Computer Vision Assignment 9

import cv2

import numpy as np

from matplotlib import pyplot as plt

from google.colab.patches import cv2\_imshow

import imutils

img = cv2.imread('blackdot.jpg')

cv2\_imshow(img)

image = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

#Set our filtering parameters

#Initialize parameter setting using cv2.simpleblobdetector

params = cv2.SimpleBlobDetector\_Params()

#Set area filtering parameters

params.filterByArea = True

params.minArea = 100

#Set circularity filtering parameter

params.filterByCircularity = True

params.minConvexity = 0.2

#Set inertia filtering parameters

params.filterByInertia = True

params.minInertiaRatio = 0.01

#Create a detector with the parameters

detector = cv2.SimpleBlobDetector\_create(params)

#Detect blobs

keypoints = detector.detect(image)

#Draw blobs on our image as red circles

blank = np.zeros((1,1))

blobs = cv2.drawKeypoints(image, keypoints, blank, (0,0,255),

                          cv2.DRAW\_MATCHES\_FLAGS\_DRAW\_RICH\_KEYPOINTS)

number\_of\_blobs = len(keypoints)

text = "Number of circular Blobs: " + str(len(keypoints))

cv2.putText(blobs, text, (20, 500),

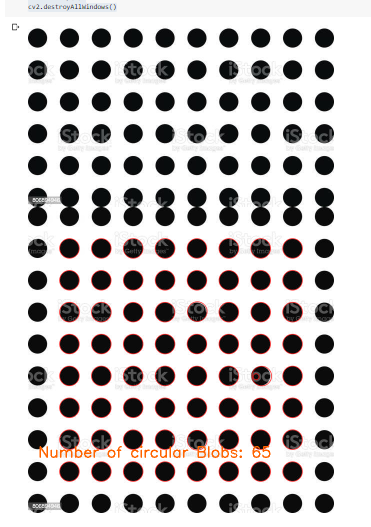
            cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0,100,255),2)

#show blobs "Filtering circular blobs only"

cv2\_imshow(blobs)

cv2.waitKey(0)

cv2.destroyAllWindows()



#histogram gradient

from skimage import feature

image = cv2.imread('dandelions.jfif')

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

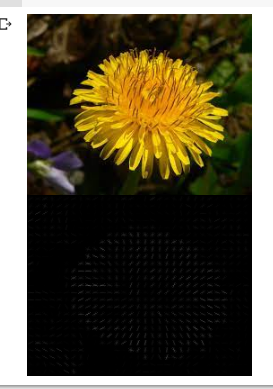
(hog, hog\_image) = feature.hog(gray, orientations = 9,

                               pixels\_per\_cell=(8,8), cells\_per\_block=(2,2),

                               visualize = True)

cv2\_imshow(image)

cv2\_imshow(hog\_image)



#reading image

img1 = cv2.imread('effifle.jfif')

gray1 = cv2.cvtColor(img1, cv2.COLOR\_BGR2GRAY)

#keypoints

sift = cv2.xfeatures2d.SIFT\_create()

keypoints\_1, descriptors\_1 = sift.detectAndCompute(img1, None)

img\_1 = cv2.drawKeypoints(gray1, keypoints\_1, img1)

plt.imshow(img\_1)

