**Name: Abhishek Goyal**

**Entry number: 2023AIB2073**

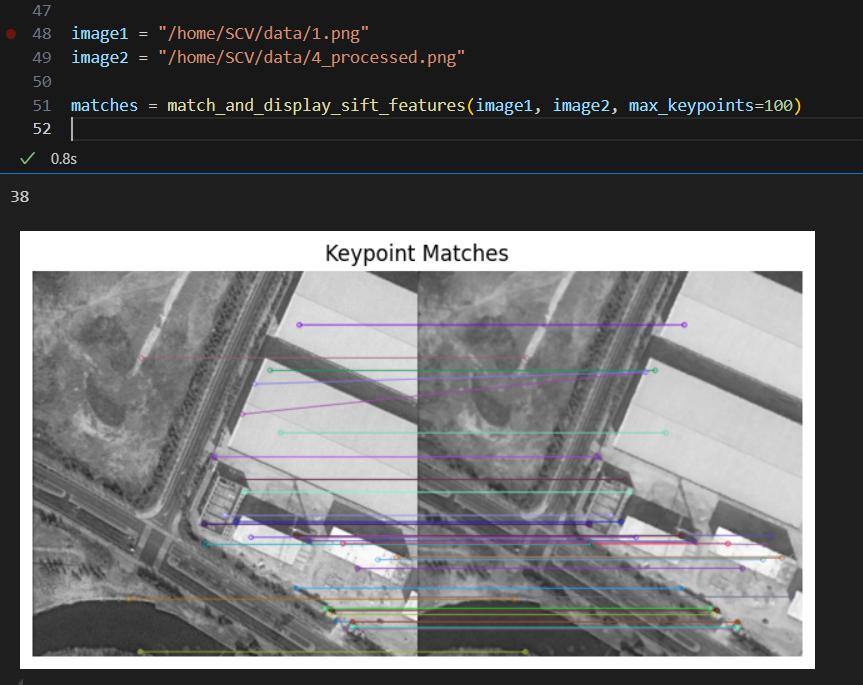
**Approach**

*Preprocessing:* To handle noise in images, I identified the distribution of pixel values. I found values 0 and 255 have higher frequency than others which is a sign of salt pepper noise. So, I applied a median filter on images 4,5 & 6. I applied a 3\*3 filter twice.

A graph of a blue channel

Description automatically generated with medium confidence

*Similarity:* I got SIFT descriptors for all the images. Then I got the best 100 key points ordered based on response value for each image. So, each image has just 100 best key points. I matched these key points of image\_1 with image\_2 using their descriptors. After matching I filtered those key points which match with a L2 distance between them is less than 100 units.



I defined,

**similarity =** **No. of key points which match / Total no. of key points of an image** (here 100)

**Result:**

Similarity between image 1 and image 2: **0**

Similarity between image 1 and image 4: **0.38**

Similarity between image 2 and image 3: **0.12**

Similarity between image 1 and image 5: **0**

Similarity between image 1 and image 6: **0**

**Python code link:** [ass\_1.ipynb](https://csciitd-my.sharepoint.com/:u:/g/personal/aib232073_iitd_ac_in/EVa4GDsB2VVEoS7xQ1t03uQBUSGtEK_XoUstpk1ThbdXBA?e=jd7HOo)