# SIMULATION RESULTS AND DISCUSSIONS

The dataset used in this process is a 5min 33sec video. This video is converted into images by using FFMPEG. By using FFMPEG the video is converted into .mp4 file and then it converts .mp4 file into set of .png files as per our requirement. Here I considered 10,000 images as input. The steps involved in downloading, installing and functionality of FFMPEG is shown as below.

FFMPEG Installation

Download FFMPEG from <https://ffmpeg.zeranoe.com/builds/>. It is recommended to download 64-bit Downloads (Download FFMPEG git-30d1213 64-bit Static). Based on the platform we download the required version of FFMPEG.

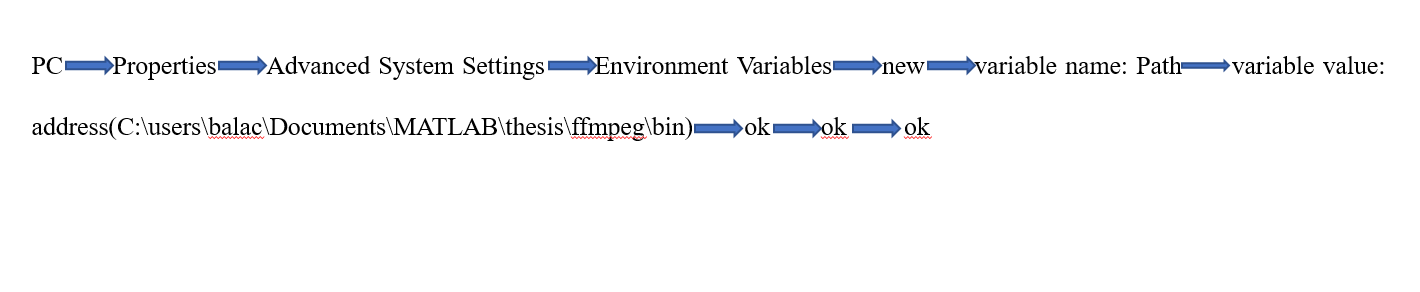
After downloading the FFMPEG we have to install the FFMPEG. To download the FFMPEG we have to make some changes in the properties of our computer. We have to add a new variable in the environment variables and the address of bin folder in FFMPEG is added as path to that new variable. The steps involved in the installation of FFMPEG are as shown below. 

Figure 8-1: Process of Installation

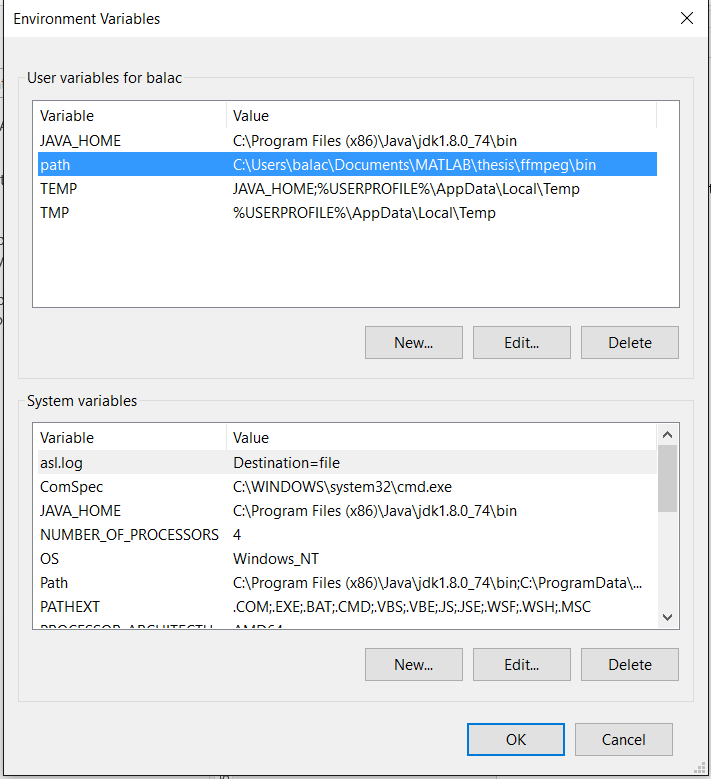


Figure 8-2: Installation of FFMPEG

After installing FFMPEG as shown in above figure, we can check the installation of ffmpeg by using the command: **ffmeg –version.** After using that command in command window we will get the result as follows.

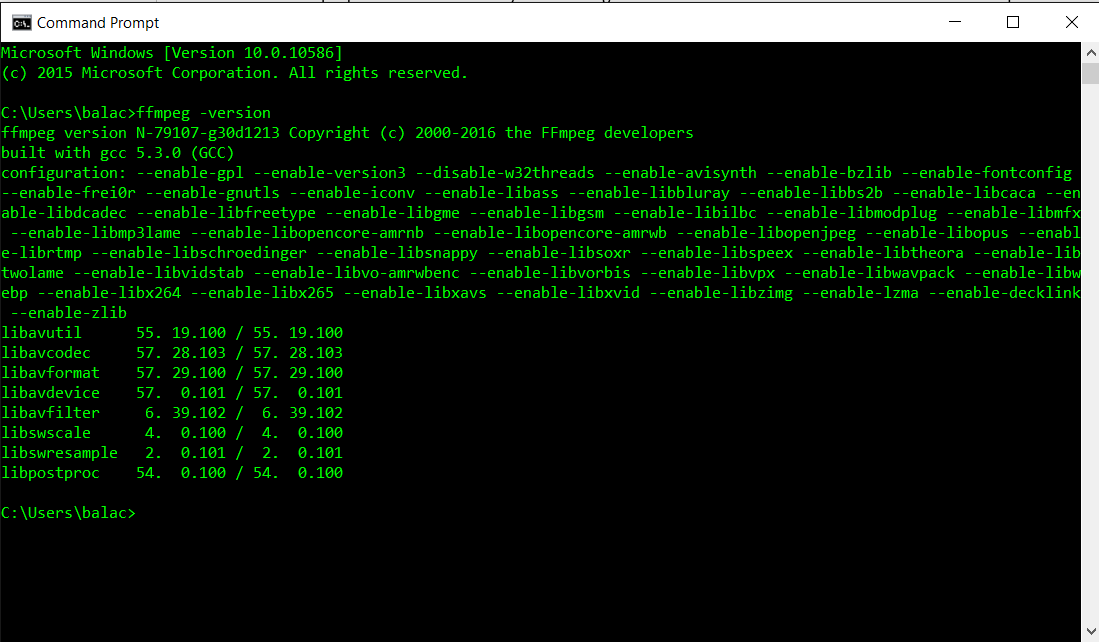


Figure 8-3: Checking the installation of FFMPEG

The input video must be converted into set of images. In this paper we are going to convert the input video into images with two resolutions. We are going to convert the video into images with resolution of 320×240 and 640×360.

The .yuv file cannot be directly decomposed into images. So, .yuv file can be converted into .mp4 file and then it is converted into .png files by using FFMPEG. The video which is in .yuv format is converted into .mp4 file by using the following command. In the command i is the input, -vf is textual representations and scale is the aspect ratio of the output.

*Command:* ffmpeg –i C:\Users\laksh\Documents\Thesis\Tools\input.yuv –vf scale=320:240 input.mp4.

This command results in .mp4 file named input-320kbps. Then we have to convert this .mp4 file to .png files.The video which is in .mp4 format is converted into .png files by using the following command.

*Command:* ffmpeg –i input-320kbps.mp4 image%d.png

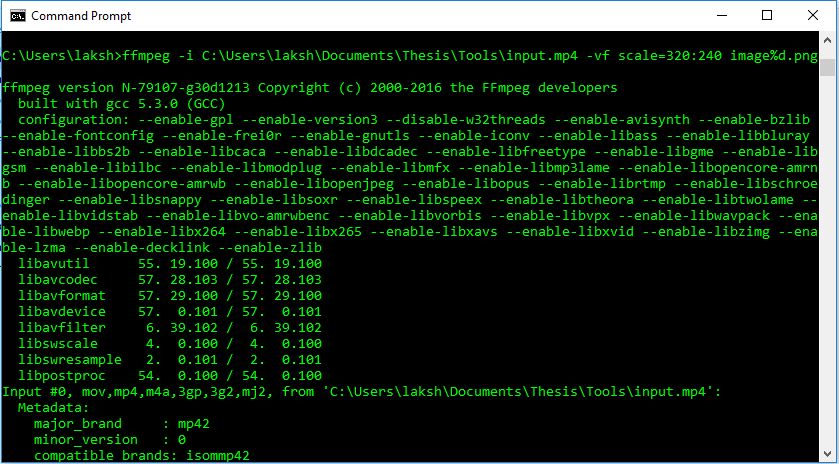


Figure 8-4: Conversion of video into images with resolution 320×240

This operation results in 2.5k images of resolution 320×240.

The .yuv file cannot be directly decomposed into images. So, .yuv file can be converted into .mp4 file and then it is converted into .png files by using FFMPEG. The video which is in .yuv format is converted into .mp4 file by using the following command. In the command i is the input, -vf is textual representations and scale is the aspect ratio of the output.

*Command***:** ffmpeg –i C:\Users\laksh\Documents\Thesis\Tools\input.yuv –vf scale=640:360 input.mp4.

This command results in .mp4 file named input-640kbps. Then we have to convert this .mp4 file to .png files. The video which is in .mp4 format is converted into .png files by using the following command.

*Command:* ffmpeg –i input-640kbps.mp4 image%d.png

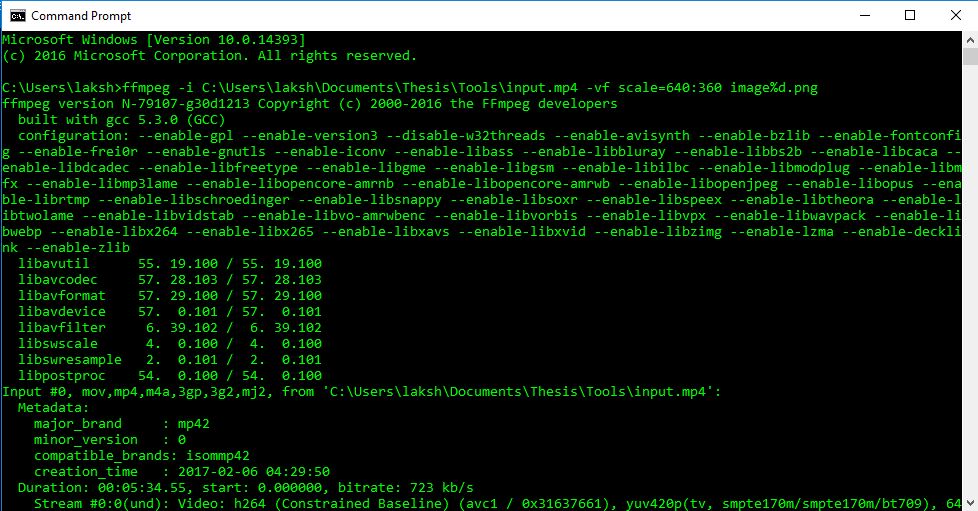


Figure 8-5: Conversion of video into images with resolution 640×360

A set of 2.5k images are generated and those images are stored in a folder. As we have large dataset, it is difficult and takes lot of time to process every image in the dataset. So, we have to limit the dataset by using Sampling. Sampling is the change of data by transforming continuous into discrete or discrete into continuous. There are two types of samplings upsampling and downsampling. Downsampling is decreasing the sample rate whereas upsampling is increasing the sample rate. In this paper, we have to reduce the dataset so we are using downsampling in this paper. In this paper, I have a dataset of 2.5k images. So, I used downsampling by the rate of 5.

no\_img=2500;

x=1:no\_img;

Indx=downsample(x, 5);

After downsampling, the resulting dataset contains 500 images. After downsampling we have to take an empty cell of 500×1 by using the command.

my\_celll=cell(size(indx,2),1)

After taking the empty single dimensional array, we have to insert the dataset into that one-dimensional array. Then we have to perform all operations by including vl\_feat. Vl\_feat contains the demo codes for sift, fisher, vlad and so many other operations. cdvs\_sift\_aggregation\_test data is loaded for sift\_cdvs and sift\_offs files.Then we have to load the trained data of pca and test data of sift. The dataset is to be scaled to 480 pixels in order to perform fisher aggregations. For fisher vectors the image is tuned to gray scale image for better classification. Fisher vectors are found by using gmm which contains means, covariance and priors. These are found by using number of clusters and data.

The size of fisher vectors is 500×1536. As we got more fisher vectors in which some are repetitive and unnecessary. So, we must find my significant fisher vectors. This is done by using a function called getFVSamples(). This is used to compute the frame significance value of each frame and a threshold is set up. If the frame significance value is greater than threshold, then it is taken into count. The frames which are greater than threshold are the significant frames. The plots of frame significance and cumulative frame significance values are as follows:

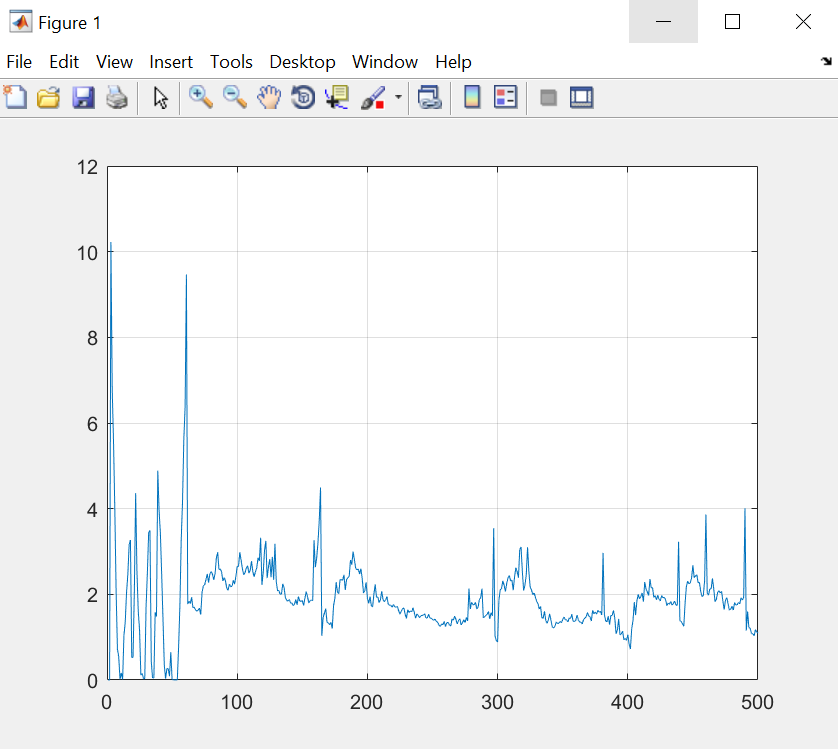


Figure 8-6: Plot of Frame Significance

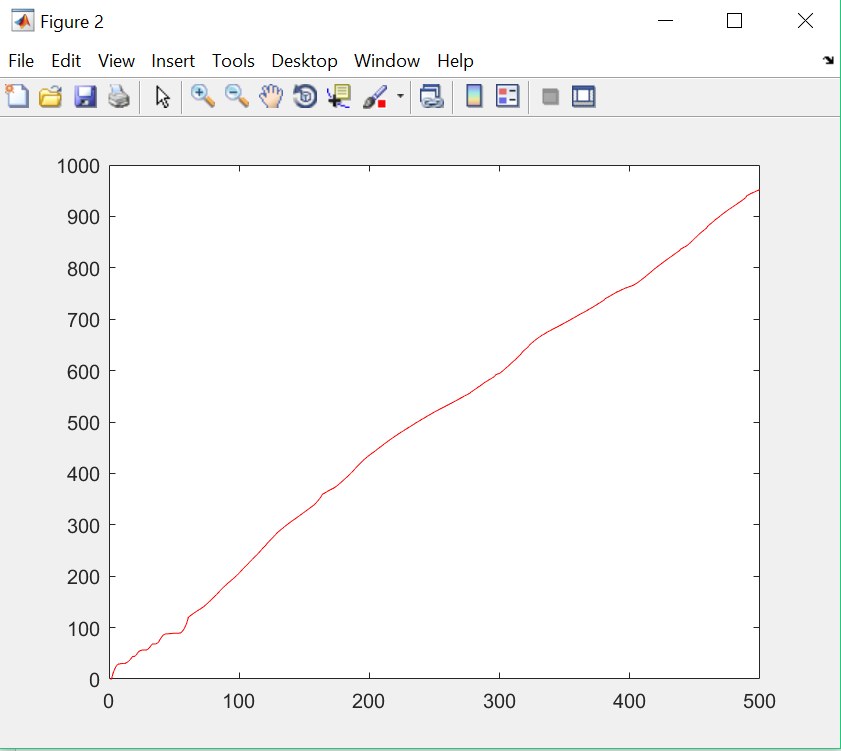


Figure 8-7: Plot of Cumulative frame significance

But these significant frames consist of some repetitive frames. So, fisher aggregation is to be performed out of which the result is 20 frames and it is named as fv. The performance of the entire technique is evaluated by using ROC plots by using True positive rate and false positive rate.

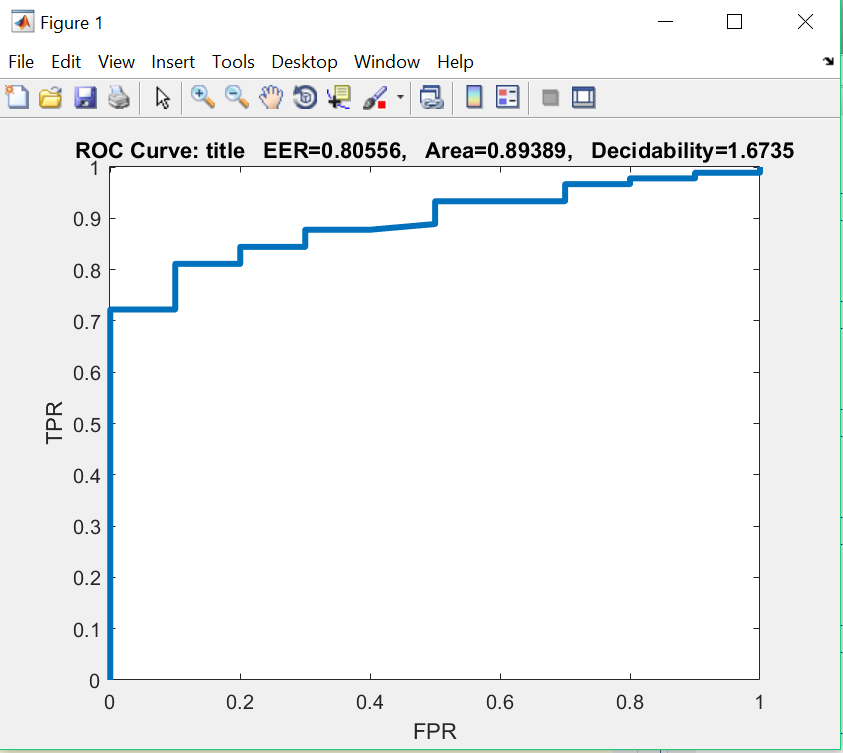


Figure 8-7: ROC curve for fisher aggregation of dataset with resolution 640×360

This is the ROC curve for 640×360 resolution. This is plotted against the True Positive Rate and False Positive Rates. By applying a threshold value to the classifier True positive and false positive rates can be obtained by which Receiver Operating Characteristics can be plotted. The AUC (Area under curve) for this plot is 0.89 which means that the classifier used in this paper is best classifier.

After aggregating the fisher vectors, we are going to convert these fisher vectors into hash codes which consists of 1’s and 0’s. By using hash frame distance, we are going to generate a sequence. Now, we are conducting the similarity check for deduplication. For deduplication, we are training some videos in database then checking the existing video with the videos in database.

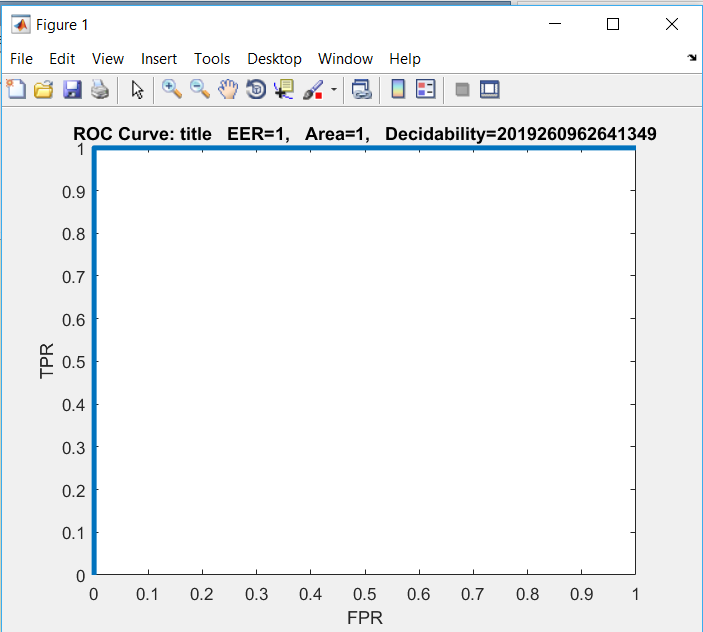


Figure 8-8: ROC curve for sequence level