

Judul	PRAKTEK I4-HISTOGRAM OF GRADIENT (HOG)
Deskripsi	Proses feature descriptor dapat menggunakan beberapa metode salah satunya adalah Histogram of Gradient (HOG) yang melakukan proses ekstraksi fitur berdasarkan tepi dan arah orientasi.
Estimasi waktu	15 menit
Prerequisite	<p>Minimal 5 image berwarna</p> <p>Minimal 5 image pejalan kaki</p> <p>Pada percobaan ini kita akan menggunakan library scikit learn yang memiliki turunan khusus untuk citra adalah skimage. Di dalam skimage sudah ada fungsi untuk proses hog.</p> <p>Dalam skimage sudah ada image collection salahsatunya image astronaut, sehingga kita akan memanfaatkan data image dari library skimage tersebut.</p>

Listing Program

```
import matplotlib.pyplot as plt

from skimage.feature import hog
from skimage import data, exposure

image = data.astronaut()

fd, hog_image = hog(image, orientations=8, pixels_per_cell=(16, 16),
                    cells_per_block=(1, 1), visualize=True,
                    multichannel=True)

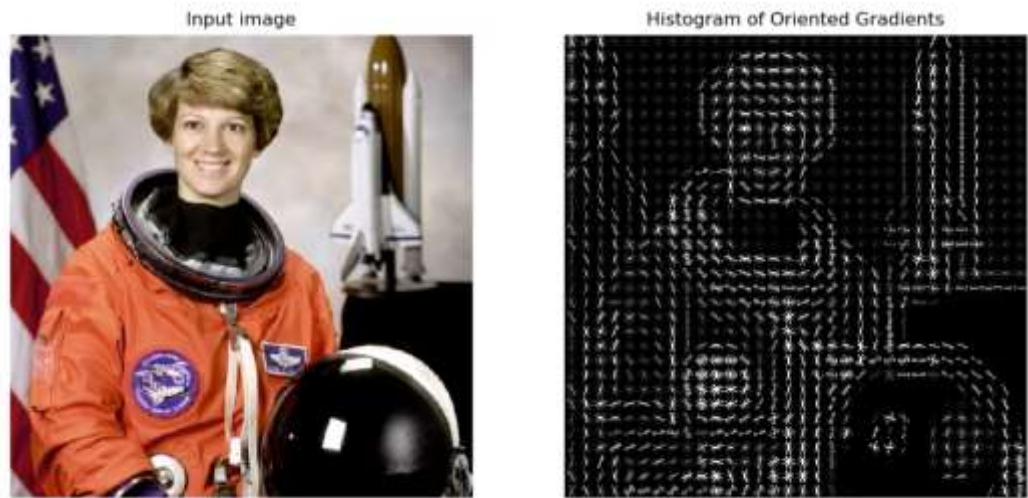
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6), sharex=True,
                              sharey=True)

ax1.axis('off')
ax1.imshow(image, cmap=plt.cm.gray)
ax1.set_title('Input image')

# Rescale histogram for better display
hog_image_rescaled = exposure.rescale_intensity(hog_image, in_range=(0,
10))

ax2.axis('off')
ax2.imshow(hog_image_rescaled, cmap=plt.cm.gray)
ax2.set_title('Histogram of Oriented Gradients')
plt.show()
```

Result



Program HOG mendeteksi pejalan kaki

```
import cv2
import imutils

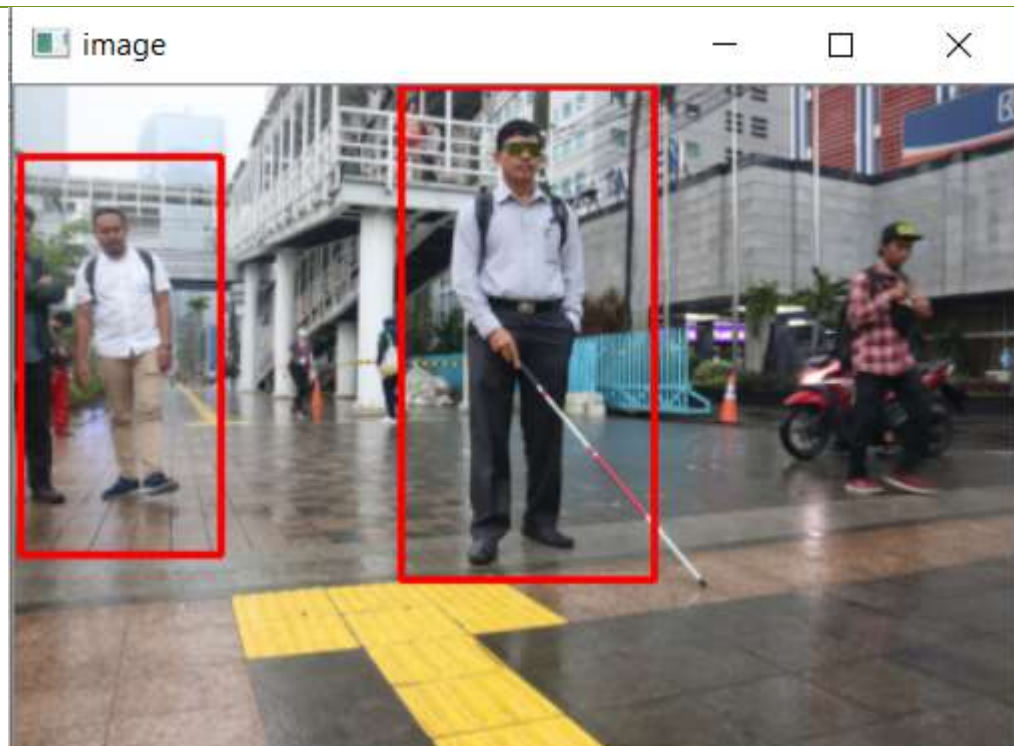
hog=cv2.HOGDescriptor()
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())
img=cv2.imread("pedestrian_2.jpg")

img=imutils.resize(img,width=min(400,img.shape[0]))

(regions, _)=hog.detectMultiScale(img,
winStride=(4,4),padding=(4,4),scale=1.05)
for(x,y,w,h) in regions:
    cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),2)

cv2.imshow("image",img)
cv2.waitKey()
```

Result



Judul	PRAKTEK I5-HAAR CASCADE-FACE AND EYE DETECTION
Deskripsi	Pada praktek ini kita akan melakukan proses deteksi wajah dan mata menggunakan library Haar cascade
Estimasi waktu	15 menit
Prerequisite	Minimal 5 image wajah berwarna

```
import cv2
```

```
face_classifier =  
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

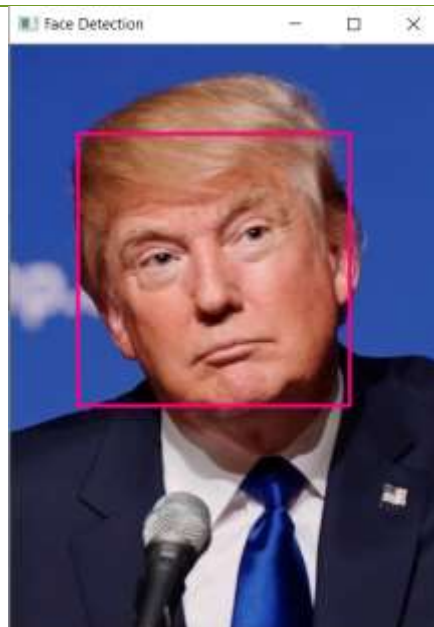
```
image = cv2.imread('Trump.jpg')  
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

```
faces = face_classifier.detectMultiScale(gray, 1.3, 5)
```

Listing Program

```
if faces is ():  
    print("No faces found")  
  
for (x, y, w, h) in faces:  
    cv2.rectangle(image, (x, y), (x + w, y + h), (127, 0, 255), 2)  
    cv2.imshow('Face Detection', image)  
    cv2.waitKey(0)  
  
cv2.destroyAllWindows()
```

Result



Judul	PRAKTEK I6-HAAR CASCADE-PEDESTRIAN DETECTION
Deskripsi	Pada praktek ini kita akan melakukan proses deteksi manusia atau pejalan kaki secara real time menggunakan library Haar cascade
Estimasi waktu	15 menit
Prerequisite	Video pejalan kaki

Listing Program

```
import cv2

# Create our body classifier
body_classifier = cv2.CascadeClassifier('haarcascade_fullbody.xml')

# Initiate video capture for video file
cap = cv2.VideoCapture('walking.avi')

# Loop once video is successfully loaded
while cap.isOpened():

    # Read first frame
    ret, frame = cap.read()
    frame = cv2.resize(frame, None, fx=0.5, fy=0.5,
interpolation=cv2.INTER_LINEAR)

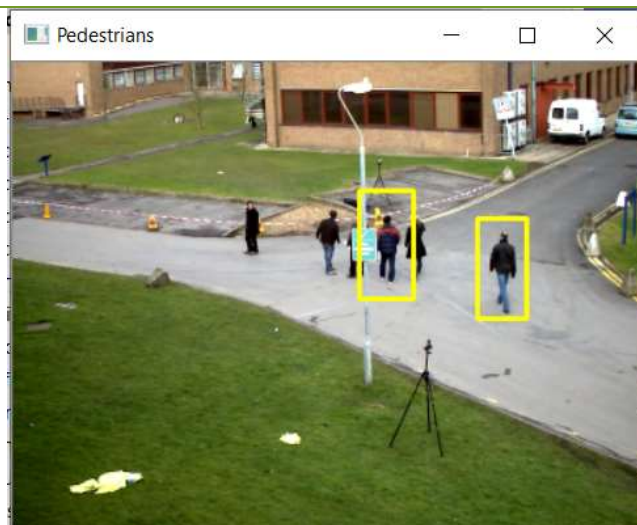
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    # Pass frame to our body classifier
    bodies = body_classifier.detectMultiScale(gray, 1.2, 3)

    # Extract bounding boxes for any bodies identified
    for (x, y, w, h) in bodies:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 255), 2)
        cv2.imshow('Pedestrians', frame)

    if cv2.waitKey(1) == 13: # 13 is the Enter Key
        break

cap.release()
cv2.destroyAllWindows()
```

Result



Judul	PRAKTEK I7-CIRCLE HOUGH TRANSFORM
Deskripsi	Pada praktek ini kita akan melakukan proses analisis bentuk khusus deteksi objek yang berbentuk lingkaran menggunakan Circle Hough Transform
Estimasi waktu	15 menit
Prerequisite	5 Image terutama yang terdapat objek lingkaran

```
import cv2
import numpy as np

img = cv2.imread('houghcircles2.jpg',0)
img = cv2.medianBlur(img,5)
cimg = cv2.cvtColor(img,cv2.COLOR_GRAY2BGR)

circles = cv2.HoughCircles(img,cv2.HOUGH_GRADIENT,1,20,
                           param1=50,param2=50,minRadius=5,maxRadius=0)
```

Listing Program

```
circles = np.uint16(np.around(circles))
for i in circles[0,:]:
    # draw the outer circle
    cv2.circle(cimg,(i[0],i[1]),i[2],(0,255,0),2)
    # draw the center of the circle
    cv2.circle(cimg,(i[0],i[1]),2,(0,0,255),3)

cv2.imshow('detected circles',cimg)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Result

