

IOT – SMART FARMING

ASSIGNMENT 3: PYTHON CODE FOR BLINKING LED AND TRAFFIC LIGHT

INDUMATHI R

RENO:610819205009

CODING:

```
import cv2
import sys
import os

def find_traffic_sign(main_images_path,selected_images_path, threshold=0.1):
    MIN_MATCH_COUNT = 10
    # Initialize SIFT detector
    sift = cv2.xfeatures2d.SIFT_create(contrastThreshold=threshold)
    for selected_image in os.listdir(selected_images_path):
        img1 = cv2.imread(os.path.join(selected_images_path,selected_image))
        match_check=False
        for main_image in os.listdir(main_images_path):
            img2 = cv2.imread(os.path.join(main_images_path,main_image))
            kp1, des1 = sift.detectAndCompute(img1, None)
            kp2, des2 = sift.detectAndCompute(img2, None)
            FLANN_INDEX_KDTREE = 0
            index_params = dict(algorithm=FLANN_INDEX_KDTREE, trees=5)
            search_params = dict(checks=50)
            flann = cv2.FlannBasedMatcher(index_params, search_params)
            matches = flann.knnMatch(des1, des2, k=2)

            good = []
            for m, n in matches:
```

```
if m.distance < 0.1 * n.distance:
    good.append(m)
if len(good) > MIN_MATCH_COUNT:
    match_check=True
    img1 = cv2.putText(img1, main_image, (0,50), cv2.FONT_HERSHEY_SIMPLEX, 0.80, (0,255,255),
    print("Match found: "+str(len(good))+ " common keypoints are found between "
    +selected_image+" and "+main_image)
    break
if match_check==False :
    print("Match not found")

cv2.imshow('img1', img1)
cv2.waitKey(0)
if __name__ == '__main__':
    find_traffic_sign(*sys.argv[1:])
```