

# Assgn\_07\_20231024

## 1. HOG

Use cv2.HOGDescriptor()

Load a image (You can choose another image)

- fill the blank area with opencv python codes and
- get the result images as shown below
  - ※ You can use other images but do the same image processing and get the same style of the answer image.

filename and type : yourname\_assgn\_06.pdf

Due Date : 30 Oct 0900 a.m. (Monday 0900 a.m. 1 day before the class)

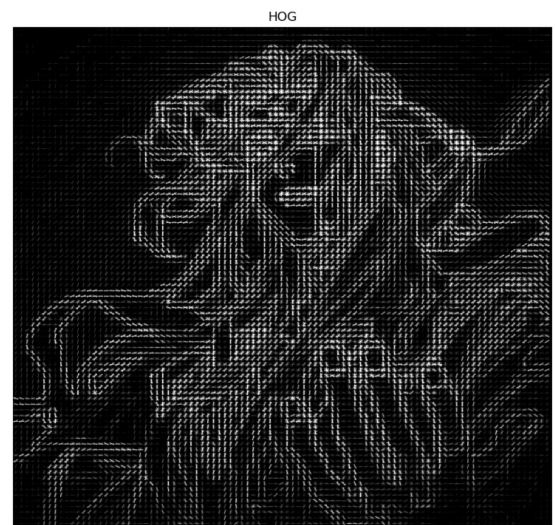
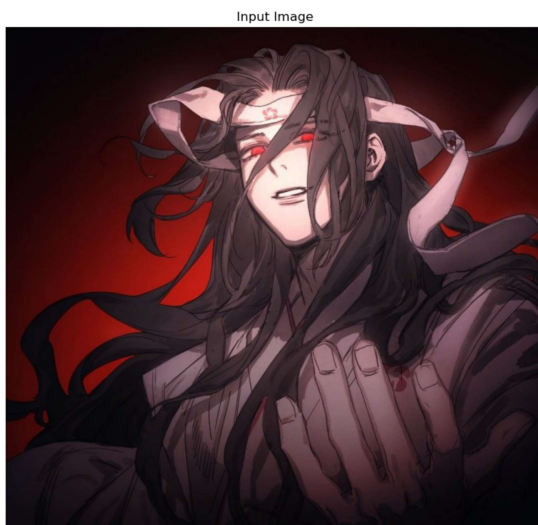
```
In [23]: import matplotlib.pyplot as plt
from skimage.feature import hog
from skimage import data, feature, exposure
import cv2
from matplotlib.pyplot import figure

image = cv2.imread('./images/practice_img/geto.jpg')
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # convert bgr to rgb

fd, hog_image = hog(image, orientations=9, pixels_per_cell=(10, 10),
                    cells_per_block=(2, 2), visualize=True, multichannel=True)
hog_image_rescaled = exposure.rescale_intensity(hog_image, in_range=(0, 4)) #

figure(figsize=(20, 20), dpi=100)

plt.subplot(121),plt.imshow(image),plt.title('Input Image'),plt.axis('off')
plt.subplot(122),plt.imshow(hog_image_rescaled,'gray'),plt.title('HOG'),plt.axis('off')
plt.show()
```



## 2. Photo Sketching

Use `cv2.divide(gray, inverted_blurred_image, scale)`

```
In [38]: import cv2
import numpy as np
from matplotlib import pyplot as plt

src = cv2.imread("./images/practice_img/geto.jpg")
gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY) # gray scale of the image

inverted_gray_image = cv2.bitwise_not(gray) # invert the gray scale image
blurred_image = cv2.GaussianBlur(inverted_gray_image, (31, 31), 0) # blur the i
inverted_blurred_image = cv2.bitwise_not(blurred_image) # invert the blurred in
pencil_sketch_image = cv2.divide(gray, inverted_blurred_image, scale = 256.0) #

# cv2.imshow('Original Image', src)
# cv2.imshow('sketch_image', pencil_sketch_image)
# # cv2.imwrite('./results/me_sketch.png',pencil_sketch_image )
# cv2.waitKey(0)
# cv2.destroyAllWindows()

fig, axs = plt.subplots(1, 2, figsize=(15,10))
axs[0].imshow(cv2.cvtColor(src, cv2.COLOR_BGR2RGB)), axs[0].axis('off'), axs[0]
axs[1].imshow(cv2.cvtColor(pencil_sketch_image, cv2.COLOR_GRAY2RGB)), axs[1].ax
plt.show()
```

