CV Homework Set #2 – Object Recognition using BOW

Yuri Khvoles id:329034847
Hadeel Bader id:315692715

A.The classifiers, clustering method and feature extraction methods that you used in your system. Also report the parameters used within.

The classifiers, clustering method and feature extraction methods that you used in your system. Also report the parameters used within.
We train our data with Support Vector Machine (SVM)
Support Vector Machine(SVM) performs classification by determining the hyperplane between two classes, we are use it with parameter
`svm.setType(cv2.ml.SVM_C_SVC)`, so it classifies it using One vs. One mode.
Parameters of kernel is:

```
svm.setKernel(cv2.ml.SVM_RBF)
svm.setC(60.0)
svm.setGamma(0.0005)
```

Classifier data saved in file:

```
svm.save('svm_data.dat')
```

Descriptor dimension (`desc_dim`) - is a variable which represents max number of features in feature vector's descriptor

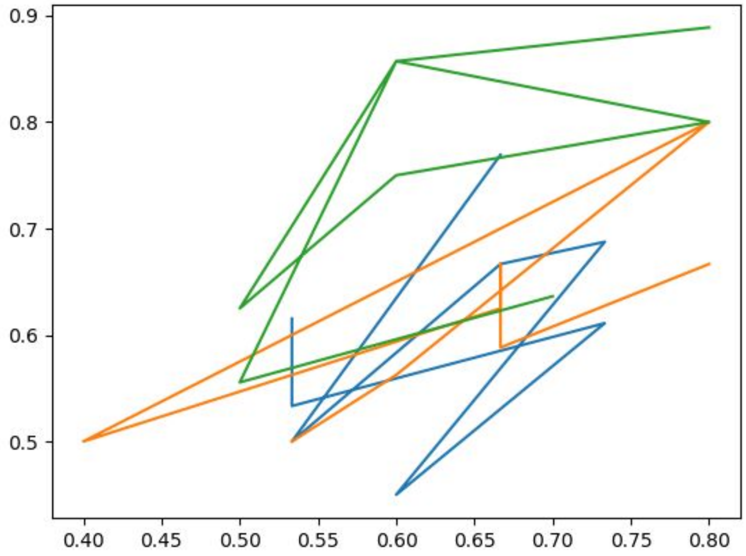
Dictionary size (`dict_size`) - is a variable which represents number of histogram's words

We take 15% as test set and 85% as training set

b) Report the performance as described in Steps 6a-c described above.

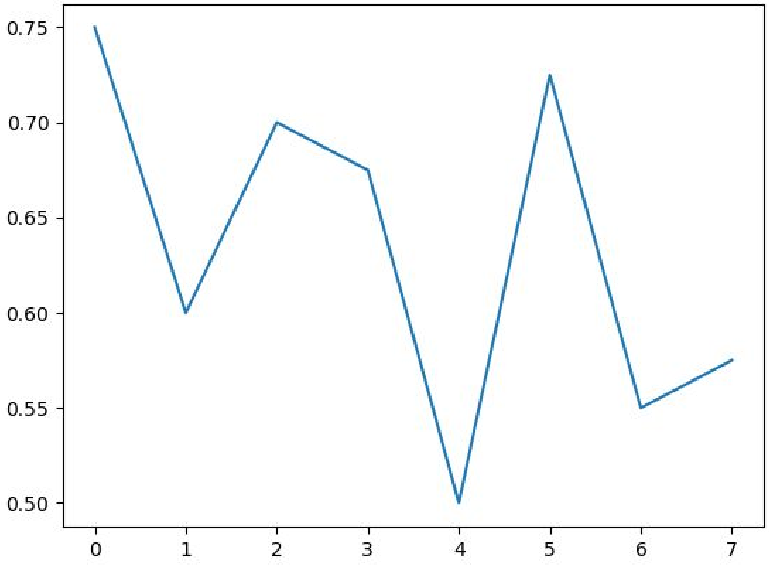
A :1) Plot ROC curve as a function of Descriptor dimension:

Dictionary size chosen as 52
Range of Descriptor dimensions chosen [116:124] shows



X-axis: Recall
Y-axis: Precision

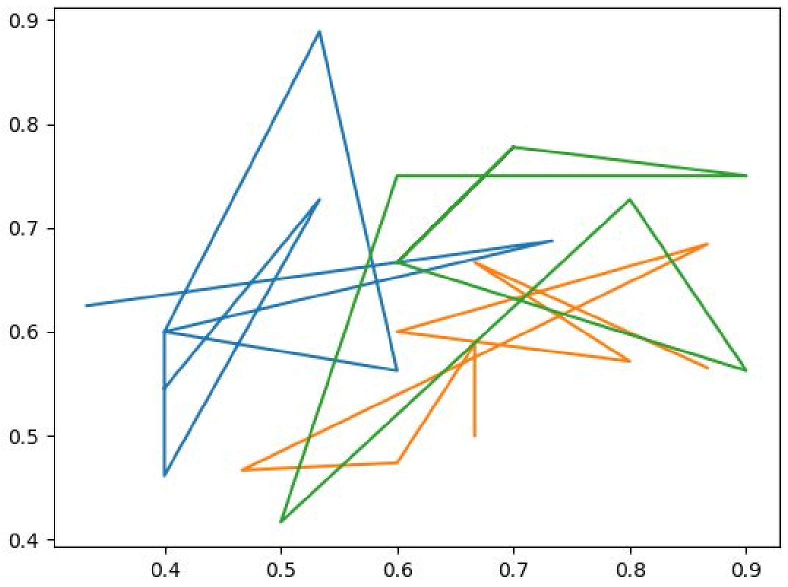
2) Plot Accuracy as a function of Descriptor dimension:



X-axis: iterations in [116:124]
Y-axis: Accuracy

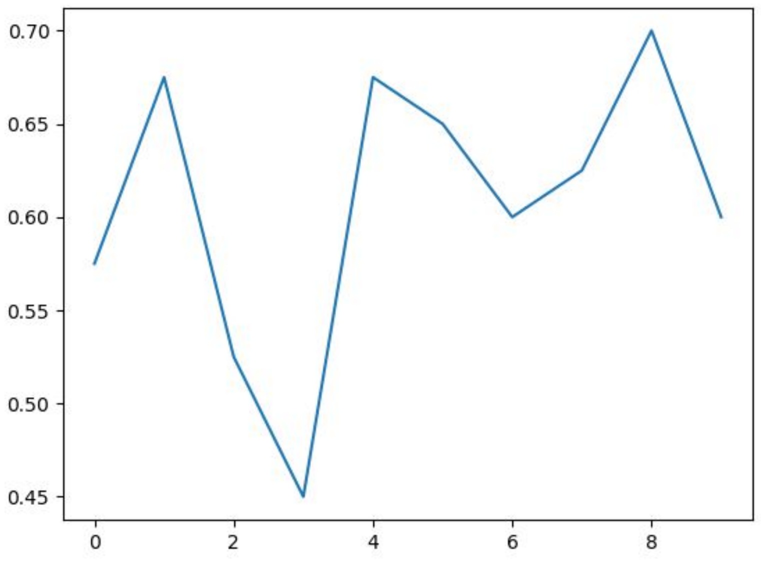
3) Optimal Descriptor dimension: 116

B :1) Plot ROC curve as a function of Dictionary size:



X-axis: Recall
Y-axis: Precision

2) Plot Accuracy as a function of Dictionary size:



X-axis: iterations in [52:64]
Y-axis: Accuracy

3) Optimal Dictionary size: 58

C :1) Precision, Recall, Accuracy for all images

All test best scores:

Accuracy: 0.7

Precision = 0.7

Recall = 0.72451923

2) Precision, Recall, Accuracy for each class (test images of each class).

Aiplanes scores:

Accuracy: 0.2

Precision = 0.53333333

Recall = 0.61538462

Motobikes scores:

Accuracy: 0.325

Precision = 0.86666667

Recall = 0.65

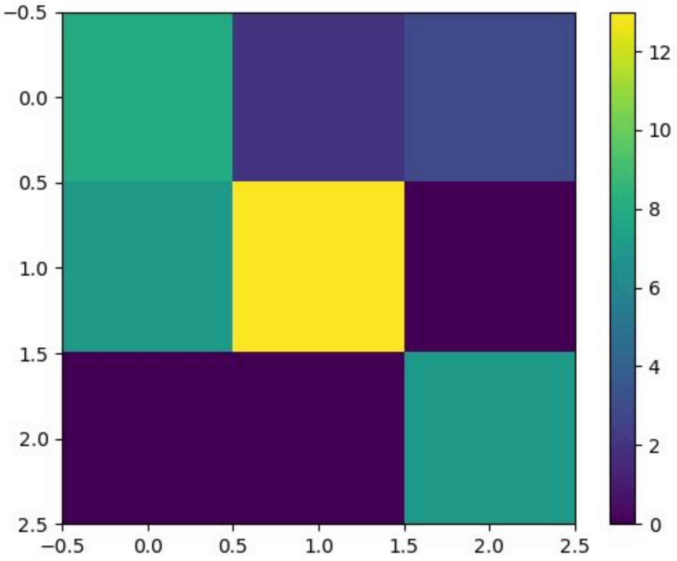
Elephants scores:

Accuracy: 0.175

Precision = 0.7

Recall = 1

3) Plot the confusion matrix

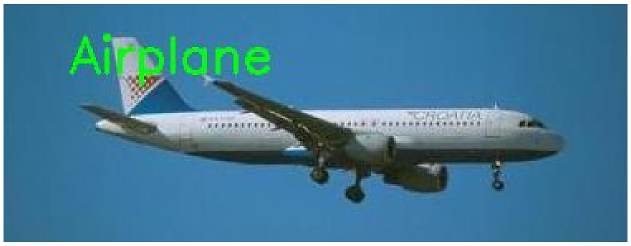


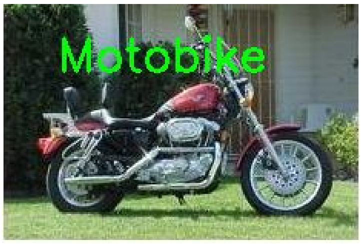
X-axis: Predicted class

Y-axis: True class

c) Include example images success and failures (Step 9d above).

Image classified correctly:





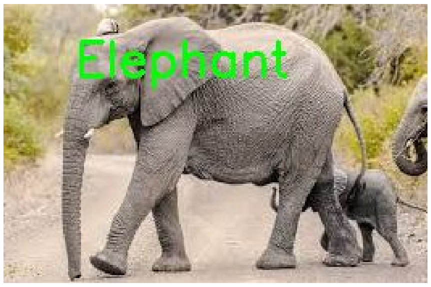
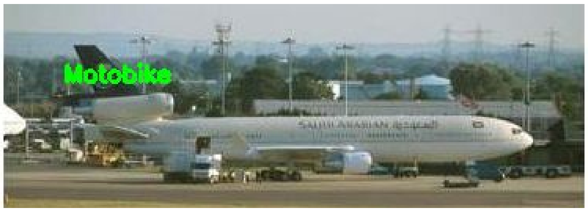
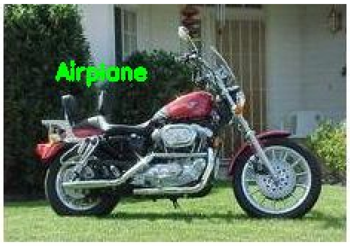


Image classified incorrectly:

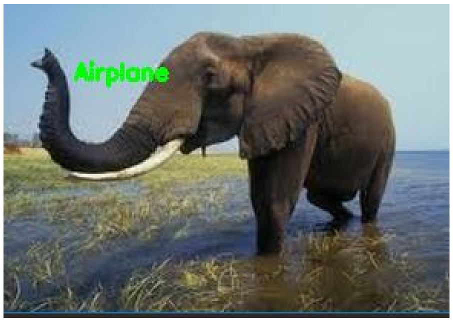
Motobike FP & Airplane FN:



Motobike FN & Airplane FP:



Elephant FN & Airplane FP:

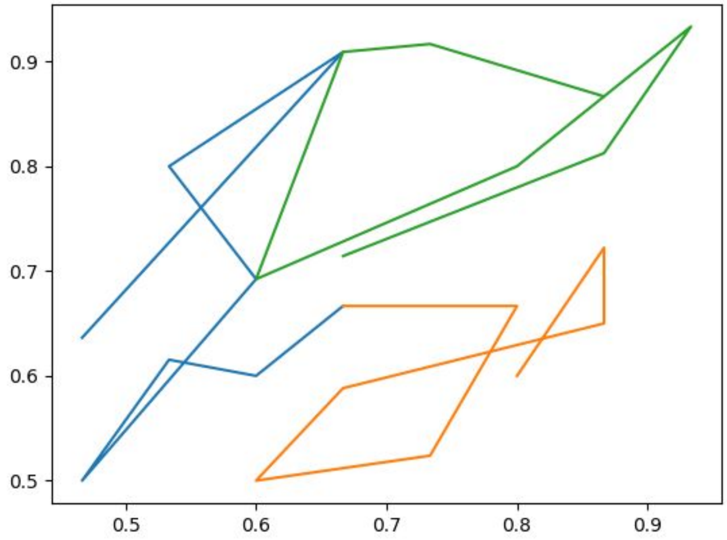


d) Explain the improvement to the system (Step 9e) and why you chose this.

We show that the data set is unbalanced (100,100,70) and saw that classifier confused in predictions of class Elephant so we add 30 additional samples to Elephant directory Now we are need lees bias `svm.setC(5.0)`

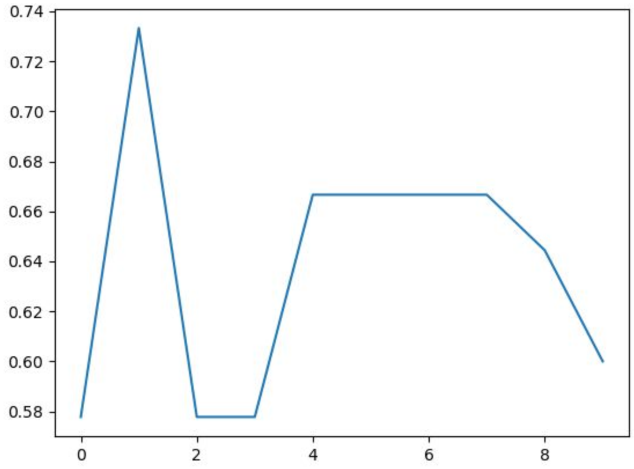
e) Report the performance of the improved system (steps 6a-c on improved system).

A :1) Plot ROC curve as a function of Descriptor dimension:



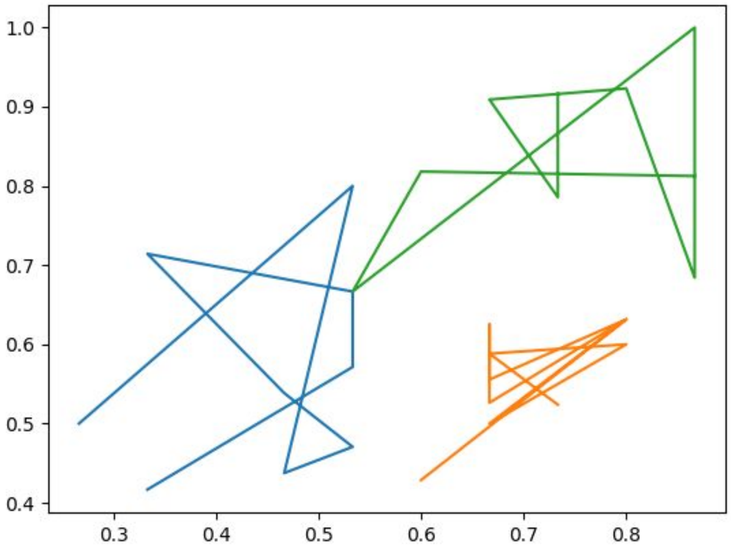
X-axis: Recall
Y-axis: Precision

2) Plot Accuracy as a function of Descriptor dimension:



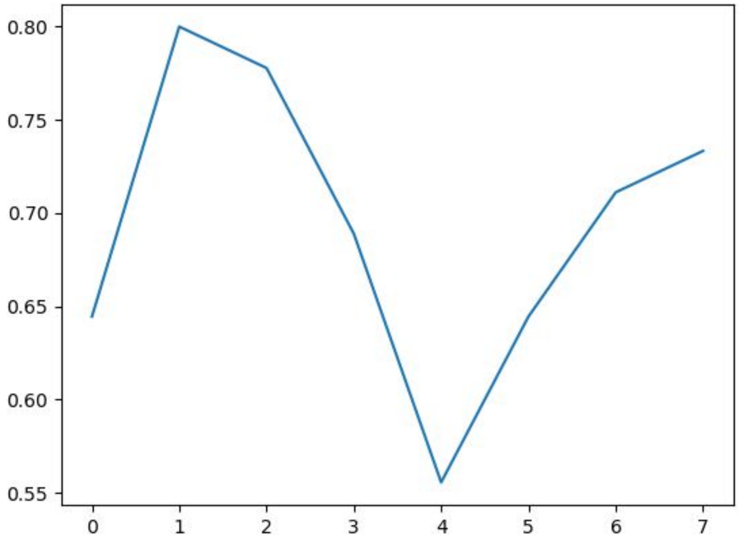
3) Optimal Descriptor dimension: 117

B :1) Plot ROC curve as a function of Dictionary size:



X-axis: Recall
Y-axis: Precision

2) Plot Accuracy as a function of Dictionary size:



3) Optimal Dictionary size: 58

C :1) Precision, Recall, Accuracy for all images:

<u>All test best scores:</u>

Accuracy: 0.7555

Precision = 0.7555

Recall = 0.7625

2) Precision, Recall, Accuracy for each class (test images of each class).

<u>Aiplanes scores:</u>

Accuracy: 0.2

Precision = 0.6

Recall = 0.6923

<u>Motobikes scores:</u>

Accuracy: 0.2666

Precision = 0.8

Recall = 0.66666667

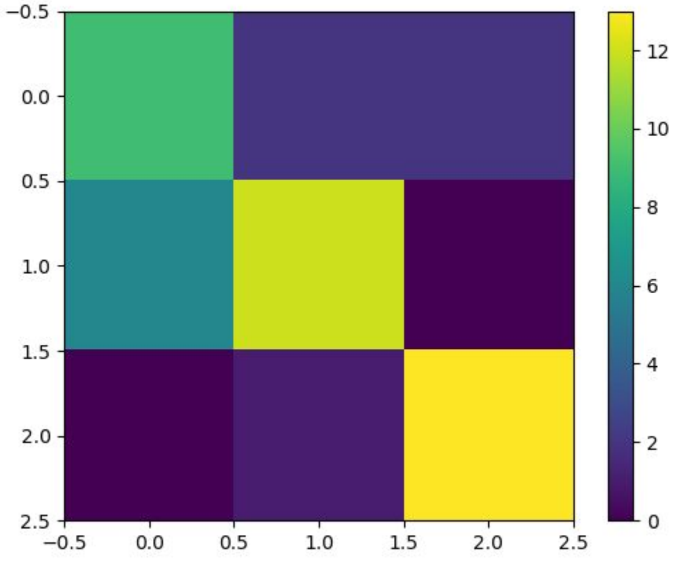
<u>Elefants scores:</u>

Accuracy: 0.28888888888888886

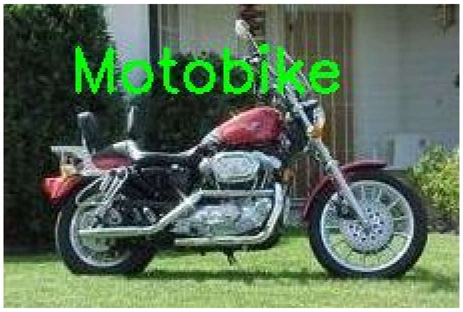
Precision =[0.86666667

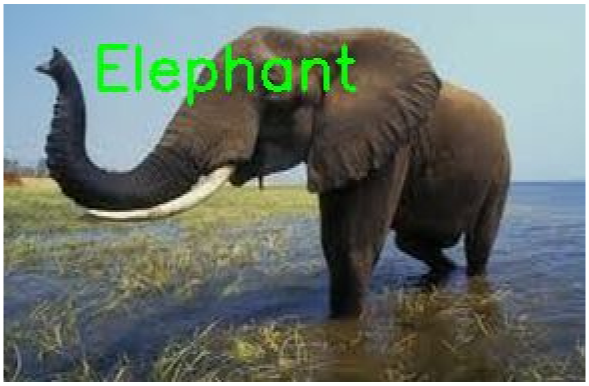
Recall = 0.92857143

3) Plot the confusion matrix:

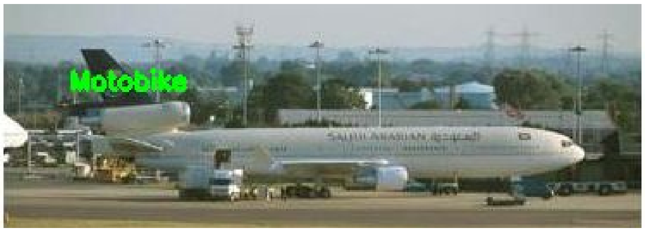


f) Show examples that failed before and succeed on the improved system.Image failed before And then succeed on the improved system:





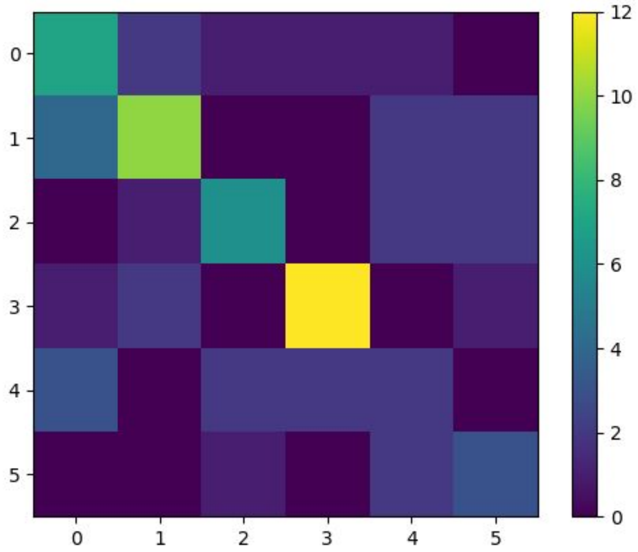
Show examples that still fail here.



g) What happens to accuracy of your system when more classes are added?
The Dataset contains 3 additional classes: Chair, Wheelchair, Ferry.
Adding the classes one by one,
plot the accuracy resulting from testing on the same test data as in Step 6 c testing(the test data in the 3 classes: Airplane, Elephant and Motorbike).

```
100 images were loaded from Datasets/Airplane/
100 images were loaded from Datasets/Motorbike/
70 images were loaded from Datasets/Elephant/
100 images were loaded from Datasets/Chair/
57 images were loaded from Datasets/Ferry/
50 images were loaded from Datasets/Wheelchair/
Train set len of Airplane is 85
Train set len of Motorbike is 85
Train set len of Elephant is 60
Train set len of Chair is 85
Train set len of Ferry is 48
Train set len of Wheelchair is 42
Test set len of Airplane is 15
Test set len of Motorbike is 15
Test set len of Elephant is 10
Test set len of Chair is 15
Test set len of Ferry is 9
Test set len of Wheelchair is 8
Accuracy is: 0.5555555555555556
All test best scores:
Accuracy: 0.5555555555555556
Precision = [0.67777778],
Recall = [0.63878367]
Airplane scores: Accuracy: 0.09722222222222222
 Precision = [0.46666667], Recall = [0.58333333]
Motorbike scores: Accuracy: 0.1388888888888889
 Precision = [0.66666667], Recall = [0.55555556]
Elephant scores: Accuracy: 0.08333333333333333
 Precision = [0.6], Recall = [0.54545455]
Chair scores: Accuracy: 0.16666666666666666
 Precision = [0.6], Recall = [0.54545455]
Ferry scores: Accuracy: 0.027777777777777776
 Precision = [0.6], Recall = [0.54545455]
Wheelchair scores: Accuracy: 0.041666666666666664
 Precision = [0.6], Recall = [0.54545455]
```

Confusion matrix:



X-axis: Predicted class
Y-axis: True class

We can see since we have too few samples of Farry and Wheelchairs it classified poorly.