SCENE SEGMENTATION AND INTERPRETATION

PROJECT REPORT

SUBMITTED BY **CHALIKONDA PRABHU KUMAR (CHAKON)**

STUDENT ID: 1925158

PASCAL PROJECT

Universitat de Girona

Masters in Computer Vision

UB Université de Bourgogne

CONTENTS

1. INTRODUCTION:

Image classification is the important task in the computer vision. There exists some complexity in terms of robust unique pattern recognition techniques when considering efficient and speed. There are many societies and organizations like European Union focused on solving this problem. Merging of the different networks of institutes with its corresponding researchers to merge their ideas and jointly trying to find the solution for this problem. They named a project titled as PASCAL (Patter Analysis, Statistical Modeling and Computational Learning).

Considering the work on this project from past many years new techniques are implemented to find the solution. This is the unique task especially when considering the possible interference of other classes, occlusion, scale, pose changes etc in the different images. The techniques that are implemented every year on this project aims to achieve high accuracy and speed.

In this report a clear details about the task given in the project is presented, with some experiments on extracting features, classification and followed by conclusion.

2. OVERVIEW:

The main objective of this project is to design a classifier which is able to predict whether a given class is presented in an input image. In this we have given the number of selected classed is 10 they are bicycle, bus, car, cat, cow, dog, horse, motorbike, person, sheep etc.

The data base is composed of the images from many sources with various levels of complexity. The data base is less when compared to the actual PASCAL challenge data base with less number of classes and number of images.

In this I implemented the sift descriptors obviously which are good and clustered them by means of the K-means create bag of words (BOW). This dictionary/vocabulary (bag of words) was used to create image descriptors, by making the normalization of the histogram model of the image from it's sift feature descriptors. Classifier based techniques like SVM (Support Vector Machine), Knn (K-Nearest Neighborhood), adaboost classifiers are used to learning and creating a final individual classifier.

The accuracy of the system will be measured from the ROC curve and its corresponding average AUC values.

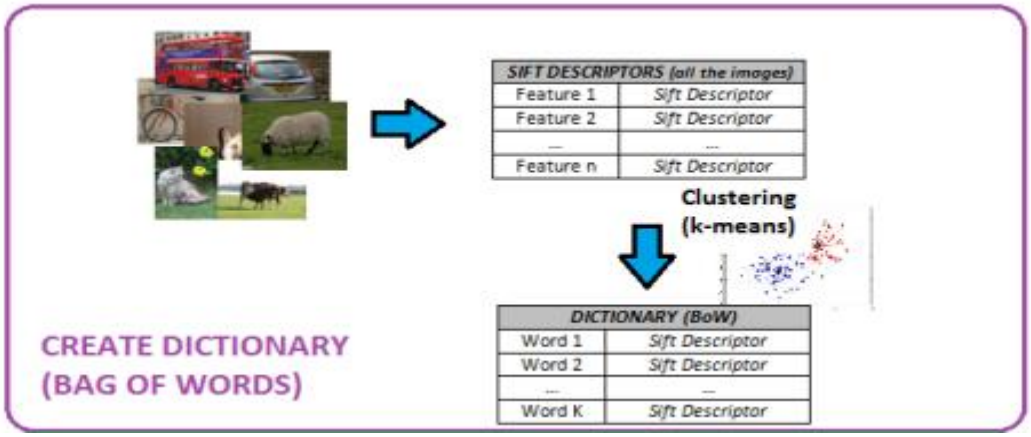
3. DESIGN AND IMPLEMENTATION:

For designing the code I use PR tools and VL feat latest version for sift descriptors, PR tools for classifying.
In the below section I am going to explain clear details of the strategy I implemented for doing the project.
The flow chart gives overview about the strategy for implementing the PASCAL project.

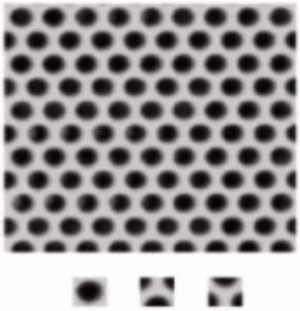
The initial step is creating the bag of words (BOW) which will be used to describe all the classes. By using this bag of words, we match all the features of a given set of training images of certain class and we built the histogram where in each bin we have how many features belonging to a specific word within the bag of words/dictionary. Once getting the bag of words a classifier for this class is defined using these histograms.
Next step is training the data set of the images with the bag of words. Once training is done computing the confidence level that a given test image belongs to a given class or not. To do this we match all the features of the test image to words of the dictionary and we compute the histogram. Using the histogram and the created classifier, we will get the desired confidence level.

A. Bag Of Words:

The initial step consists of creating a bag of words that will describe the characteristics of each image. In computer vision, this is well known as Bag of Words (BoW). First we should extract the different features of the images from different classes with different combinations of training images and describing the each combination individually.



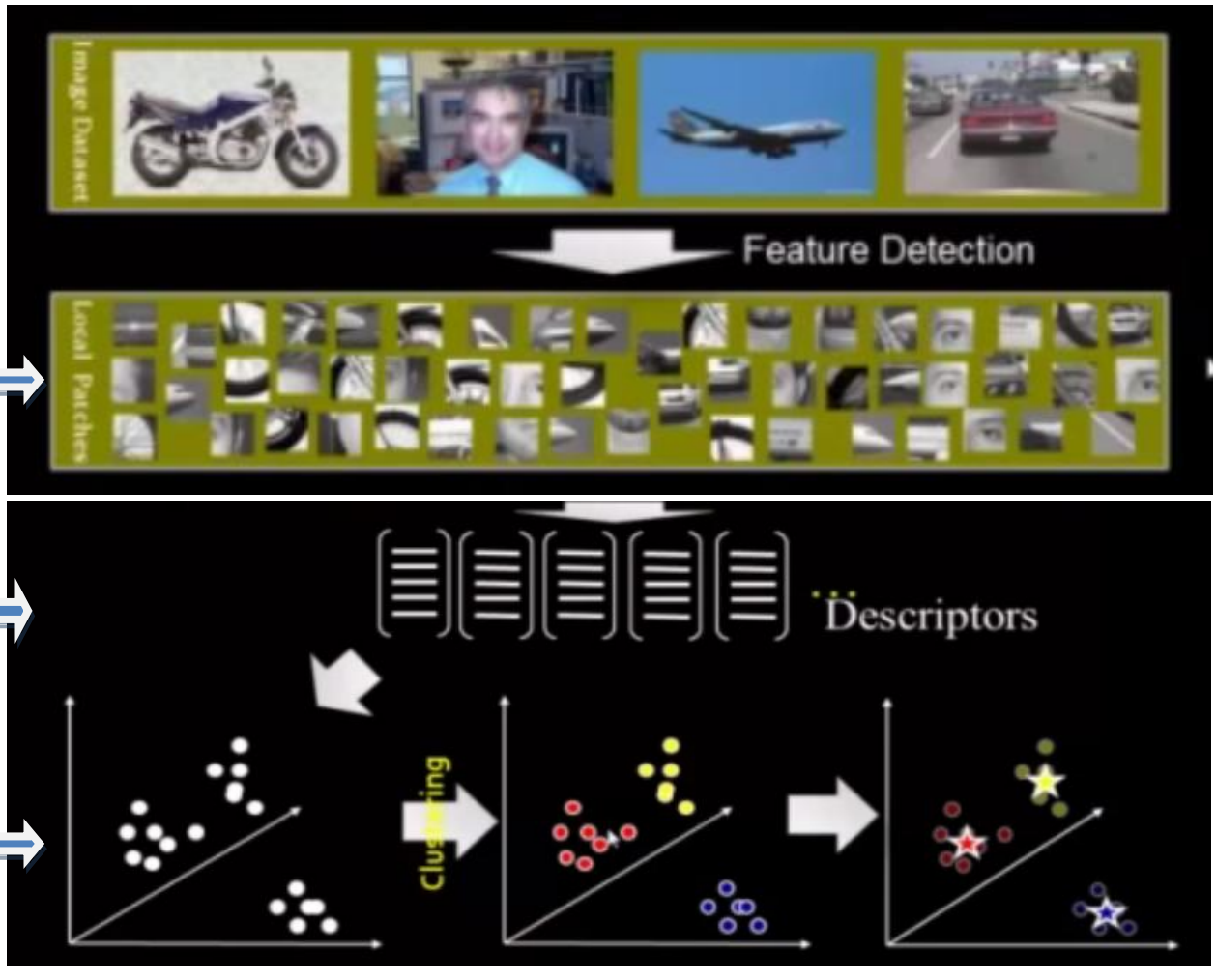
The above image gives idea how the bag of words is done.



The idea of this Bag of Words features idea come from the texture recognition. How Bag of Words inspired from texture recognition concepts. Let's see that in brief. Texture is characterized by the repetition of basic elements or textons. From the image we can primitives of the image is repeated in maximum of the entire image. We see the image in terms of these primitives.

Then we will represent this image in histogram with texture elements/ textons. By doing there is reduction in dimensionality so that we have vector which represent the image.

The same procedure is applicable for creating the bag of words. First we will represent the image with the set of patches using the Harries interest point detector or any method. Once the patches are obtained we compute the descriptor for each of the patch these are nothing but vectors. In this case I used very important descriptor method SIFT (Scale invariant Features Transform). Once all the descriptors are added to a database, the next step includes: projecting them into a 128 dimension space and apply clustering in order to find the appropriate mean of the clusters in the feature space. As I am using the SIFT descriptors it contains 128 values. The methods I applied for clustering us K-means, here k will be the number of words that we want to have in order to define the dictionary/bag of words. Also different number of clusters I used in order to obtain the better results. The below image show how the bag of words is done in this project gives clear details.



HARRIS Interest Point detectors/ Sift etc can be used

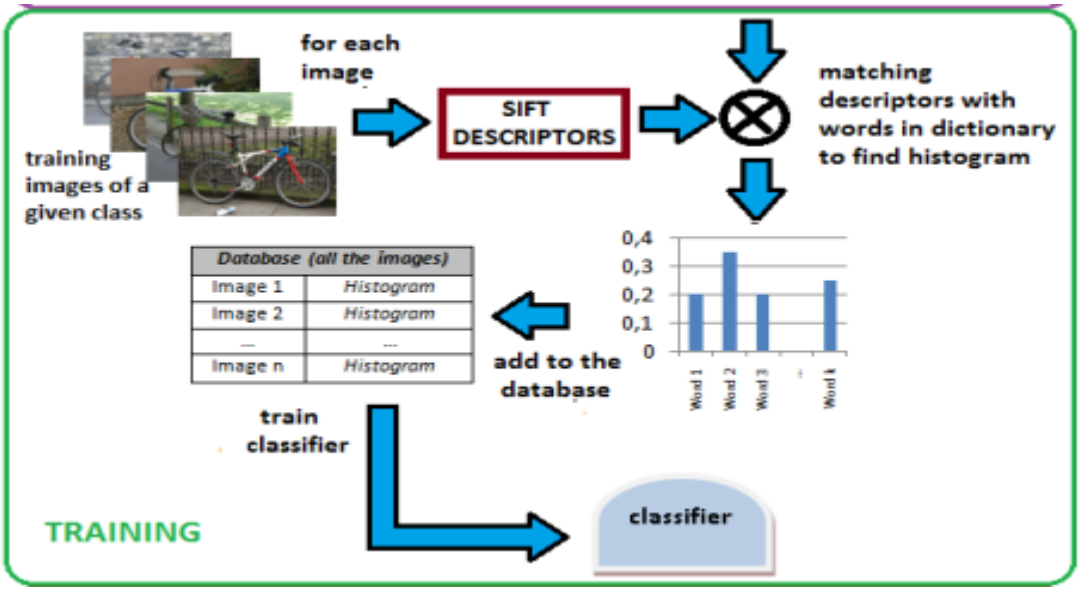
SIFT/ SURF, Gaussian mixture, GLCM etc can be used

K- Means clustering method is used.

By doing this steps we will get the Bag of Words with specified number words and its corresponding feature descriptor is computed.

B. Training :

Once the Bag of words (BoW) is created, the next step consists in building the classifiers for each class that will allow us to determine the test image belong to a class or not. This is done for each class (i.e. 10 in our case) in the data set. As we know each class contains a set of training images which contain positive images (belong to the class), negative images (not belong to the class) and difficult images (belongs to the class but is really difficult to detect). Using this set of training images, we compute for each of them the SIFT descriptors and we match each of these descriptors with the closest word of the dictionary (computing the closest Euclidean distance).



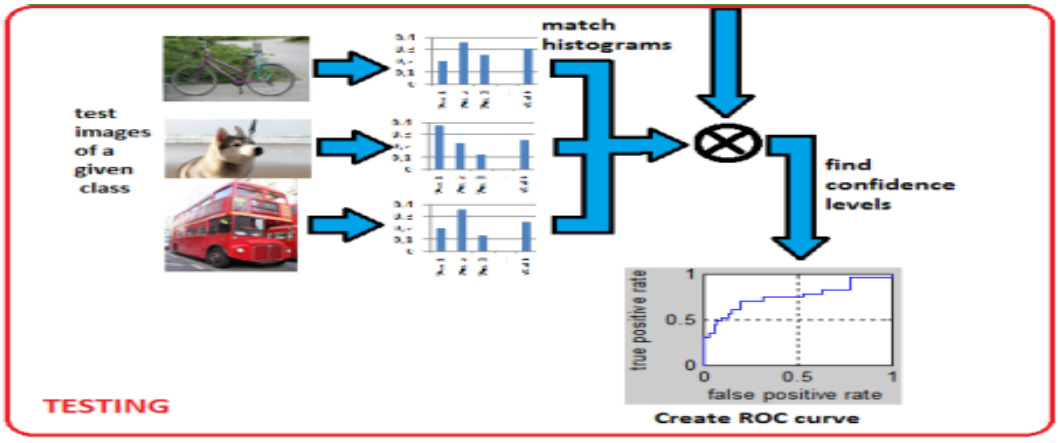
Once all the descriptors are matched, we built a histogram for each image where in each bin we will have the number of features that are matched according to the distribution of the words in the dictionary. The histogram should be normalized because it is possible that different images will contain different images. All of these histograms are added into a database where we also store the ground truth representing the image. This means that for each histogram we know whether the image belongs to the class or

not. Once this database is done we should proceed with the classifier for seeing the results using the different classifiers were used and results are noted. I used Support Vector Machine and Adaboost for classification part.

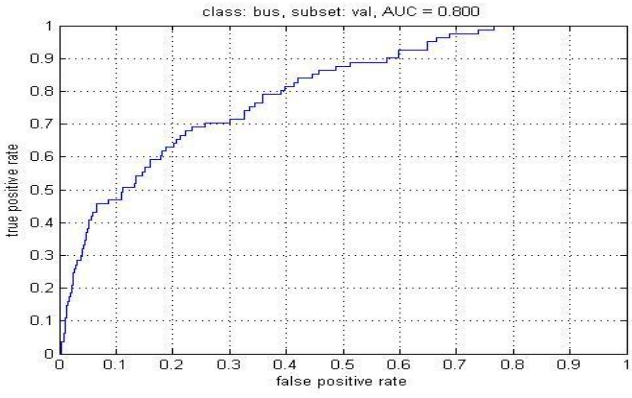
The brief details of SIFT and classifiers are explained later in the section.

C. TESTING:

Final step consist of testing how efficient our method by using a set of test images. These sets are also specific for each of the classes and contain different images that can belong or not belong to the class.



The goal consists in finding for each one of the images the confidence level that the images have to belong to the evaluated class in order to find the final ROC curve for the specific class.



To do this for each test image we compute the SIFT descriptors of the features and we match all the features with the closest word in the bag. Then the histogram is computed in the same way as was done in the training step and finally we use the classifier in order to predict the confidence level of this histogram to see whether belong to current class or not.

ALGORITHMS USED FOR PROJECT:

a. FEATURE DESCRIPTOR:

In order to detect the features from one image and then describe them, we have tried to use two different algorithms SIFT and SURF.

Scale invariant features transform (SIFT) is an algorithm in computer vision to detect and describe local features in images which are robust enough to be used as the solution to well known correspondence problem. The algorithm was published by David Lowe in 1999. Other applications include object recognition, robot mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking and match moving.

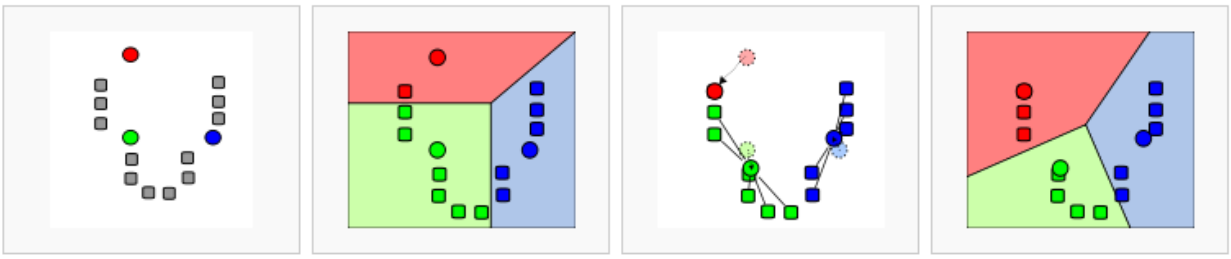
SIFT features are local and based on the appearance of the object irrespective of the size, view point, illumination, rotation etc. In addition to these properties, they are highly distinctive and relatively easy to extract which allows users to correctly do object identification with low probability of mismatch and are easy to match against a large database of local features. Apart from this the 128 dimensional information of the feature makes it quiet unique and large to be used for classification type problem.

Apart from SIFT descriptors, SURF is also widely used for feature extraction. SURF (Speeded up robust feature) designed by Dr. Herbert Bay which is almost similar to SIFT. But SURF us much faster than SIFT, but it turns out this depends on the implementation skill of the programmer. Apart from its speed the reason SURF was not finally selected was the limited dimensional vector for each feature.

b. CLUSTERING :

K-means clustering algorithm which can cluster n observations into k partitions based on the distance from the nearest mean. I am using the K means in order to reduce a feature space of 128 dimensions into a feature space of points with a number of dimensions equal to the number of words

in the bag of words. The demonstration of the standard algorithm is shown below.



1) *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).

2) *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3) The centroid of each of the *k* clusters becomes the new mean.

4) Steps 2 and 3 are repeated until convergence has been reached.

c. CLASSIFICATION:

1. K- Nearest Neighborhood:

In pattern recognition the k-nearest neighbor is regarded to be the simplest of all the machine learning algorithms. This is a method for clustering objects based on the training examples in the feature space. If k=1 then the object is simply assigned to the most common amongst neighbor puts more weight to the contributors of its closest match.

2. ADABOOST:

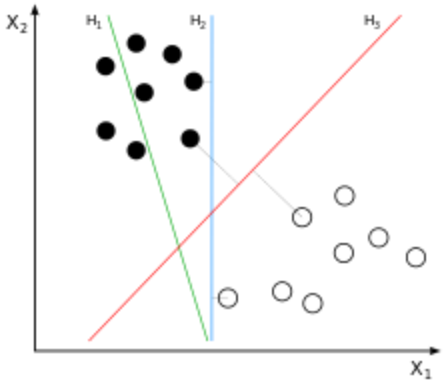
Adaboost algorithm is boosting technique which assumes at the initiation that given a weak learning classifier which slightly better than random, can be boosted to find the perfect prediction classifier. This is done iteratively by emphasizing more importance on the wrong classification done by the previous weak learner and tries to tend towards the more appropriate global classifier.

3. SUPPORT VECTOR MACHINE:

This algorithm is considered as a linear classifier. The basic idea consist of given set of points that belong to subspace where these points belong to once class over two categories an SVM algorithm will predict id a new points belongs to one class or another. To do so, it finds the hyper plane that best splits on class from the other.

The best hyper plane will be the one that maximizes the distance with the points that are closer to it. Then, all the points that are in one site of the hyper plane belong to one class and the ones in the other side belong to the other class.

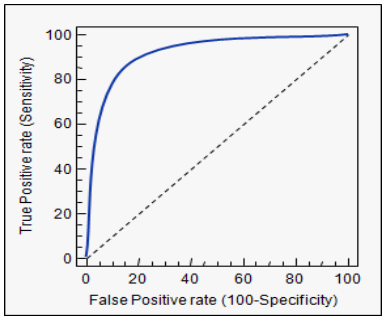
There are infinite hyper planes that can have the maximum margin among the data of the classes. The binary case of the SVM is shown below.



The data also contains more than two classes to solve this, the SVM is

- Divide each category in new categories and combine them.
- Build k(k-1)/2 models where k is the number of categories.

4. ROC CURVE:



The ROC curve is a used to know how classifier is good. The Y-axis of the curve gives the rate of the classifier returning a positive image (TRUE positive) when it's a positive while the X-axis shows the rate of the classifier returning an image (FALSE Positive) suggesting positive where as in reality it's not. So, intuitively we want the classifier to return as much as True positive compared to false positive which would tend to make the area under the curve.

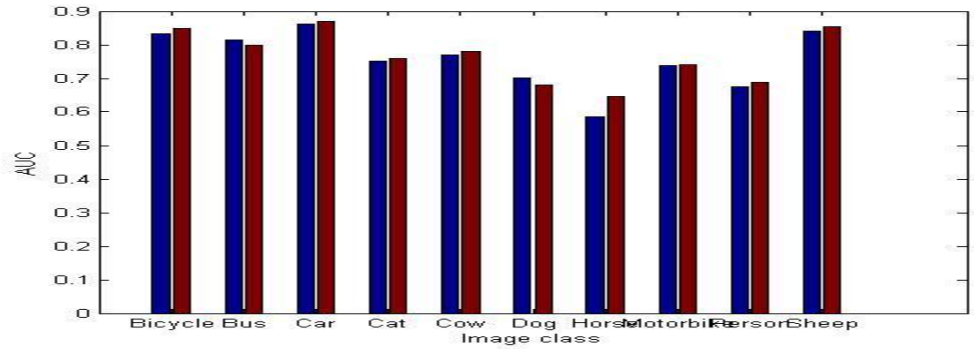
4. RESULTS AND EXPERIMENTS:

In this I run the code with the change in image class and number of clusters. The results are obtained based in the strategy we explained. In this case I used SIFT descriptors for extracting features and used SVM as the classifier. The results obtained with the change in classes and number of clusters is shown below.

Image class	Number of Clusters	Bicycle	Bus	Car	Cat	Cow	Dog	Horse	Motorbike	Person	Sheep
5	50	0.815	0.00	0.810	0.745	0.00	0.705	0.618	0.00	0.00	0.812
10	50	0.808	0.00	0.833	0.729	0.00	0.00	0.00	0.700	0.00	0.827
5	100	0.821	0.817	0.845	0.744	0.00	0.670	0.638	0.590	0.659	0.839
10	100	0.845	0.830	0.838	0.757	0.00	0.00	0.00	0.00	0.674	0.841

From the above results it is evident that the number classes and number of clusters plays an important role. As we increase the number of clusters bag words also increase due to this there is a high probability of increase in the classification results. The above table is summed with few points some classes ROC curve are zeros this is because of number of classes we choose and clusters used in the project. With it is evident that as we increase the number of clusters we can achieve better results so I draw the results for different clusters and keeping the image classes at 10.

Image class	Number of Clusters	Bicycle	Bus	Car	Cat	Cow	Dog	Horse	Motorbike	Person	Sheep
10	200	0.834	0.815	0.861	0.752	0.770	0.701	0.585	0.738	0.676	0.842
10	400	0.850	0.800	0.871	0.760	0.779	0.681	0.646	0.741	0.687	0.855

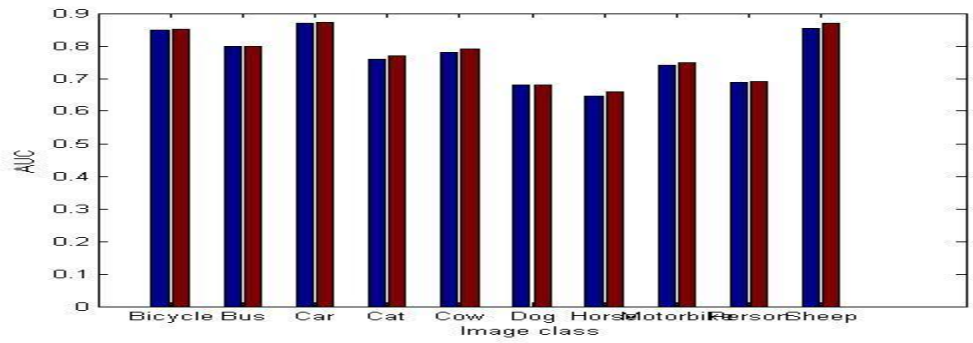


The plot above shows how the AUC of ROC curve increases blue bars are for 200 clusters with 10 image classes and red are for 400 clusters with 10 image classes. From the plot it is clearly visible that number of cluster increases we can increase size of bag of words which in turn improves the results

EXPERIMENTS:

In this I used adaboost instead of SVM in training step for this classifier the results are almost same as SVM. But there is slight difference in the AUC.

Image class	Number of Clusters	Bicycle	Bus	Car	Cat	Cow	Dog	Horse	Motorbike	Person	Sheep
10	400 (SVM)	0.850	0.800	0.871	0.760	0.779	0.681	0.646	0.741	0.687	0.855
10	400 (Adaboost)	0.851	0.799	0.873	0.77	0.79	0.68	0.66	0.75	0.69	0.87



The plot shows the SVM vs adaboost classifier. Blue is for SVM and Red bar represents for adaboost classifier. As from the above plot we can observe that both are almost equal with small change in AUC.

In the project tutorial sheet we were given 10 classes what happens if we change the number of classes more like 15 then AUC for ROC curve will increases.

Image class	**Number of Clusters**	**Bicycle**	**Bus**	**Car**	**Cat**	**Cow**	**Dog**	**Horse**	**Motorbike**	**Person**	**Sheep**
10	400 (SVM)	0.850	0.800	0.871	0.760	0.779	0.681	0.646	0.741	0.687	0.855
15	400 (SVM)	0.836	0.819	0.875	0.760	0.793	0.685	0.626	0.75	0.69	0.861

From the above results it is evident that for most of the classes we can see there is small increment in the area under curve.

By doing this we can say that AUC depends in the bag of words, more features that are used. SURF is also more like SIFT but its fast and robust in sift it has length 128 is high when compare to SURF. The most important thing is SURF is computationally fast when compare with SIFT.

5. DISCUSSION AND PROBLEMS:

- Small issues in the coding are solved in the lab.
- More time is consuming to get the results.
- Being only one I manage to do project but unable to explore more on features. Did experiment with increasing the bag of words, image classes etc

6. FUTURE WORKS:

- To investigate more on the features like color sift in a different color space (like HSV), Gaussian mixture features and will try to improve the Area under curve.
- To work on the classifiers to get the better results.

7. ORGANIZING WORK:

S.No	DATE	Work
1	14/4/2013	Look on to the given data
2	20/4/2013	Followed tutorial for bag of words
3	22/4/2013	Coded for bag of words
4	23/4/2013	Questions are cleared with Arnau
5	24/4/2013	Coded for Bag of Words
6	28/4/2013	Meet Lab guide Arnau for minimizing errors and done with training
7	29/4/2013	Got results and taking advice to write the report
8	30/4/2013	Started report and done with experiments
9	31/4/2013	Some experiments and completed report

8. CONCLUSION:

In this project as the number of features increase we can get the better result. SIFT and SURF is the good descriptors which will give better results. Apart from this we can use LBP pattern, color sift (like HSV), dense sift etc features to the database so that we can get good result. But sometimes SIFT doesn't work good for some classes because as i went through the database images we saw that, some of them are highly occluded some of them are too far away (in scale). Apart from being far away they also have images of car with horses in front that are described as positive image. When sift features are extracted to make the classifier for a car the SIFT features focuses more on the horse than the car, which makes classifying classes even more difficult. In order to overcome this issue some regions of interest should be extracted from each class image to make the bag of words more distinctive.

9. REFERENCES:

- [Youtube video for Bag of Words] https://www.youtube.com/watch?v=iGZpJZhqEME
- [wiki pages about SIFT] http://en.wikipedia.org/wiki/Scale-invariant_feature_transform
- [wiki pages about ROC] http://en.wikipedia.org/wiki/Receiver_operating_characteristic
- [Pascal Project] http://www.pascal-network.org/
- [VL_feat library] http://www.vlfeat.org/
- [PrTools] http://www.prtools.org/
- David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110