Indian Institute of Science Education and Research
Bhopal

Computer Vision(DSE-312/EECS-320)

Assignment-2

Name: Adheesh Trivedi
Roll No.: 22016
Time of submission: Marks Obtained:

Please follow the instructions given in the assignment carefully.

1. **All questions are mandatory. Plagiarism and copying from anywhere (similar submission) can debar you from this course and invite the academic dishonesty policy.**
2. **Implement all algorithms purely in Python without using specialized libraries like OpenCV or PIL for the processing. You may use libraries for basic operations (like loading an image), but the algorithms should be coded from scratch.**
3. **Comment on your code extensively to explain your logic and the steps you are implementing.**
4. **Display both the original and processed images to compare results.**
5. **Make a short 7-minute video and explain your code.**
6. **A report reflecting on what you have learned. Visualization of the output must be there along with other necessary details.**

1. Apply the filters mentioned below on the image attached and analyze their impact. Describe what you found after applying each filter and why certain phenomena occur. ***(Marks:6)***

   - SIFT
   - Bag of Words
   - HOG

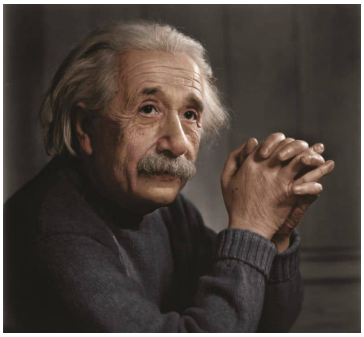

Figure 1: Image of Albert Einstein

Answer:

SHIFT

First I implemented and tested Gaussian blur.

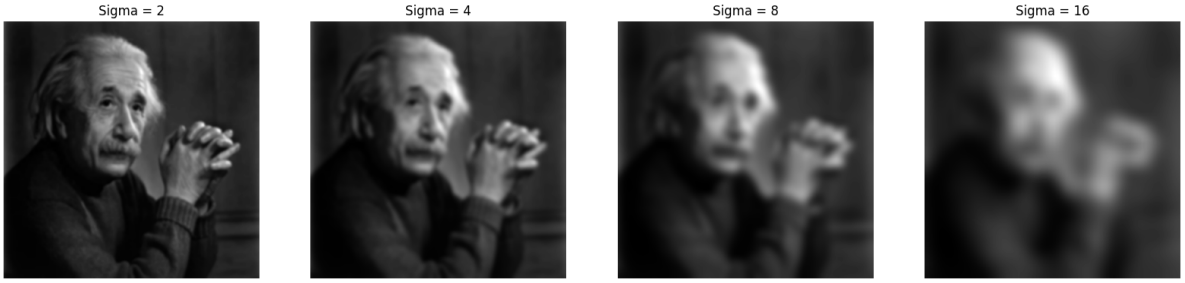

Figure 2: Gaussian blur with different sigma values

Then I computed the **SIFT keypoints** in the image. The keypoints are shown in the image below. The method to compute the keypoints is as follows:

Parameters:

1. *Inital sigma* the base sigma for the Gaussian kernel.
2. *Octaves* the number of octaves in the image. For each octave, the image is downsampled by a factor of 2 and the initial sigma multiplied by two.
3. *Scales* the number of scales in each octave. The number of scales is the number of times the image is blurred with a Gaussian kernel.
4. *Threshold* the threshold for the difference of Gaussian (DoG) image. If the difference of two consecutive scales is greater than the threshold, the pixel is considered a keypoint.

Algorithm:

1. First create a Gaussian pyramid. For each octave, there will be a list of blurred images.
2. Compute the difference of Gaussian (DoG) images. For each octave, there will be a list of DoG images.
3. Compute the keypoints. For each octave, there will be a list of keypoints.

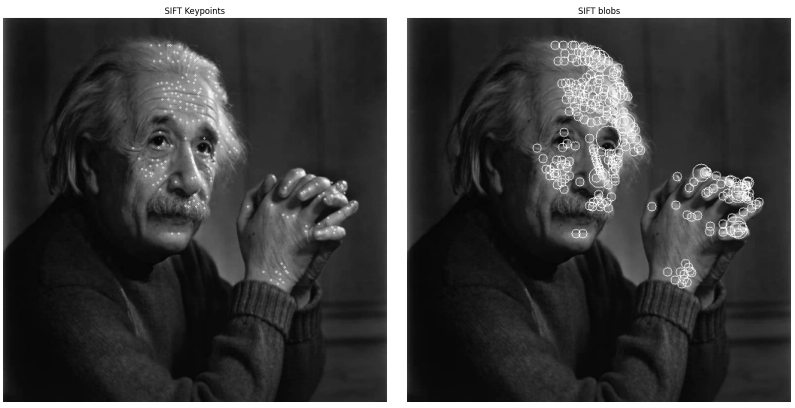

Figure 3: SIFT keypoints and blobs

Bag of Words

For Bag of Words, I have randomly sampled few 8x8 patches from the main image. Now these are my words. To match them and extract the feature vector, I have used l2 norm to find the distance between the patches. The feature vector is the histogram of the patches. The histogram is the frequency of each patch in the image. The feature vector is then can be used to classify the image.

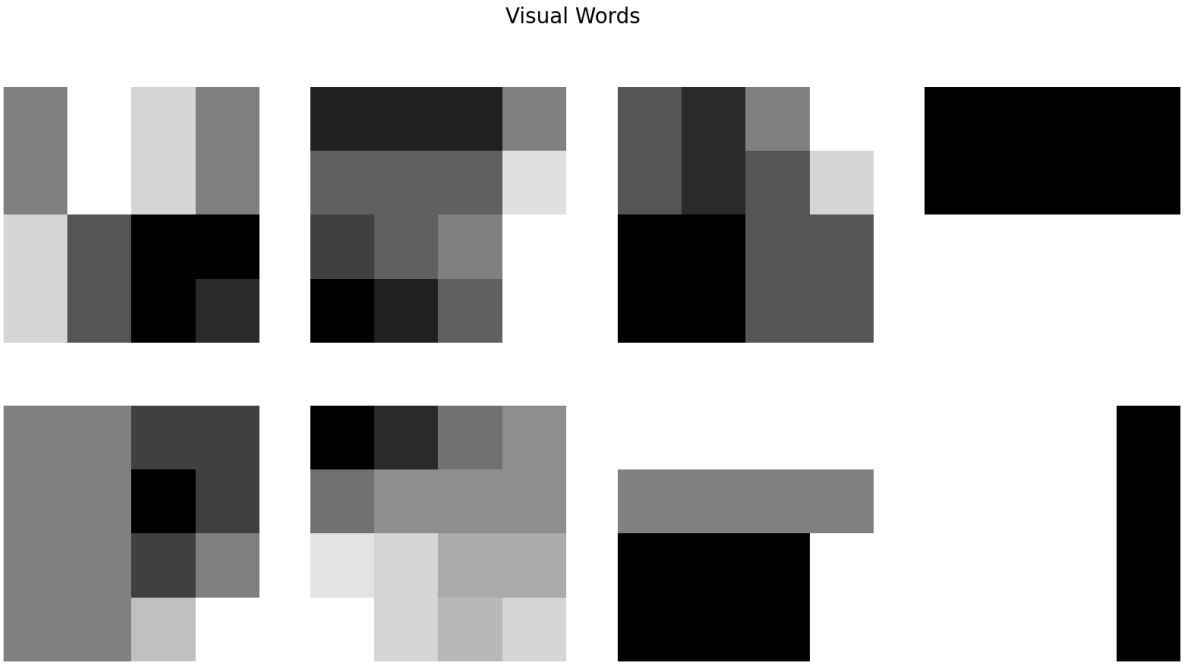

Figure 4: Bag of Words patches

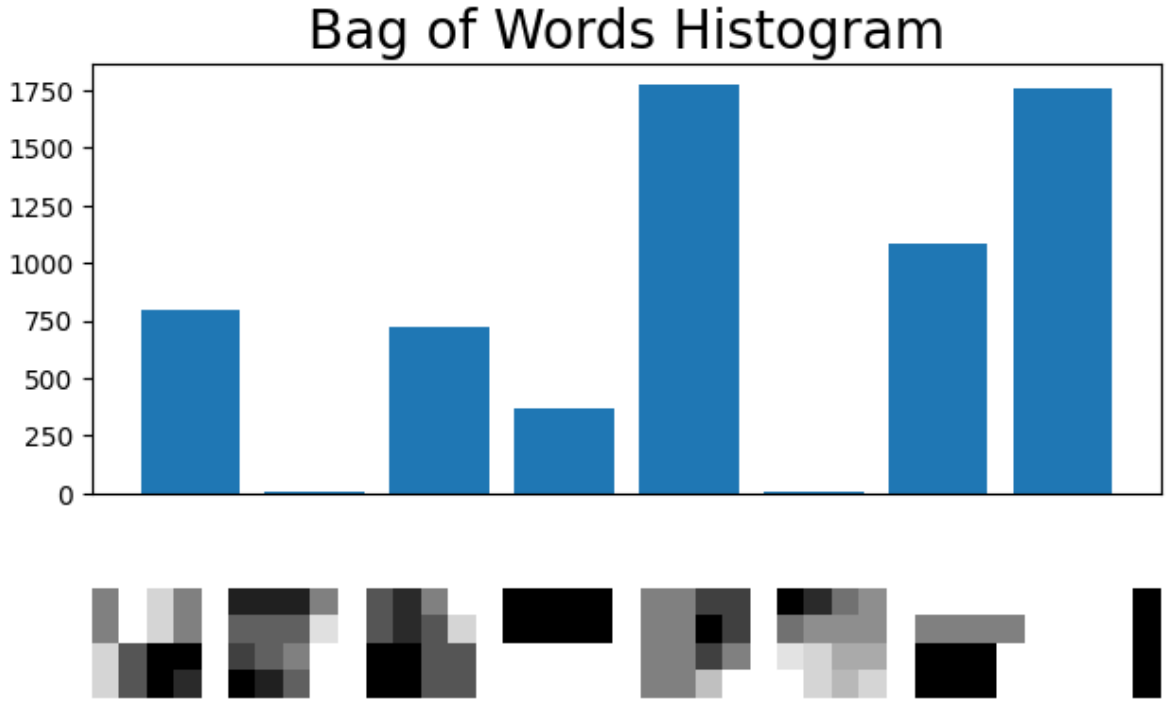

Figure 5: Bag of Words histogram

HOG

For HOG, I have computed the gradient of the image. The gradient is then divided into cells. The cells are then divided into blocks. The histogram of the gradient is computed for each block. The histogram is then normalized. The normalized histogram is then concatenated to form the feature vector.

Figure 6: C-HOG Image

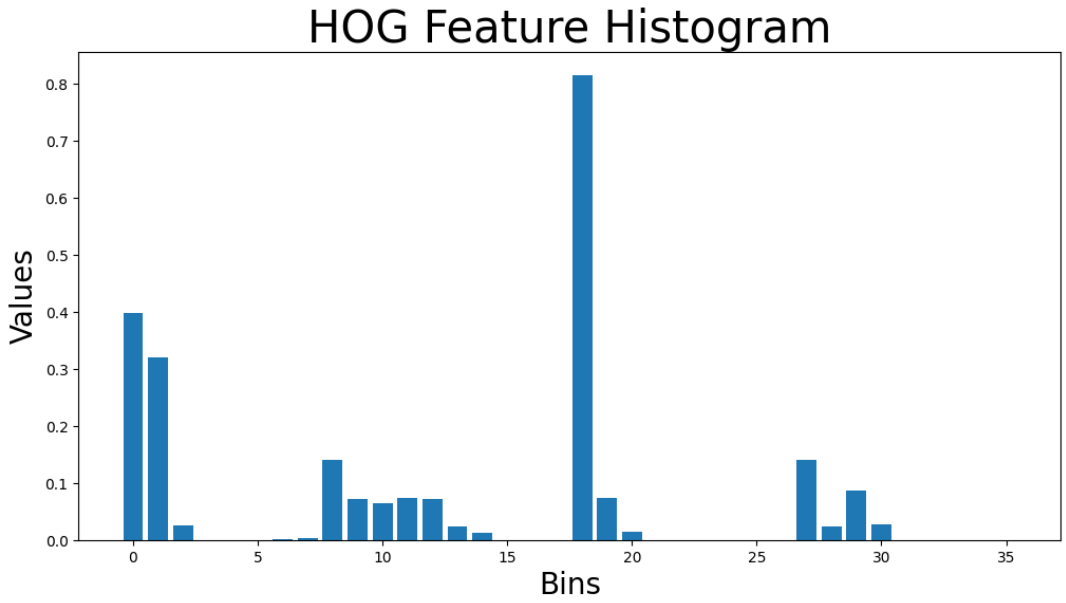

Figure 7: HOG feature

2. Imagine you're monitoring pedestrian movement at a crosswalk. Your task is to track the direction and speed of pedestrians in a video using optical flow analysis. Use OpenCV's built-in video vtest.avi, which simulates real-world pedestrian movement. ***(Marks: 6)***

   - Using the Lucas-Kanade method, track specific points in the video to capture the movement of pedestrians.
   - Visualize the direction of movement using arrows to indicate the flow direction at each point.
   - Provide a brief summary: What patterns do you observe in pedestrian movement? Are there any areas where pedestrians tend to cluster or move faster?

Answer:

Tracking points using Lucas-Kanade method

I have used Lucas Kanade's method to track the points in the video. The points that are to be tracked are resolved by *cv2.goodFeaturesToTrack()* function. The points are then tracked. The points are then visualized using trailing lines to indicate the motion of point.

Figure 8: Optical flow

Visualization through arrows

The direction of the motion is visualized using arrows. The arrows are drawn from the point to the direction of the motion. The length of the arrow is proportional to the speed of the motion.

Figure 9: Optical flow with arrows

Summary

The movement of pedestrian is composed of mostly straight lines. The pedestrians tend to cluster at the center of the crosswalk, at the branching. The pedestrians tend to move faster in the center of the crosswalk. The pedestrians tend to move slower at the edges of the crosswalk. This can be observed in the image attached above. The pedestrians tend to move in the direction of the road.