5. THE PROPOSED CLASSIFIERS – *HORBoVF* AND *ACORBoVF*

This chapter starts with an introduction to the basics of image classification. It also discusses the bag of visual words classification. The second section contains a detailed description of the two classifiers: the *HORBoVF* and the A*CORBoVF* whereas the third section carries out a detailed performance analysis of these classifiers.

5.1 IMAGE CLASSIFICATION

5.1.1 Introduction to Image Classification

For human beings, identification of images is quite easy as human eyes can see and the brain can process all the aspects of an image to recognize it. However, the same is not easy for a computer. This process of recognizing an image based on some set of given images is called image recognition or identification, also classification. Many times, it is also called image labeling. To do the same, for a computer, a set of several steps must be carried out. Overall this process is called image classification but is divided into mainly three parts, preprocessing, feature extraction and classification. The following figure shows the overall image classification process.

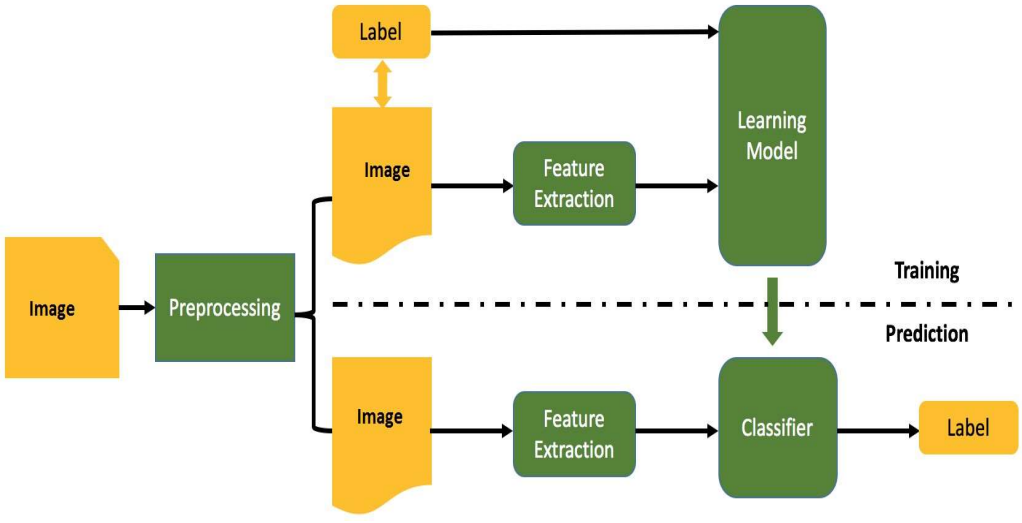

Figure 5.1. Image classification process

Upon feature extraction, these extracted features are used for further classification of the image. This classification is carried out using training of the system with supervised or unsupervised learning. The following figure shows the difference and various techniques of supervised and unsupervised learning.

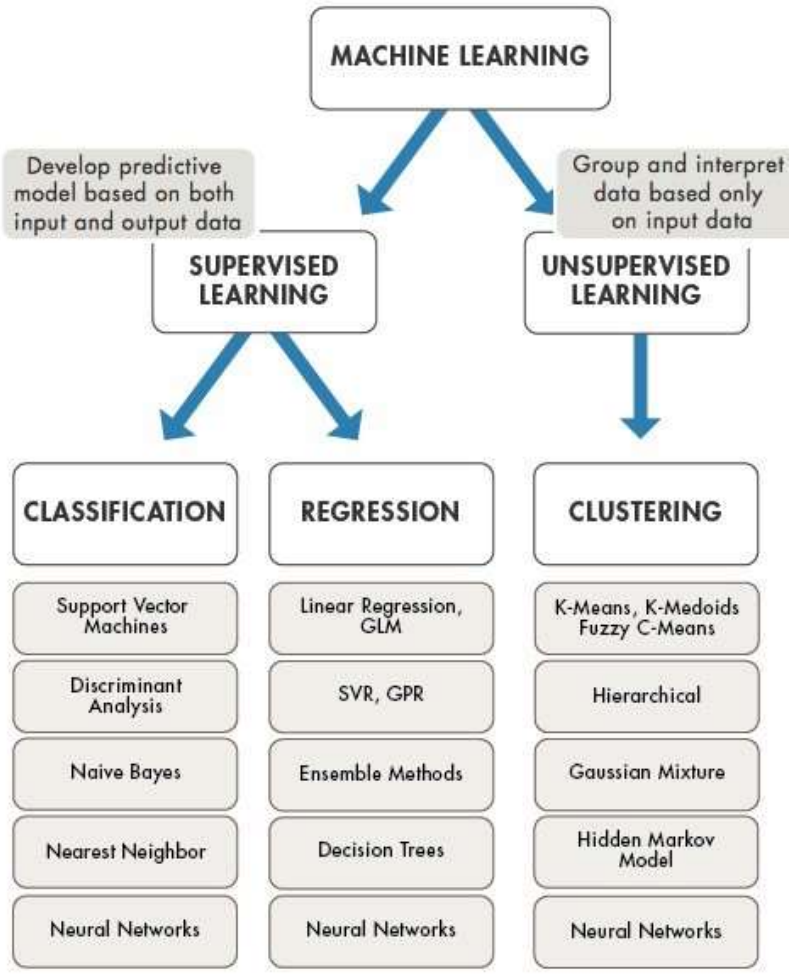

Figure 5.2. Supervised Vs. Unsupervised Learning

However, as discussed earlier, the image classification is not as simple as text classification for some challenging factors. Following is the list of the factors that affect image classification.

1. Viewpoint variation: The viewpoint of the image is an important aspect in image classification as it shows the angle and direction from which the image has been captured.

2. Illumination conditions: The lighting conditions at the time of capturing image make a lot of difference as the pixel colors, and their corresponding values may change which can affect the decision making.
3. Scale variation: Changing the size of the image or the same object also affects the classification process.
4. Deformation: If the object is sheared or deformed in shape the image identification becomes difficult.
5. Background clutter: This happens when the color of an object in the image and background in the image is almost of similar color or pattern. This is one of a most challenging factor in image classification.
6. Occlusion: When the object of interest is of very small in size concerning the overall size of the image, it is called occlusion.
7. Intra-class variations of the objects also affect the labeling of images.

Following image explains the discussed factors visually.

Figure 5.3. Factors that affect image classification process

The next subsection introduces a bag of visual words classification techniques which is most widely used for image classification. This has been used in the proposed approaches also.

5.1.2 Bag of Visual Words Classification

Bag of Visual Words is an extension to the NLP algorithm, Bag of Words, used for image classification. It is one of the famous concepts used in image classification except for Convolutional Neural Networks.

When some data are to be fetched from the considerably large database, say from millions of documents, it would be necessary to have a faster and quicker approach. One cannot compare every word from each document. That could be time-consuming and may lead to catastrophe. For this, a bag of words concepts were developed in analogy with a dictionary of words. Here, the bag is a collection of important words of every document. The irrelevant and less important words like “is,” “am,” “the,” “it” etc. are discarded.

In the same line of concepts, for images, a bag of visual words is created. Visual words are important points in the image, called features. Using these features, a large vocabulary is created representing each image as a histogram of the frequency words (features) in the image. To use this model, as stated earlier, firstly, it is needed to extract the features and generate vocabulary using clustering, then, extract the feature descriptor using frequency analysis, and finally, label the image. Following image shows the steps to of generating visual vocabulary and the overall flow of classification using a bag of visual words model.

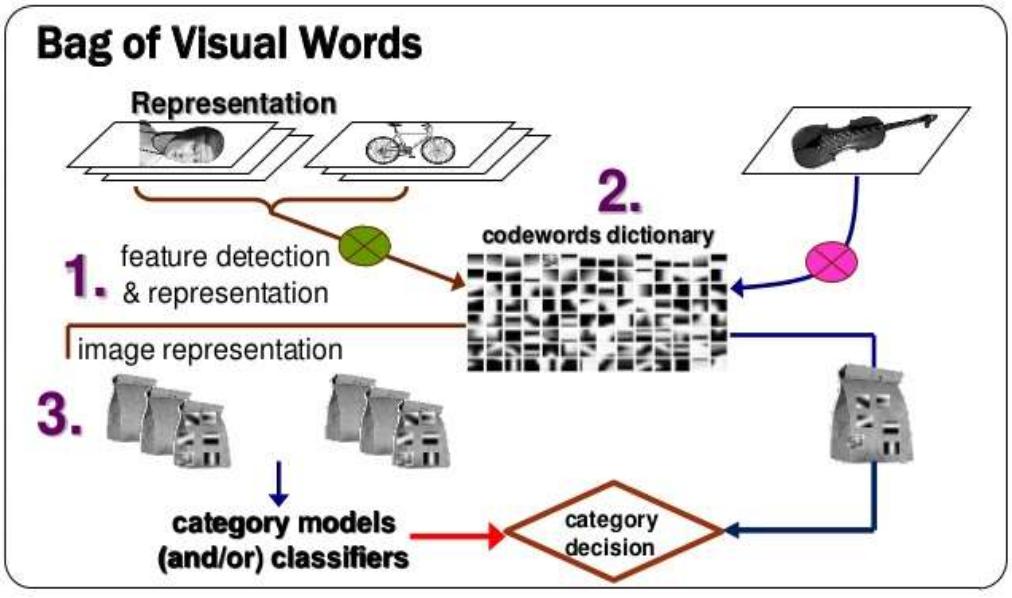

Figure 5.4. Classification process of visual words model

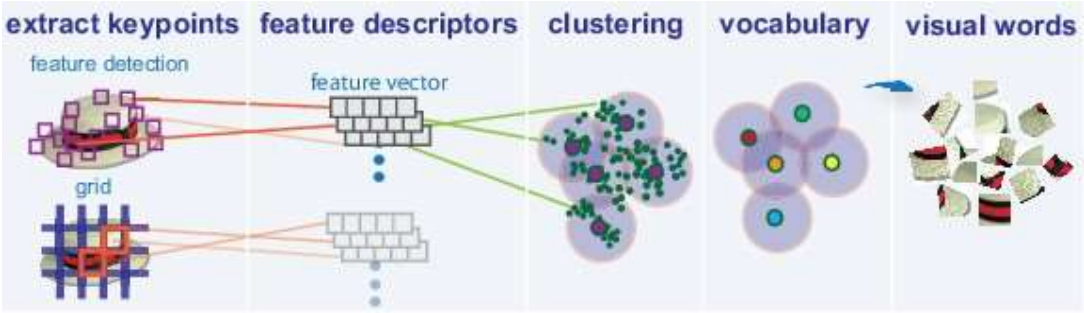

Figure 5.5. Generating visual vocabulary

Here, K-Means clustering is used for vocabulary creation. Suppose, there are X objects that are to be divided into K clusters. The input can be a set of features, $X=\{x_1,x_2,\ldots,x_n\}$. The aim is to lower the distance between each point in the scattered cloud and the assigned centroids.

$$\underset{s}{arg\,min} \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2$$

Where, μ is the mean of points for each S_i(cluster) and S denotes a set of points partitioned into clusters of $\{S_1, S_2,\ldots S_i\}$

Hence, it can be inferred that for each cluster centroid, there exists a group of the point around it, known as the center. The steps are given below:

1. It starts with an initial random solution. It can be called centroids of the clusters. These centroids need to be randomly placed within the bounds of data.
2. In the next step, K-Means iterates over each of the input features and decides which feature is the closest to the cluster centroid w.r.t itself. Once the closest centroid is found, the feature is put into that cluster of the found centroid.
3. In this third step, the reallocation of cluster centroid is done. The new cluster centroid is calculated based on the aggregate of all members of that particular cluster. In this, the cluster centroid moves either outwards for loosely coupled features or inwards for tightly aligned features. This reallocation process continues until the cluster centroid does not change concerning its previous position. This can be done through some threshold values too.

K-Means is one of the most widely used clustering approaches in unsupervised learning. Bag of visual words uses a training model in which similar features are partitioned that will be used in extrapolation to make a decision. It is these features which are clustered in the form of vocabulary that will help in labeling of images using SVM, SKLEARN, VLFEAT or any other machine learning toolkit. The following figure shows a visualization of a bag of visual words model after clustering.

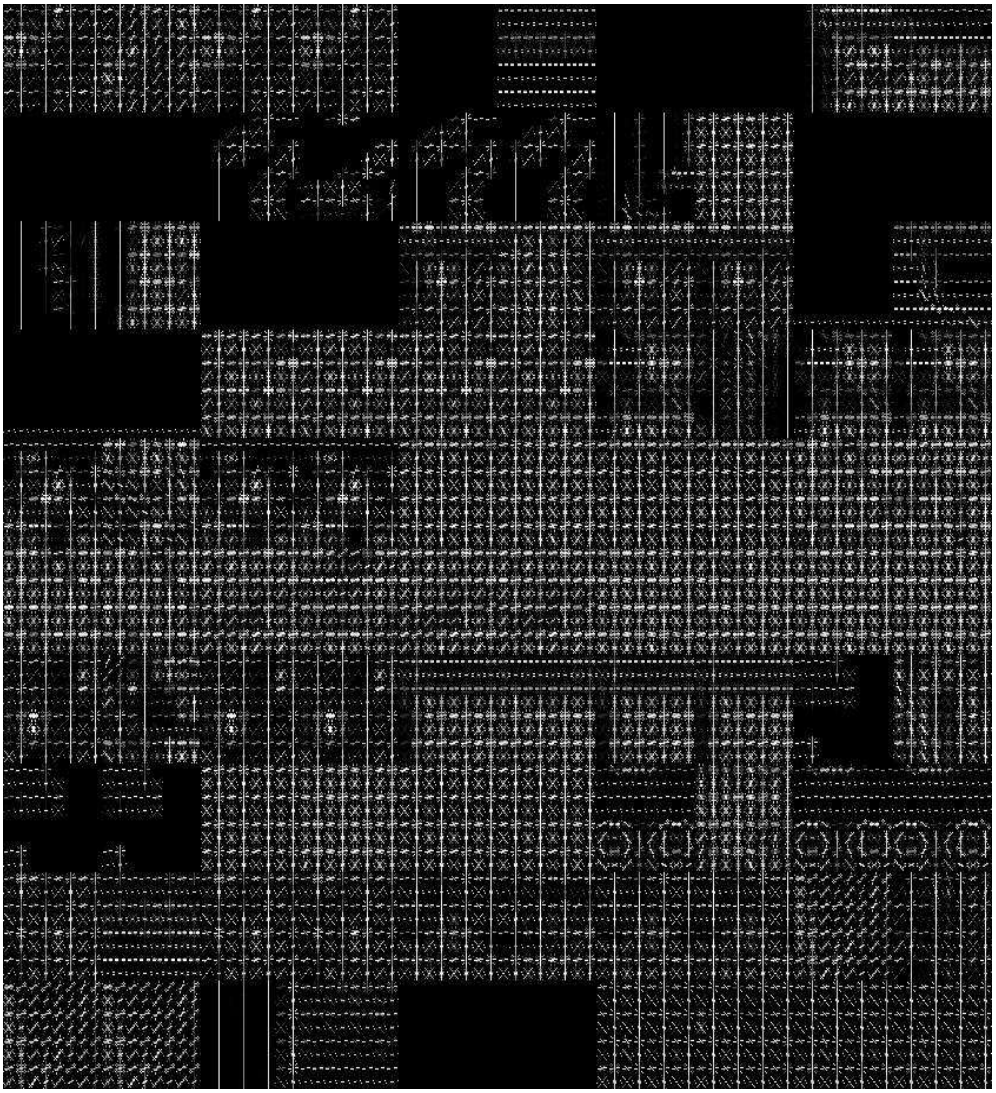

Figure 5.6. Visualization of Bag of visual words model after K-Means clustering

5.2 THE PROPOSED CLASSIFIERS

As observed in sections 4.2 and 4.3 of chapter 4, the overall performance of the ORB has been improved through the *HORB* and the *ACORB,* but it does not yield better, in fact, does not even improve, the performance of partially visible images. With an intention to

improve the performance for partially visible images, the two classifiers the *HORBoVF* and the *ACORBoVF* have been developed which have been explained in the following text.

5.2.1 The *HORBoVF* – A Histogram, the ORB, and a Bag of Visual Features based 3-Stage Hybrid Classifier

The *HORBoVF* is a three-stage classifier which is a fusion of Histogram matching, the ORB feature detector and a Bag of visual words. Here, instead of simply creating a visual vocabulary, a Bag of Visual Features – A dynamically created visual dictionary of specific features of images is generated. This visual vocabulary is created using the ORB feature detector and K-Means clustering. The clusters are formed based on the relevant features of the images. Once the clusters are formed, later, this visual dictionary is used for labeling of an image.

First, the captured image is converted into a histogram and matched with a histogram of dataset images. Histogram intersection is used for finding the closest match. Here, the *HORB*, proposed in section 4.2 of chapter 4, is used and output of the *HORB*, the top *k* feature subsets are sent to the next layer for image matching using the ORB. The feature extraction and matching for each of the given subsets with the features of the given image is carried out. The output of this layer is sent to the third and final stage of classification for verification purpose where clustering takes place, and labeling is done for the given image using the SVM classifier. The proposed Bag of Visual Features approach is shown in the figure below with the overall dictionary creation process:

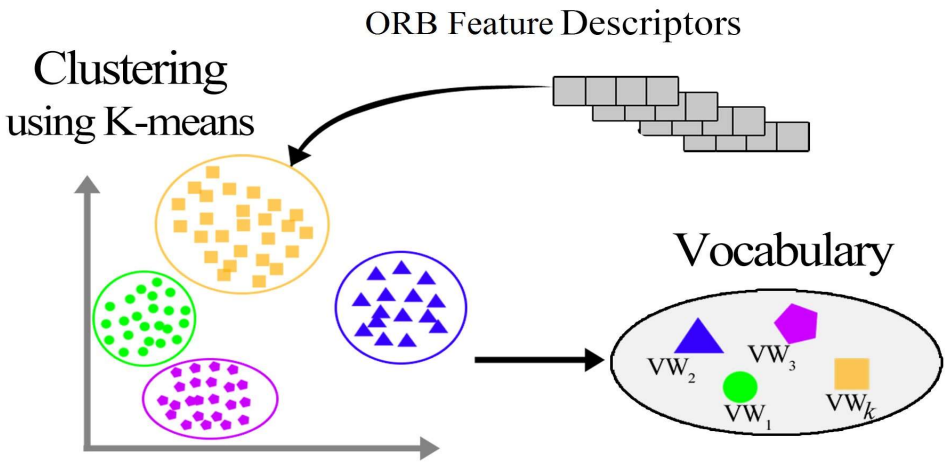

Figure 5.7. ORB based dictionary vocabulary creation process

The purpose of the third layer is just the verification of the output of the second layer. The whole algorithm has been divided into a three-stage filtered classification process. The overall algorithm is described below:

1. Initialize *totalNoOfImages (N)* in the dataset, the *featureSet* with distinguishable features for all images, *imageObject*, in the dataset.
2. Initialize *featureThreshold, similarityDistanceThreshold*
3. Histogram Intersection Filtering:
   a. Generate histograms of all *N* images, *imageObject*, in dataset and histogram of *inputImage*
   b. For each histogram of *imageObject*, perform histogram intersection with a histogram of *inputImage*
   c. Find top *K* histogram intersection values and corresponding *imageObject* into *topKIntersects* and *topKImageObjects,* where $K < N$.
4. ORB based decision making:
   a. Create an ORB feature detector for *inputImage*
   b. For each *imageObject* in *topKImageObjects* with the corresponding *featureSet*, Perform feature matching:
      If *imageObject* passes *featureThreshold* and *similarityDistanceThreshold* then,
         Add it to the *LabelList*.
   c. Use the best-matched *imageObject* from *LabelList* as an output image.
5. Decision Verification using BoVF:
   a. Generate raw bag of feature descriptors.
   b. Perform K-Means clustering to create a visual vocabulary of feature descriptors.
   c. Find appropriate descriptor for *inputImage* using visual vocabulary.
   d. *Label* the image and verify it with the output of stage 4.

The HORBoVF algorithm

The following figure shows the structure and steps of the *HORBoVF* algorithm visually.

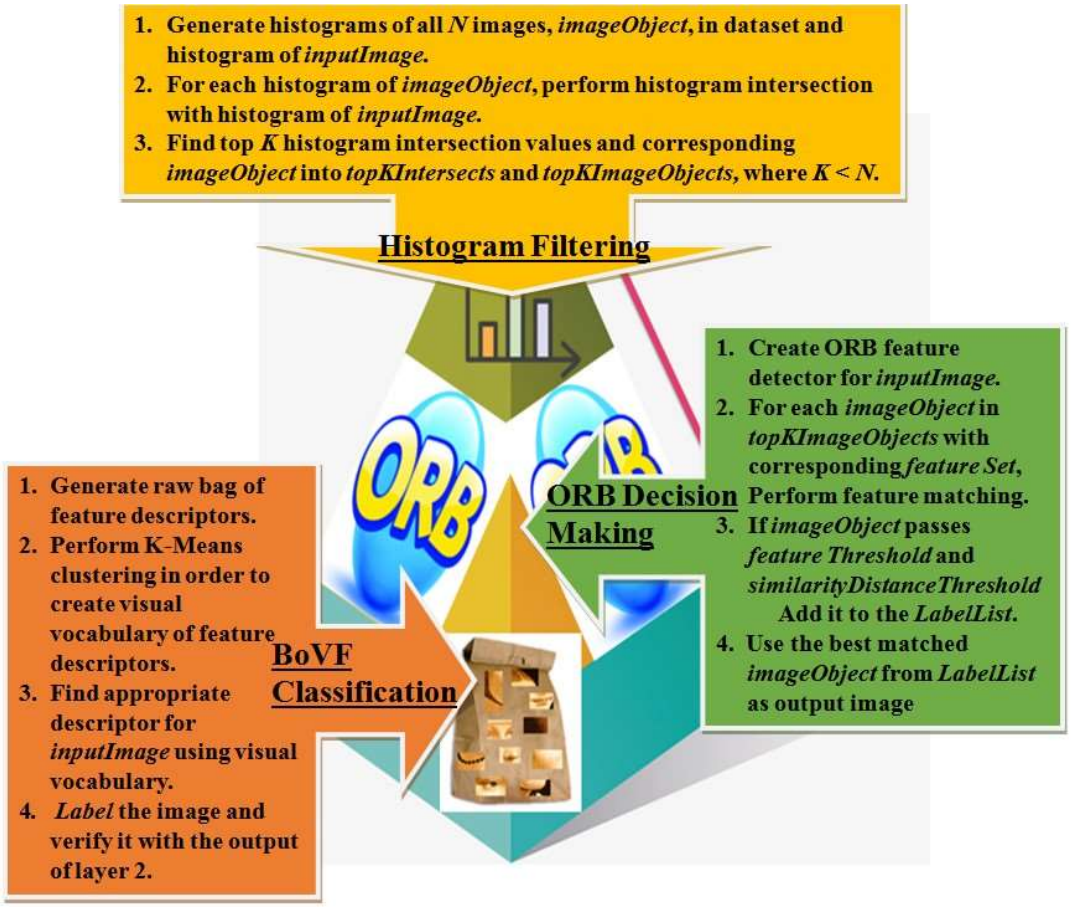

Figure 5.8. The *HORBoVF* Pyramid

Due to three stages in the *HORBoVF*, complexity is also explained individually. The complexity of the *HORB* has already been discussed in 4.2 of chapter 4, which is $O(I_i*D_i*n)+O(k*F_I*F_D)$. Where I_i is the i^{th} pixel of the histogram of the input image and D_i is i^{th} pixel of image D from a dataset of n images during histogram intersection. After, histogram intersection, for feature matching of remaining k $(k<n)$ feature subsets, F_I is the number of features of the input image I and F_D is a number of features of each of the feature subset. The K-Means clustering has the complexity of $O(n*F_D)$, where n is the number of images in the dataset and F_D is the number of features of each image. The complexity of the SVM classifier is $O(n^3)$, where n is the size of the training set. The detailed result is discussed in section 5.3.

5.2.2 The *ACORBoVF* – An ACO, the ORB, and a Bag of Visual Features based 2-Stage Hybrid Classifier

It is proved in section 4.3 of chapter 4 that the performance of the ORB can be improved through the ACO. This same heuristic-based approach of the *ACORB* has been combined

with Bag of Visual Features' classifier. This is a two-stage classifier and in the line of the *HORBoVF*. Here, the same dynamic Bag of Visual Features, created for the *HORBoVF*, is being used for labeling of the images. The overall algorithm has been divided into two stage classification process; one layer is less than the *HORBoVF*. In the first stage, the captured image is used as an input to the *ACORB* and the closest match is decided. The matched label for the closest image is sent to the next stage of classification for verification purpose. The *ACORBoVF* algorithm is described below:

1. Initialize the *numberOfAnts*, *pheromone*, *maximumIterations*, *distanceThreshold* and *foodQty*
2. Assign any random *Ant* to the *foodLocation*. Evaluation criteria are the least distance between the *Ant* and the *foodLocation,* i.e., *distanceThreshold* and the *foodQty*.
   a. If the *Ant* does not get food at the *foodLocation* in specific *foodQty* in *maximumIterations* then,

      Select another *Ant* to build the solution.

      else

      Evaluate the *Ant*'s food selection based on *distanceThreshold*.
   b. If food selection passes the *distanceThreshold*, add it to *AntList* and update the *pheromone* value
   c. Select the best *Ant* from AntList for labeling.
3. Decision Verification using BoVF
   a. Generate raw bag of feature descriptors.
   b. Perform K-Means clustering to create a visual vocabulary of feature descriptors.
   c. Find appropriate descriptor for *inputImage* using visual vocabulary.
   d. *Label* the image and verify it with the output of stage 2.

The ACORBoVF algorithm

The following figure shows how the *ACORB* meets BoVF.

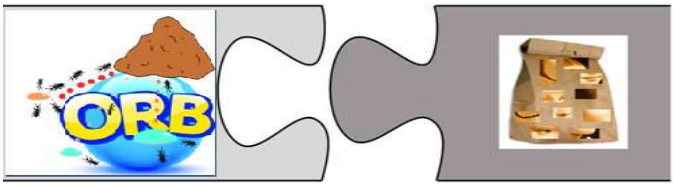

Figure 5.9. *The ACORBoVF* Structure

Due to the two stages, the complexity of the algorithm is also explained individually for each stage. As it is discussed in 4.3 of chapter 4 that the complexity of the *ACORB* is $O(n*F_I*F_D*m)$ where n is the maximum number of iterations, F_I is the number of features of the input image I, F_D is a number of features of each of the feature subset ant and m is the number of ants. The K-Means clustering has the complexity of $O(n*F_D)$, where n is the number of images in the dataset and F_D is the number of features of each image. The complexity of the SVM classifier is $O(n^3)$, where n is the size of the training set. The result analysis is discussed in details in the next section.

5.3 THE TESTING AND PERFORMANCE ANALYSIS OF THE *HORBoVF* AND THE *ACORBoVF*

This section discusses performance analysis of the *HORBoVF* and the *ACORBoVF* jointly as both of these uses a common classifier, dynamic BoVF in the final verification stage after feature detection using the *HORB* and the *ACORB* respectively. Since the performance analysis of the *HORB* and the *ACORB* has already been discussed in 4.2.2 and 4.3.2 of chapter 4, this section will discuss the performance of Bag of Visual Features, only, in context of different values of K (Number of Clusters) used in K-Means clustering.

To measure the accurate performance, the initial value of K is taken as 10 and has been incremented in the successive iterations with values 15, 20 and 24. The value ten has been selected as the initial value as there are ten different denominations, classes, of Indian currencies. Also, in the *HORB* and the *ACORB*, the top 10 images are evaluated for labeling. The value 24 has been taken since there are a total of 24 different types of images, considering the front and back side of the currency, including the new currencies. The variations in the K have been taken to find a specific value of K beyond which the further classification would be useless. The clusters, here, are created based on the unique features detected via ORB for each image. The following table 5.1 shows the time taken by K-Means clustering to create a visual dictionary for different denominations of the Indian currencies along with various values of K.

Denomination	# of Images	Time Taken to Create Bag of Words (Seconds)			
		K=10	K=15	K=20	K=24
5	2	0.84	0.88	1.13	1.27
10	4				
20	2				
50	4				
100	2				
200	2				
500	2				
500_old	2				
1000	2				
2000	2				

Table 5.1 Dictionary creations and Clustering time for different values of K

The following subsections discuss results and analysis for different values of K.

5.3.1 For K=10

The following table 5.2 shows the overall performance of the *HORBoVF* for K=10

Denomination	# of Test Images		# of Correct Classes		Execution Time (Seconds)		Classification Accuracy (%)	
	F*	P**	F	P	F	P	F	P
5	130	80	116	49	194.63	84.37	89.23	61.25
10	226	80	201	54			88.94	67.5
20	220	80	196	48			89.09	60
50	228	80	199	49			87.28	61.25
100	268	80	228	46			85.07	57.5
200	139	52	117	35			84.17	67.31
500	160	78	140	42			87.50	53.846
500_old	165	80	138	48			83.64	60
1000	83	80	72	47			86.75	58.75
2000	200	80	179	41			89.50	51.25
Average time taken for an image & Accuracy					**0.107**	**0.1095**	**87.19**	**59.61**
Overall Accuracy = 78.988%					**Average Time = 0.1077 Seconds**			

Table 5.2 The *HORBoVF* Performance for K=10

Here, F indicates the fully visible images whereas P indicates the partially visible images. For K=10, in comparison to 1352 out of 2589 images for the ORB, the *HORBoVF* identifies 2045 images correctly giving an overall accuracy of 78.988% as

compared to the ORB's 52.220%. Here, for the fully visible images, the accuracy is 87.191%. The noticeable change is for the partially visible images, where the accuracy is 59.610% giving a tremendous 50% increase in the accuracy as compared to the ORB. This result shows that the motive for using the Bag of Visual Features as a classifier has been a successful attempt. The total number of correctly classified images and its accuracy are shown in the following figure 5.10 and figure 5.11 respectively.

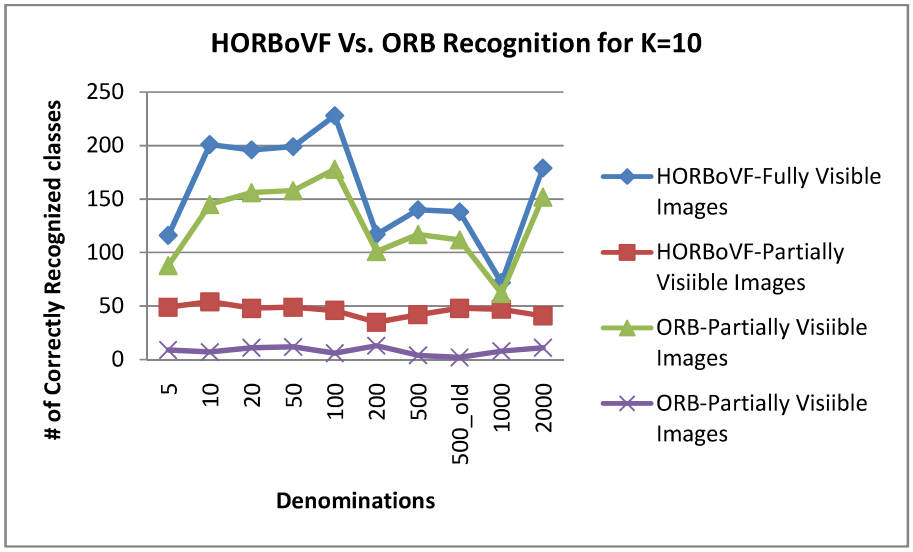

Figure 5.10. Number of correctly identified images using the *HORBoVF* and the ORB

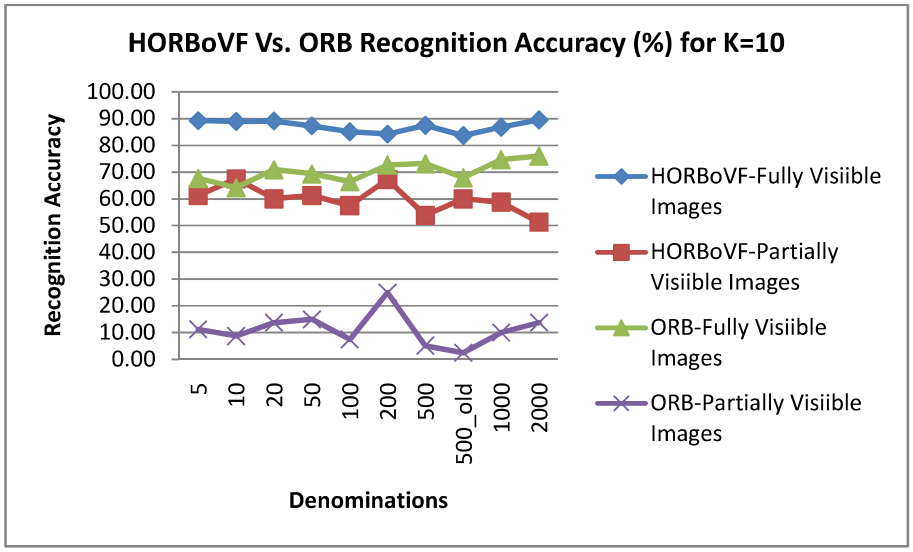

Figure 5.11. Recognition accuracy of the *HORBoVF* and the ORB

Once the clusters are formed, the algorithm takes an average 0.107 seconds to label the image for both kinds of images, as visible in figure 5.12. So, for 2.335 seconds of the *HORB* per image for feature detection, it sums up to 2.442 seconds, excluding cluster formation, giving 26.767% higher accuracy than the ORB, overall.

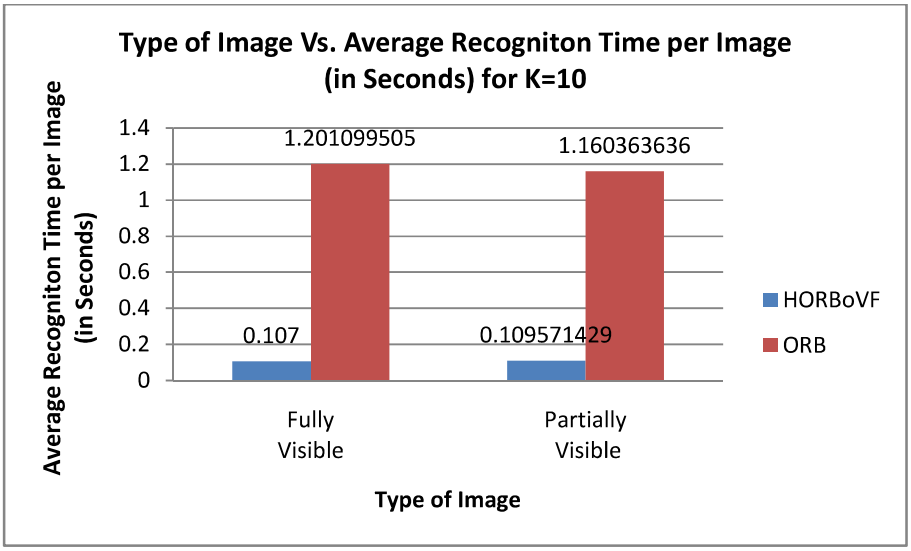

Figure 5.12. Average time taken per image by the *HORBoVF* and the ORB

The overall performance is shown in the following figure 5.13.

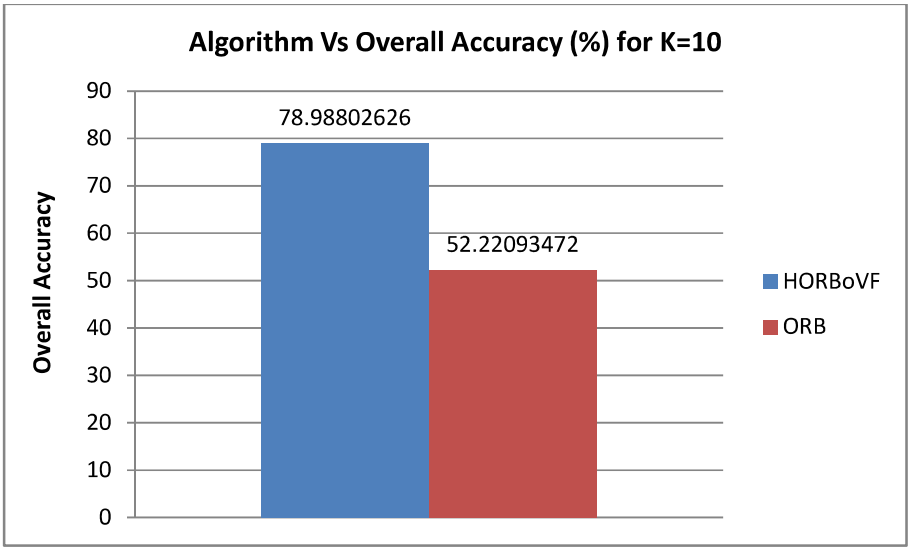

Figure 5.13. The overall performance of the *HORBoVF* and the ORB

5.3.2 For K=15

For K=15, the following table 5.3 shows the summarized performance of the *HORBoVF*. Here, in comparison to 1352 out of 2589 images for the ORB, the *HORBoVF* identifies

2155 images correctly giving an overall accuracy of 83.237% as compared to the ORB's 52.220%. Here, for the fully visible images, the accuracy is 90.159%, and the same for the partially visible images is 66.883%.

Denomination	# of Test Images		# of Correct Classes		Execution Time (Seconds)		Classification Accuracy (%)	
	F	P	F	P	F	P	F	P
5	130	80	118	54	202.36	87.49	90.77	67.5
10	226	80	205	56			90.71	70
20	220	80	195	52			88.64	65
50	228	80	209	51			91.67	63.75
100	268	80	238	50			88.81	62.5
200	139	52	129	39			92.81	75.00
500	160	78	143	54			89.38	69.231
500_old	165	80	146	53			88.48	66.25
1000	83	80	78	57			93.98	71.25
2000	200	80	179	49			89.50	61.25
Average time taken for an image & Accuracy					**0.1112**	**0.1136**	**90.15**	**66.88**
Overall Accuracy = 83.237%					**Average Time = 0.1119 Seconds**			

Table 5.3 The *HORBoVF* Performance for K=15

The total number of correctly classified images and its accuracy are shown in the following figure 5.14 and figure 5.15 respectively.

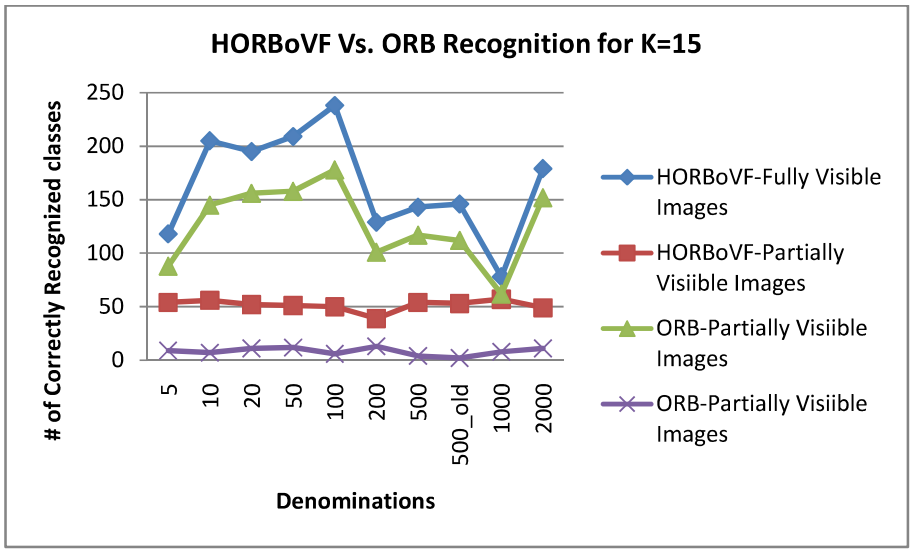

Figure 5.14. Number of correctly identified images using the *HORBoVF* and the ORB

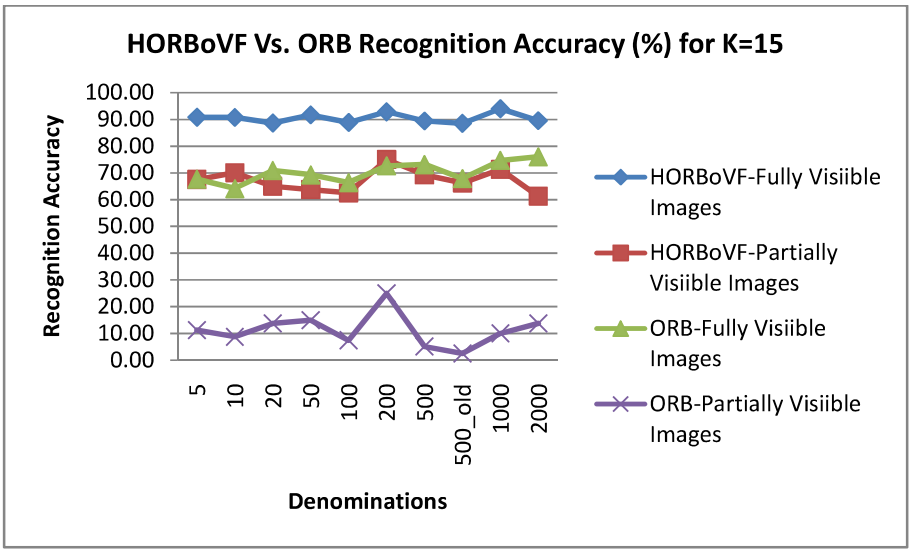

Figure 5.15. Recognition accuracy of the *HORBoVF* and the ORB

Once the clusters are formed, for K=15, the algorithm takes an average of 0.112 seconds to label the image for both kinds of images as visible in figure 5.16. So, with 2.335 seconds of the *HORB* per image for feature detection, it sums up to 2.447 seconds only, after cluster formation, giving 31.015% higher accuracy than the ORB. Here, it can be observed that the labeling time remains almost the same as for K=10.

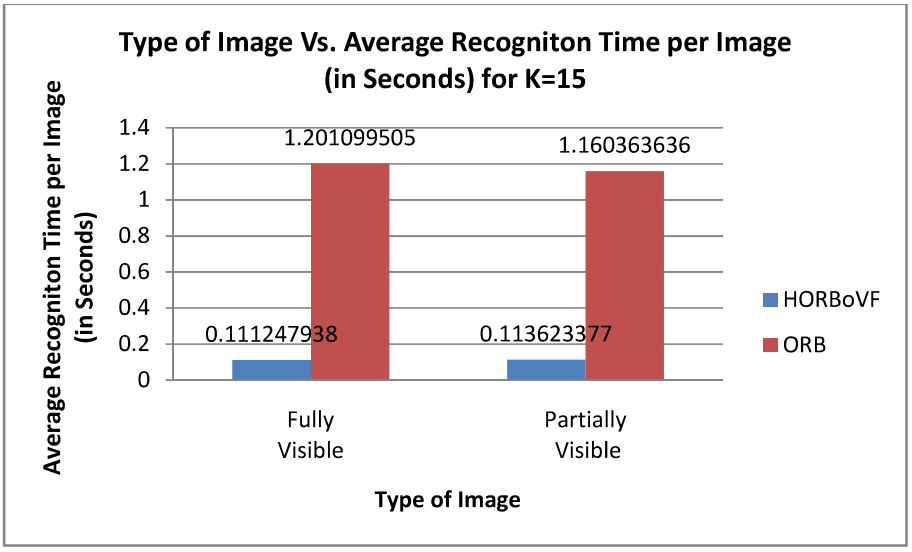

Figure 5.16. Average time taken per image by the *HORBoVF* and the ORB

The overall performance is shown in figure 5.17.

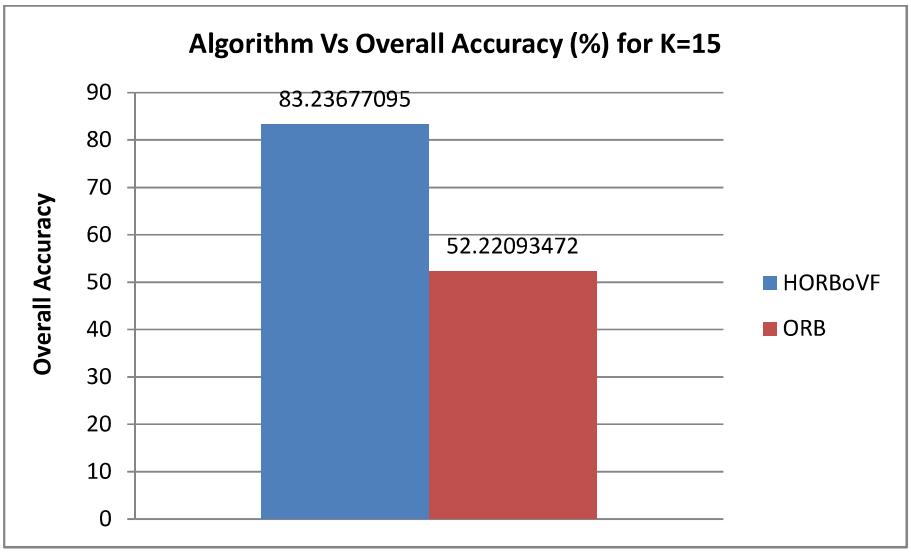

Figure 5.17. The overall performance of the *HORBoVF* and the ORB

5.3.3 For K=20

For K=20, the following table 5.4 shows the overall performance of the *HORBoVF* For K=20, in comparison to 1352 out of 2589 images for the ORB, the *HORBoVF* identifies 2368 images correctly giving an overall accuracy of 91.464% as compared to the ORB's 52.220%. The eye-catching results are visible here. For the fully visible images, the accuracy has reached 98.241%, and the same for the partially visible images is 75.454% giving an increase of almost 8-9% in both types of images than that for K=15.

Denomination	**# of Test Images**		**# of Correct Classes**		**Execution Time (Seconds)**		**Classification Accuracy (%)**	
	F	**P**	**F**	**P**	**F**	**P**	**F**	**P**
5	130	80	128	59	212.83	91.03	98.46	73.75
10	226	80	223	63			98.67	78.75
20	220	80	218	56			99.09	70
50	228	80	225	61			98.68	76.25
100	268	80	263	59			98.13	73.75
200	139	52	137	42			98.56	80.77
500	160	78	156	58			97.50	74.359
500_old	165	80	159	58			96.36	72.5
1000	83	80	81	63			97.59	78.75
2000	200	80	197	62			98.50	77.5
Average time taken for an image (seconds)					**0.117**	**0.1182**	**98.24**	**75.45**
Overall Accuracy = 91.464%					**Average Time = 0.1173 Seconds**			

Table 5.4 The *HORBoVF* Performance for K=20

The total number of correctly classified images and its accuracy are shown in the following figure 5.18 and figure 5.19 respectively.

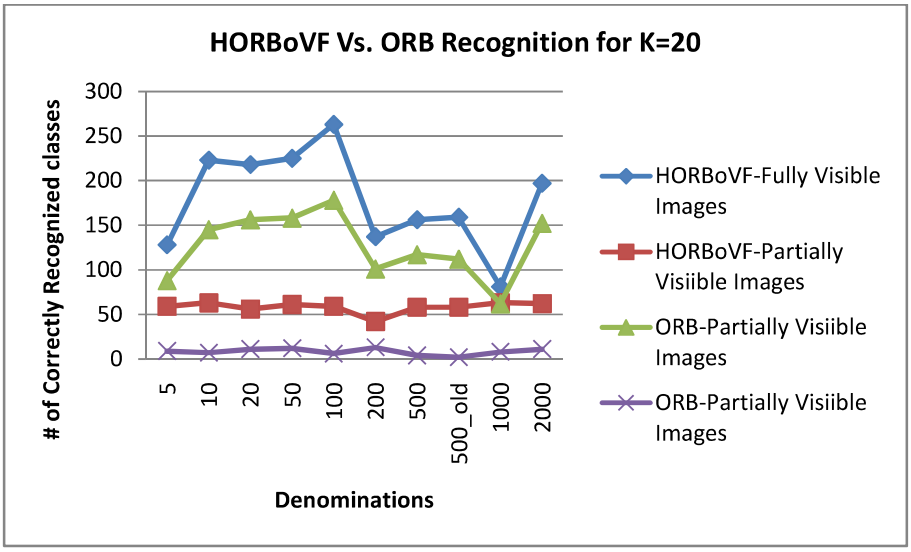

Figure 5.18. Number of correctly identified images using *HORBoVF* and ORB

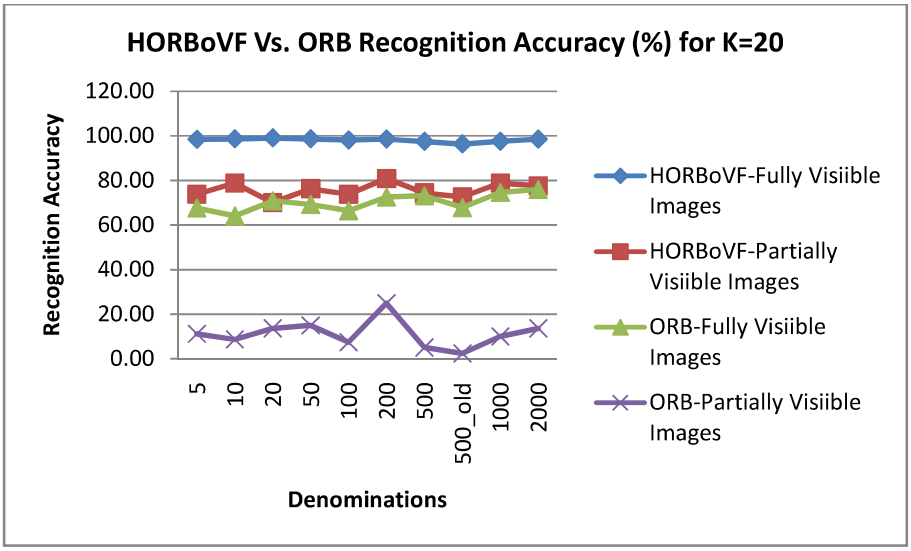

Figure 5.19. Recognition accuracy of *HORBoVF* and ORB

Once the clusters are formed, for K=20, the algorithm takes an average of 0.117 seconds to label the images of both kinds, as visible in figure 5.20. So, for 2.335 seconds of the *HORB* per image for feature detection, it sums up to 2.452 seconds and gives 39.242% higher accuracy than the ORB. Here also, it can be observed that the labeling time remains almost the same.

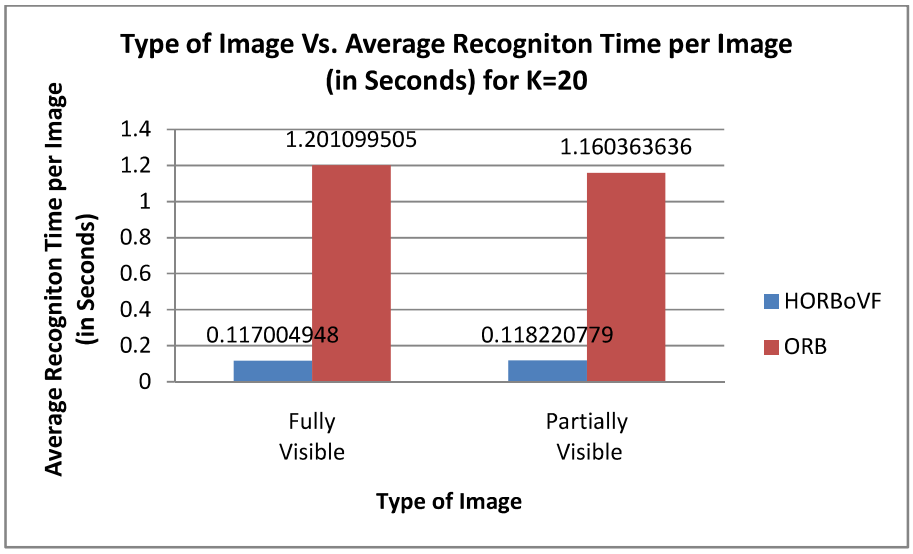

Figure 5.20. Average time taken per image by the *HORBoVF* and the ORB

The overall performance is shown in the following figure 5.21.

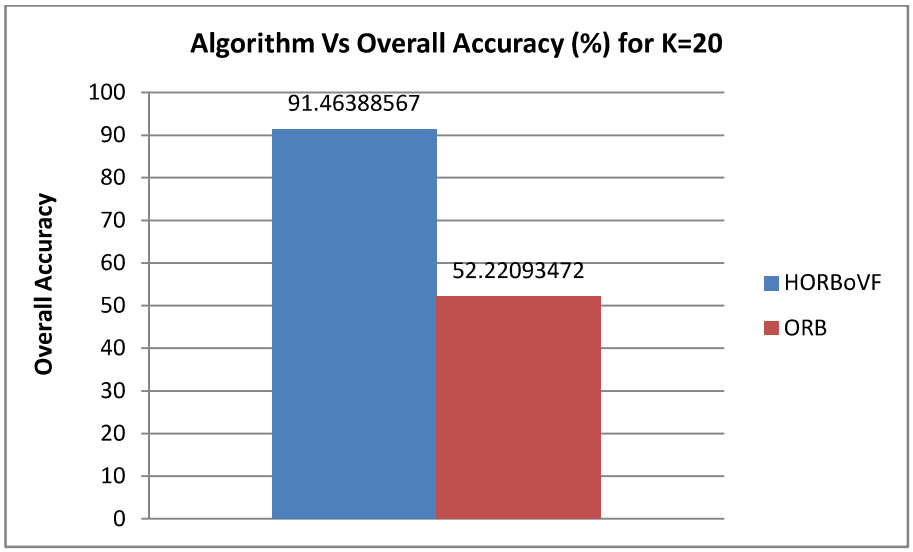

Figure 5.21. The overall performance of the *HORBoVF* and the ORB

5.3.4 For K=24

For K=24, the following table 5.5 shows the almost identical performance of the *HORBoVF* as it is for K=15. It can be observed that in comparison to 1352 out of 2589 images for the ORB, the *HORBoVF* identifies 2370 images correctly, giving an overall accuracy of 91.541% as compared to the ORB's 52.220%. For the fully visible images, the accuracy is 98.351%, and the same for the partially visible images is 75.454%. Both are almost the same as what has been observed for K=20. This indicates that there are no

further chances of more clustering. Logically, there would not be further clustering as there are 24 different types of images for labeling.

Denomination	# of Test Images		# of Correct Classes		Execution Time (Seconds)		Classification Accuracy (%)	
	F	P	F	P	F	P	F	P
5	130	80	128	59	212.97	91.03	98.46	73.75
10	226	80	223	63			98.67	78.75
20	220	80	218	56			99.09	70
50	228	80	225	61			98.68	76.25
100	268	80	263	59			98.13	73.75
200	139	52	137	42			98.56	80.77
500	160	78	156	58			97.50	74.359
500_old	165	80	161	58			97.58	72.5
1000	83	80	81	63			97.59	78.75
2000	200	80	197	62			98.50	77.5
Average time taken for an image (seconds)					**0.1171**	**0.1182**	**98.35**	**75.45**
Overall Accuracy = 91.541%					**Average Time = 0.1174 Seconds**			

Table 5.5 The *HORBoVF* Performance for K=24

The total number of correctly classified images and its accuracy are shown in the following figure 5.22 and figure 5.23 respectively.

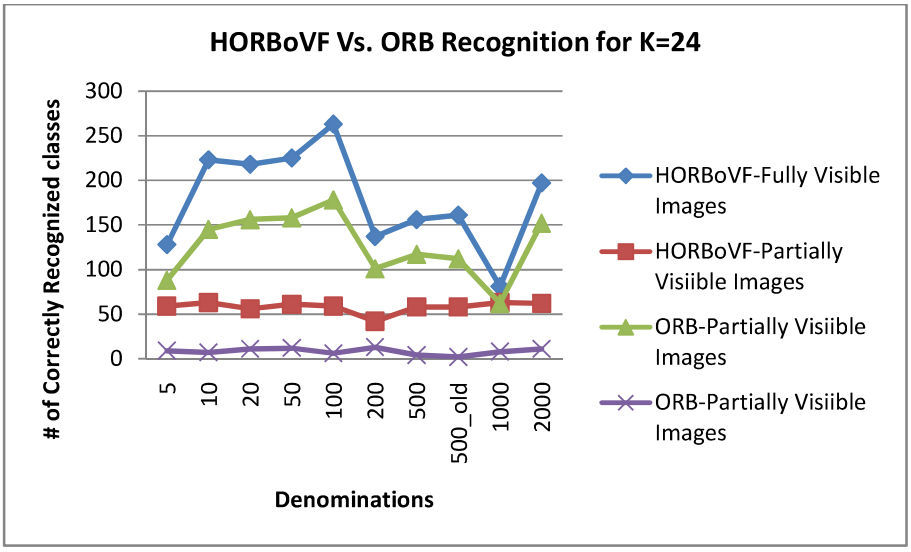

Figure 5.22. Number of correctly identified images using the *HORBoVF* and the ORB

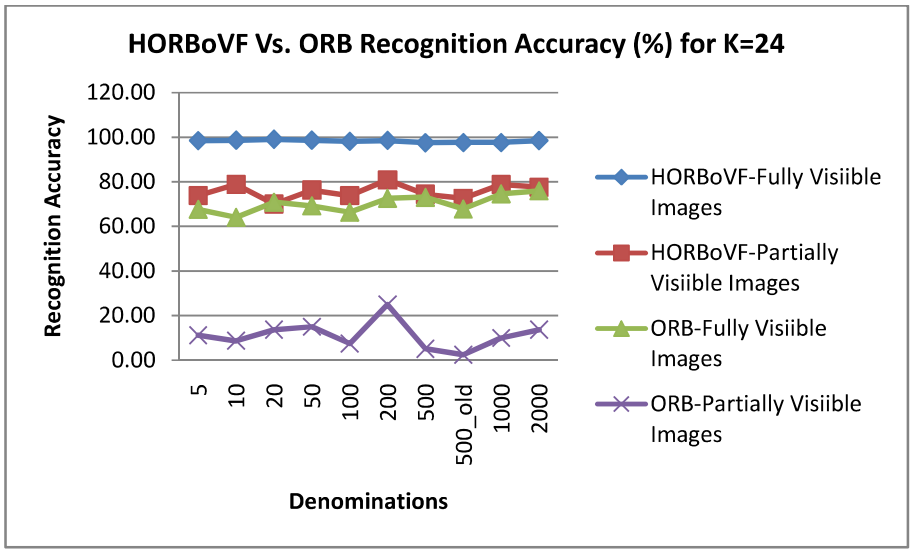

Figure 5.23. Recognition accuracy of the *HORBoVF* and the ORB

Even the labeling time too remains the same as what has been observed for K=20. Once the clusters are formed, for K=24, the algorithm takes an average of 0.117 seconds to label the image for both kinds of images which is the same as for K=20. This is shown in figure 5.24.

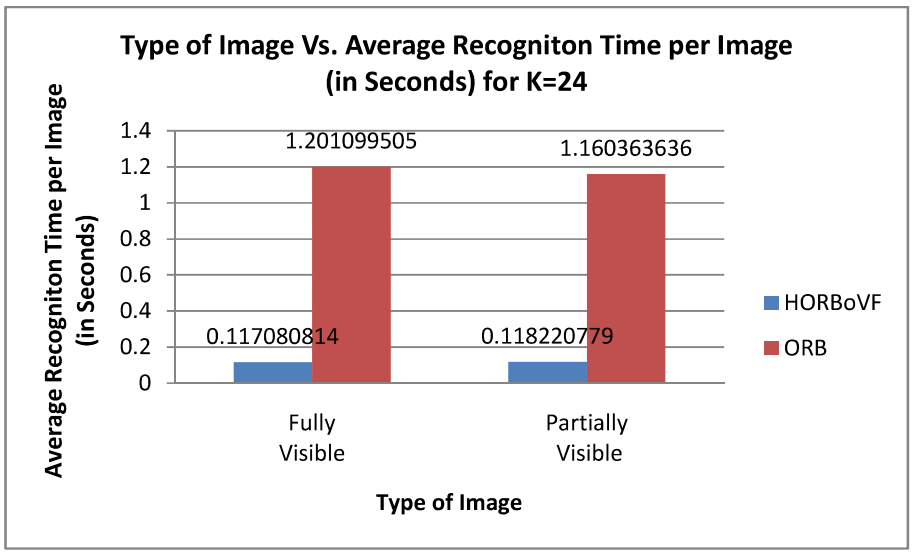

Figure 5.24. Average time taken per image by the *HORBoVF* and the ORB

The overall performance of the *HORBoVF* remains same for K=24 too as shown in figure 5.25, indicating that there are no further chances of improvement

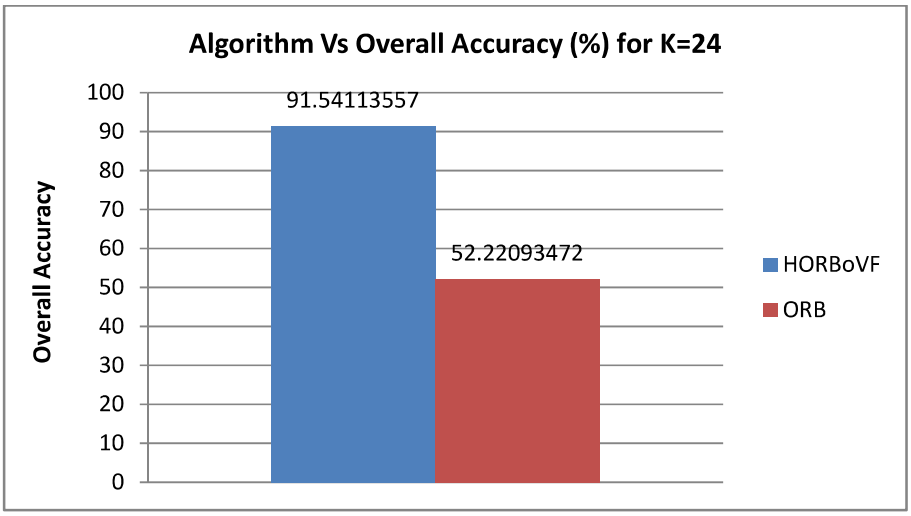

Figure 5.25. The overall performance of the *HORBoVF* and the ORB

Now, to consider the *ACORBoVF*, performance point of view, there will be the difference in overall time consumption only since the clusters are formed based on the features of the images using the same K-Means clustering approach and hence the clusters are going to remain same. As discussed in section 4.3.2 of chapter 4, the *ACORB* takes an average 2.283 seconds to detect the features. So, for K=24, with 0.117 seconds for labeling using BoVF, the total classification time sums up to 2.4 seconds, a little less than that of the *HORBoVF*. Though, as a feature detector, accuracy point of view the *ACORB* performs 0.386% better than the *HORB*, but since the final stage uses the same classifier, the overall accuracy of the *ACORBoVF* will remain same, i.e., 91.541%. The summarized performance of all the clusters in terms of time and accuracy is given below.

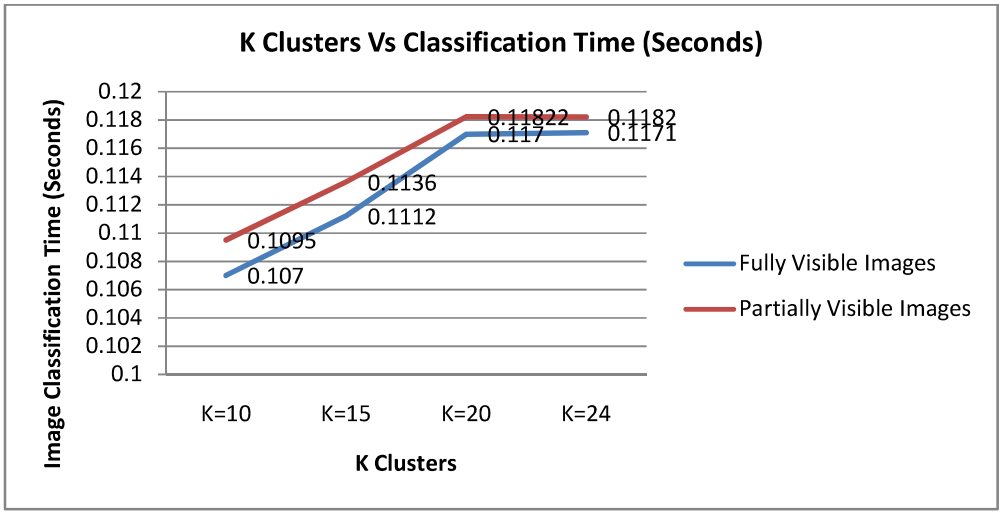

Figure 5.26. Average time consumption by the *HORBoVF* for different values of K

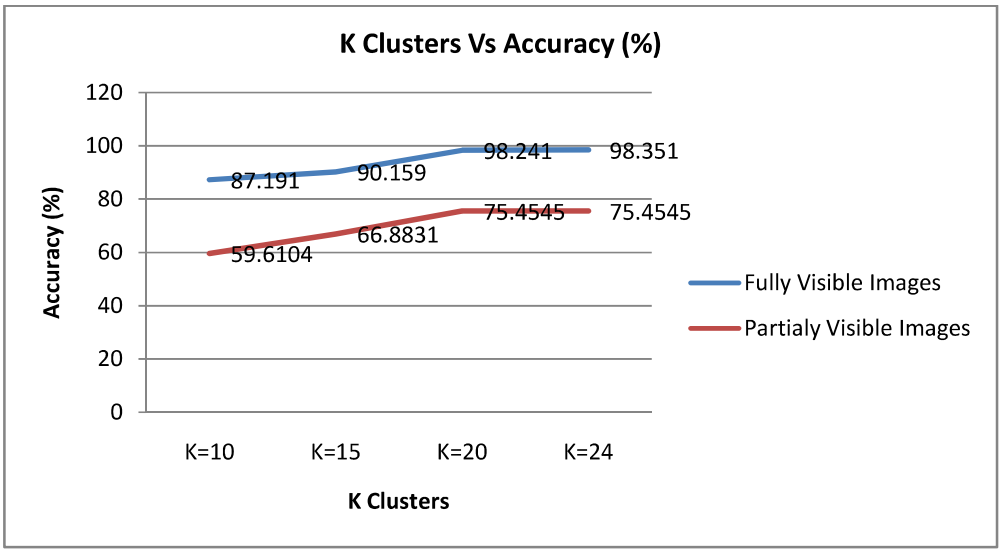

Figure 5.27. The overall accuracy of the *HORBoVF* for different values of K

The steep increase in accuracy of labeling of images is due to bag of visual words which in turn uses clustering approach to generate visual vocabulary of the features.

SUMMARY

This chapter introduced the image classification and bag of visual words in details. Then, it discussed the two novel image classifiers, the *HORBoVF* and the *ACORBoVF*, which are based on the *HORB* and the *ACORB* with a bag of visual words. Finally, it gave an insight into a detailed performance analysis of the proposed classifiers with reference to the ORB in terms of time and accuracy both. The next chapter discusses a TensorFlow-based implemented and retrained model, *Te₹₹*ency, with detailed performance analysis.