Name : Aum M. Dhabalia **Date :** 21/10/2024

Roll No. : 21BEC027

Experiment…9 – SURF and HOG feature descriptor

Objective :

- To understand the concepts of SURF and HOG algorithm.
- To compute SURF and HOG features.

Import necessary libraries…

```
'''
Created on 21 October 2024 Mon 2:09:15 pm
'''
import numpy as np
import cv2
import matplotlib.pyplot as plt
import imutils

hog = cv2.HOGDescriptor()
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())
```

Task 1. Write a program to compute the HOG feature descriptors of the image.

```
I = cv2.imread(r"D:\Nirma Files\Computer Vision\Experiments\BO6.jpeg")
Ig = cv2.cvtColor(I,cv2.COLOR_BGR2GRAY)
hog_feature = hog.compute(Ig)
print(f"HOG feature vector shape : {hog_feature.shape}")

plt.imshow(cv2.cvtColor(I,cv2.COLOR_BGR2RGB))
plt.title("HOG feature visualization")
plt.axis("off")
plt.show()
```

Output

```
PS D:\Nirma Files\Computer Vision\Experiments> python exp9.py
HOG feature vector shape : (680400,)
```

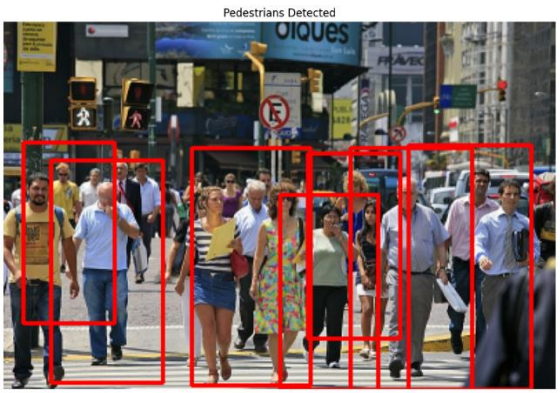
Observation :

- Histogram of Oriented Gradients (HOG) descriptor is effective when fine tuned.
- HOG is dependent on image quality parameters like resolution, lighting, etc.

Task 2. Write a program to detect pedestrians in an image using HOG.

```
hog = cv2.HOGDescriptor()
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())
image = cv2.imread(r"D:\Nirma Files\Computer Vision\Experiments\pedestrian.jpg")
image = imutils.resize(image,width=min(400, image.shape[1]))
(regions, _) = hog.detectMultiScale(image,winStride=(4,4),padding=(4,4),scale=1.05)
for (x, y, w, h) in regions:
        cv2.rectangle(image, (x, y),(x + w, y + h),(0,0,255), 2)
# Showing the output Image
cv2.imshow("Image", image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output



Observation :

- Image on left shows pedestrians walking and image on the right shows the pedestrians detected using HOG descriptor.
- If the image was of poor quality then detection would be hard.

Conclusion:-

As the experiment performed,

- the concept of SURF and HOG algorithms were familiarized.
- Histogram of Oriented Gradient (HOG) was used for detection but detection becomes harder when image quality is poor.
- using HOG pedestrians were detected in an image.

Libraries and functions used are matplotlib, numpy, OpenCV and imutils.