Avinash - 1102244010

Assgn_07_20231024

1. HOG

Use cv2.HOGDescriptor()

Load a image (You can choose another image)

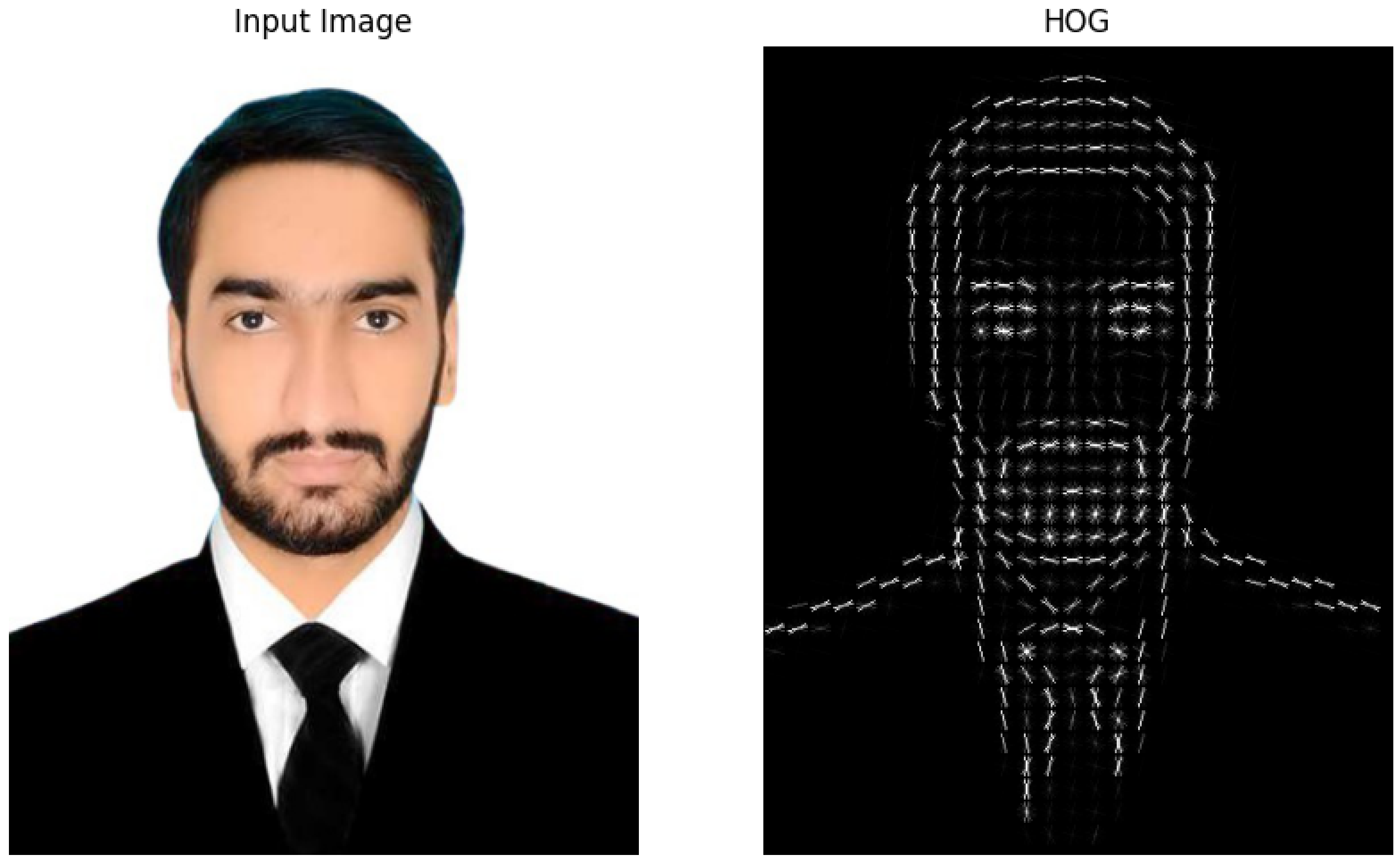
- fill the blank area with opencv python codes and
- get the result images as shown below
  ※ You can use other images but do the same image processing and get the same style of the answer image.

filename and type : yourname_assgn_06.pdf
Due Date : 30 Oct 0900 a.m. (Monday 0900 a.m. 1 day before the class)

In [1]:

```
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('./myPic.jpg')
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # convert bgr to rgb

fd, hog_image = hog(image, orientations=8, pixels_per_cell=(16, 16), cells_per_block=(1, 1), visualize=True, multichannel=True) # create hog img
hog_image_rescaled = exposure.rescale_intensity(hog_image, in_range=(0, 10)) # rescaled

figure(figsize=(10, 10), 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()
```

```
C:\Users\Avinash Kumar\anaconda3\envs\cv\lib\site-packages\ipykernel_launcher.py:10: FutureWarning: `multichannel` is a deprecated argument name for `hog`. It will be removed i
n version 1.0. Please use `channel_axis` instead.
  # Remove the CWD from sys.path while we load stuff.
```

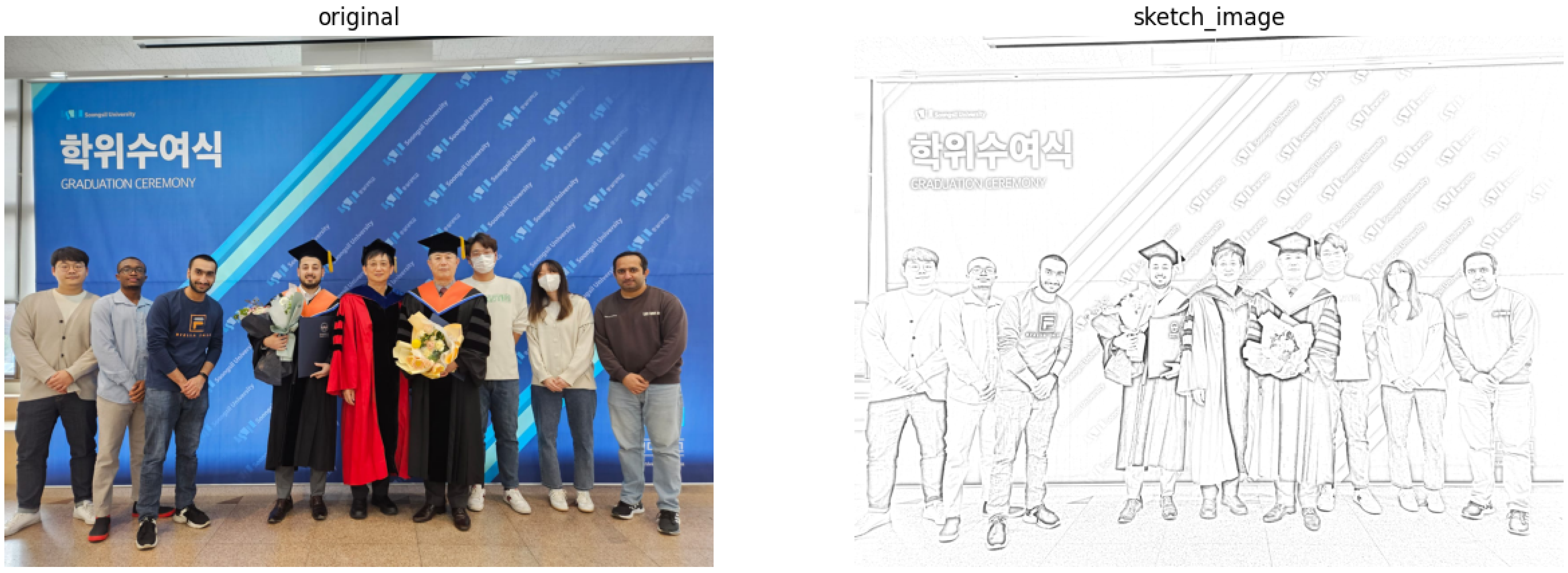


2. Photo Sketching

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

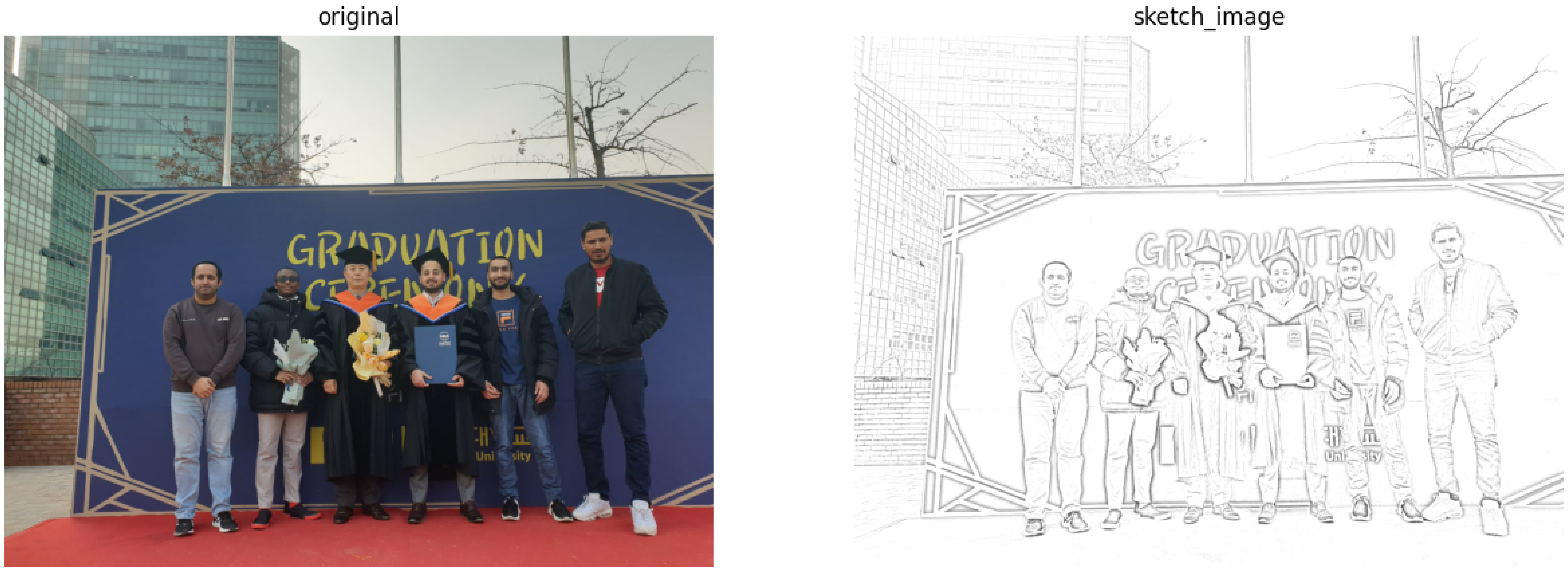
In [2]:

```python
import cv2
import numpy as np
from matplotlib import pyplot as plt
src = cv2.imread("./pic1.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, (21, 21), sigmaX=0, sigmaY=0) # blur the image by gaussian blur
inverted_blurred_image = cv2.bitwise_not(blurred_image) # invert the blurred image
pencil_sketch_image = cv2.divide(gray, inverted_blurred_image, scale=256.0) # create the pencil sketch image
cv2.imshow('Original Image', src)
cv2.imshow('sketch_image', pencil_sketch_image)
# cv2.imwrite('./results/me_sketch.png',pencil_sketch_image )
cv2.waitKey(1000)
cv2.destroyAllWindows()
# Plot using matplotlib
fig, axs = plt.subplots(1, 2, figsize=(15,10))
axs[0].imshow(cv2.cvtColor(src, cv2.COLOR_BGR2RGB)), axs[0].axis('off'), axs[0].set_title('original')
axs[1].imshow(pencil_sketch_image, cmap='gray'), axs[1].axis('off'), axs[1].set_title('sketch_image')
plt.show()
```



In [3]:

```python
import cv2
import numpy as np
from matplotlib import pyplot as plt
src = cv2.imread("./pic2.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, (21, 21), sigmaX=0, sigmaY=0) # blur the image by gaussian blur
inverted_blurred_image = cv2.bitwise_not(blurred_image) # invert the blurred image
pencil_sketch_image = cv2.divide(gray, inverted_blurred_image, scale=256.0) # create the pencil sketch image
cv2.imshow('Original Image', src)
cv2.imshow('sketch_image', pencil_sketch_image)
# cv2.imwrite('./results/me_sketch.png',pencil_sketch_image )
cv2.waitKey(1000)
cv2.destroyAllWindows()
# Plot using matplotlib
fig, axs = plt.subplots(1, 2, figsize=(15,10))
axs[0].imshow(cv2.cvtColor(src, cv2.COLOR_BGR2RGB)), axs[0].axis('off'), axs[0].set_title('original')
axs[1].imshow(pencil_sketch_image, cmap='gray'), axs[1].axis('off'), axs[1].set_title('sketch_image')
plt.show()
```



In [ ]: