars (100 images)	0	100	0	0
Faces (100 images)	0	1	99	0
Motorbikes(100 images	0	0	1	99

For K = 500

prediction and accuracy of key-points that is found by sift detect algorithm

accuracy = 0.8625

Confusion Matrix

Test Pictures / Predictions	Airplanes	Cars	Faces	Motorbikes
Airplanes (100 images)	86	6	6	2
Cars (100 images)	4	87	1	8
Faces (100 images)	2	6	84	8
Motorbikes(100 images	2	6	4	88

prediction and accuracy of grid-1 (Key points are created in each 10th pixel)

accuracy = 0.9875

Confusion Matrix

Test Pictures / Predictions	Airplanes	Cars	Faces	Motorbikes
Airplanes (100 images)	97	2	0	1
Cars (100 images)	0	100	0	0
Faces (100 images)	0	0	98	2
Motorbikes(100 images	0	0	0	100

prediction and accuracy of grid-2 (Key points are created in each 20th pixel)

accuracy = 0.995

Confusion Matrix

Test Pictures / Predictions	Airplanes	Cars	Faces	Motorbikes
Airplanes (100 images)	99	0	0	1
Cars (100 images)	0	100	0	0
Faces (100 images)	0	0	100	0
Motorbikes(100 images	0	0	1	99

Mean-Shift Experiments

In the mean shift algorithm, the best result I can get is 25 percent that classified every object as an airplane regardless of grid-1 grid-2 or Keypoint by sift detect

prediction and accuracy of MeanShift (this result is same for k = 50,250,500 and Keypoint)
accuracy = 0.25

Confusion Matrix

Test Pictures / Predictions	Airplanes	Cars	Faces	Motorbikes
Airplanes (100 images)	100	0	0	0
Cars (100 images)	100	0	0	0
Faces (100 images)	100	0	0	0
Motorbikes(100 images	100	0	0	0