INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

Statistical Methods in Artificial Intelligence (SMAI)

Submitted To

Dr. Anoop M. Namboodiri

Project – Scale Invariant Feature Transform Detection

Reference link to the original paper → Distinctive Image features from Scale-Invariant keypoints

Submitted by Team no. 61:

Guneesh Vats – 2021122007
Abdul Hadi – 2021900006
Venkata Ramana – 2020102043
Priyansh Khunger – 2020101056

Contents of the Report

Introduction and Objective

➢ We are going to implement SIFT algorithm that will transform Image data to scale invariant key-point Co-ordinates and find an effective way of describing feature points across multiple Images.
➢ SIFT is the shortcomings of Harris corner Detector that is invariant to rotation but not to scale and illumination which prompted the development of SIFT.
➢ SIFT image features are partially invariant to changes in illumination and viewpoint.

Theory of SIFT

4 steps of SIFT feature extraction and image matching algorithm:

1. Scale-space extrema detection
2. Keypoint localization
3. Orientation Assignment
4. Keypoint descriptor and Keypoint matching

➢ Scale-Space extrema detection

- Key points whose descriptors are invariant to image scaling must first be detected by identifying locations and scales which are identifiable from different view of the same object.
- To accomplish these features which are stable across multiple scales must be found using a continuous function of scale.

- It has been shown that the function of scale must be based on the Gaussian Blur operator, which allows for images to be scaled without introducing any false details.
- The scale function is defined as the convolution of the Gaussian blur Operator and the Image:

$$\boxed{L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y)}$$

L : is the blurred image.
I : is the original image.
(x, y) : are the location coordinates.
σ : is the scale value.
G : is the Gaussian Blur operator given by:

$$\boxed{G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}}$$

- The **Difference of Gaussians Technique** is then sued to enhance features and find **stable keypoints in the defined scale-space**. The Difference of Gaussians is given as the difference of two blurred images, with one image having a scale k time the other.

$$\boxed{\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}}$$

- Using DoG extrema are detected by comparing each point with its 26 neighbours.

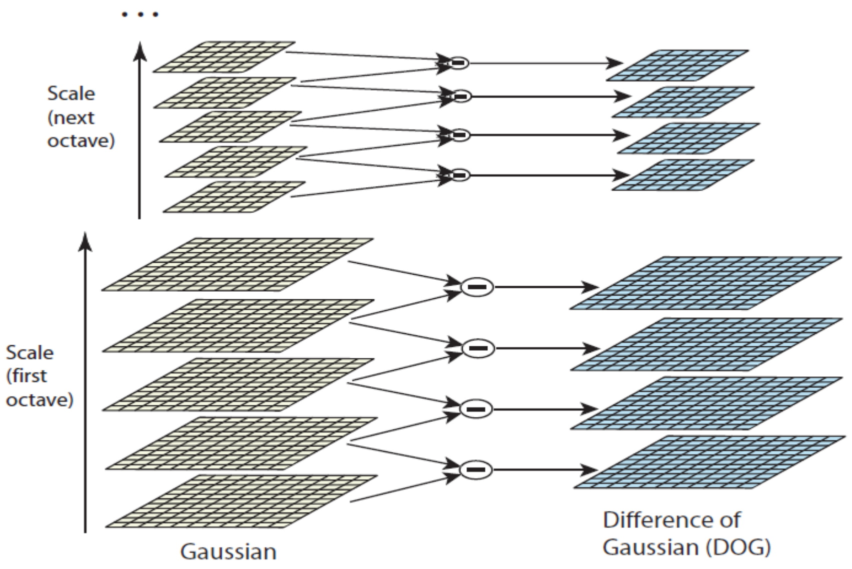
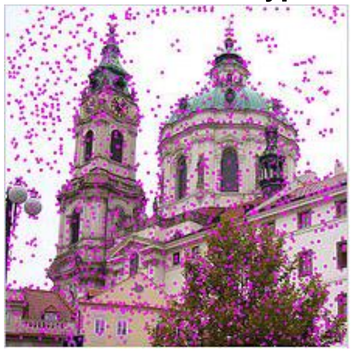


Figure (Lowe, 2004)

➢ Key point Localization

- After scale-space extrema detection, this will produce too many keypoint candidates.
- The next step is to perform a detailed fit to the nearby data or accurate location, scale, and ratio of principal curvatures.

Initial detection of keypoints →

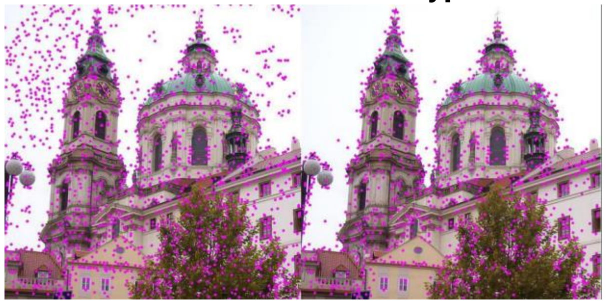


- For each candidate keypoint, interpolation of nearby data is used to accurately determine its position. Where the interpolation is done using the quadratic Taylor expansion of the DoG scale-space function.

$$\boxed{D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}}$$

- **Removal of low-contrast keypoints is done by computing the value of the second-order Taylor expansion at the offset.** If this value is less than 0.03, the value is discarded. Otherwise, it is kept with the final scale-space location: original location + offset.

Removal of low-contrast keypoints →



$$\boxed{|D_{extremal}| < 0.03}$$

- **Eliminating edge responses:** The DoG function will have strong responses along edges, even if the candidate keypoint is not robust to small amounts of noise.
- Hence keypoints that have poorly determined position and yet high edge responses must be eliminated.

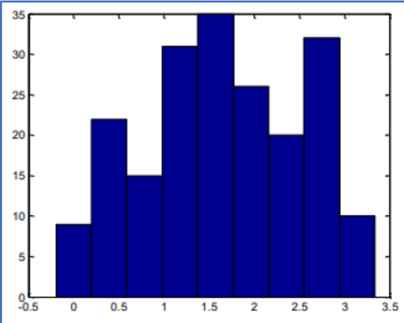
➢ Orientation Assignment

- Each keypoint is assigned one or more orientations based on local image gradient directions.
- Important step in achieving invariance to rotation.
- Since keypoint descriptor can be represented relative to this orientation and this achieve Invariance to rotation.
- Compute the gradient magnitudes and orientations using pixel difference

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

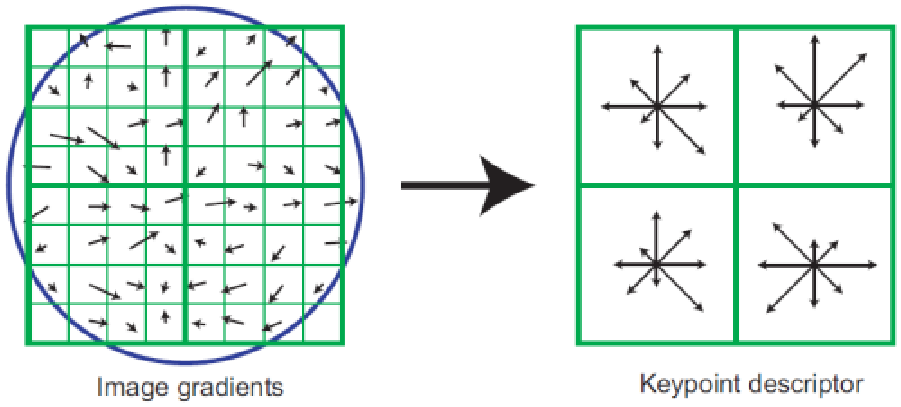
- Create histogram of local gradient directions at a selected scale.
- Assign orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale orientation)



Histogram of gradient orientation – the bin-counts are weighted by gradient magnitudes and a Gaussian weighting function. Usually, 36 bins are chosen for the orientation.

➢ Keypoint Descriptor

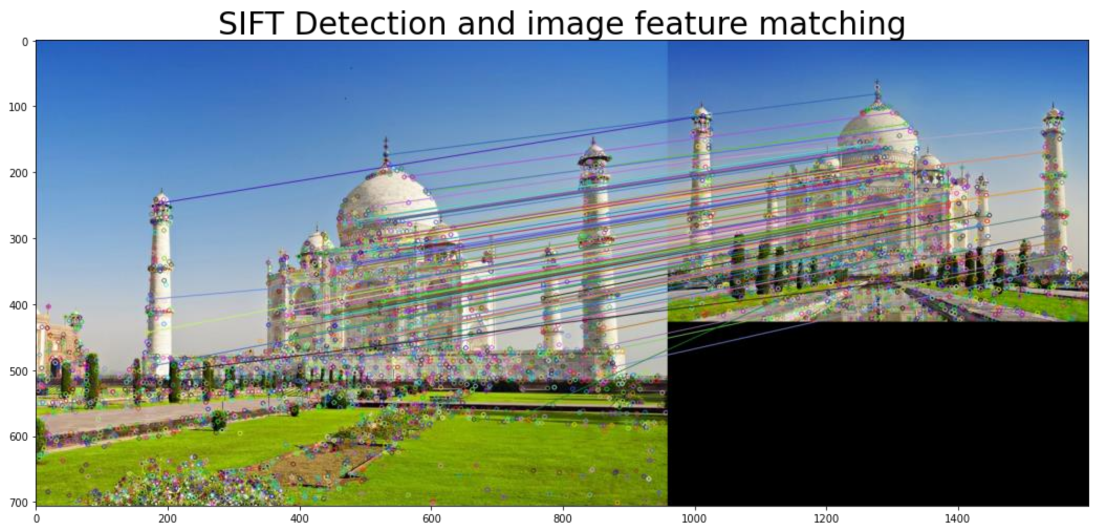
- This is the final stage of the SIFT algorithm before the keypoint process. It consists of normalized 128-dimensional vector.
- Highly distinctive and partially invariant to the variations such as illumination, 3D viewpoint, etc.
- Performed on the image closet in scale to the keypoints scale.
- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below), compute an orientation histogram for each cell.
- 16 cell * 8 orientations = 128-dimensional descriptor.



- The image gradient magnitudes are weighted by a Gaussian function with **σ** equal to one half the width of the descriptor.
- A threshold of 0.2 is applied in order to reduce the effects of non-linear illumination.
- The feature matching accuracy is above 50% for viewpoint changes of up to 50 degrees.

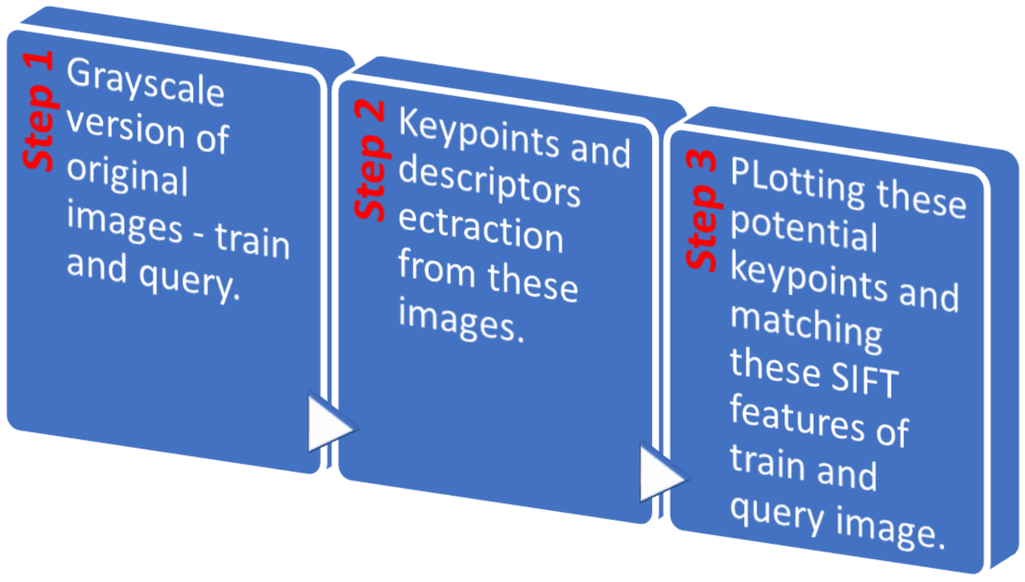
➢ Keypoint Matching

- Applying Nearest neighbour matching of local image descriptors.
- We will implement best-bin-first approximation for selecting point matches in the train and query image.
- After that we will do the Affine Hough transform based evidence accumulation of object models.
- Following is an example of which we will show the implementation later in the report of feature matching:



Implementation Strategy

General Flow of the SIFT feature extraction and matching algorithm in code:


Step 1
Grayscale version of original images - train and query.
Step 2
Keypoints and descriptors ectraction from these images.
Step 3
PLotting these potential keypoints and matching these SIFT features of train and query image.

In 1st part of the code we are just extracting the features from 4 x celebrity image taken from google images.

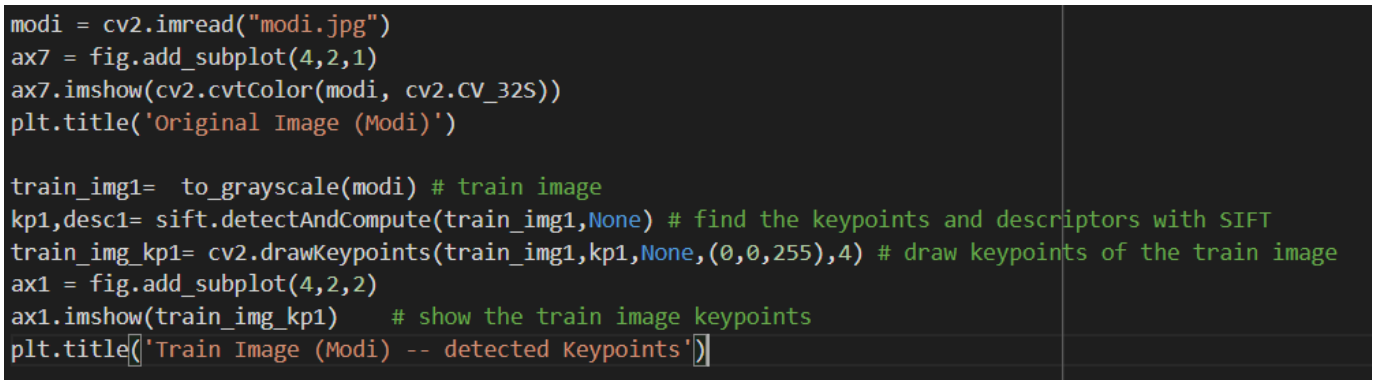
In the 2nd Part we have taken train and query image of 2 monuments. Applied the above algorithm and obtained the matching feature point and plotted them on the images.

- For SIFT algorithm we need train and query images which have different viewpoint, scale and illumination and we chose the images accordingly.

Code Implementation and Results

We took 4 images of 4 different public figures - MS Dhoni, Amitabh Bachchan, Narendra Modi and Anushka Sharma and extracted keypoints discriptors from their original image according to the code flow explained in above section. This is well implemented in code through opencv library commands explained below:

Through imread we read the coloured images in grayscale, We earlier in the code created an object of the function SIFT of cv2 library and then use it to extract keypoints and distractors - using sift.detectAndCompute command.

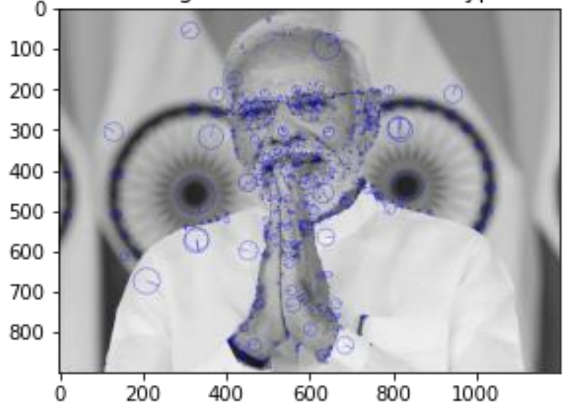


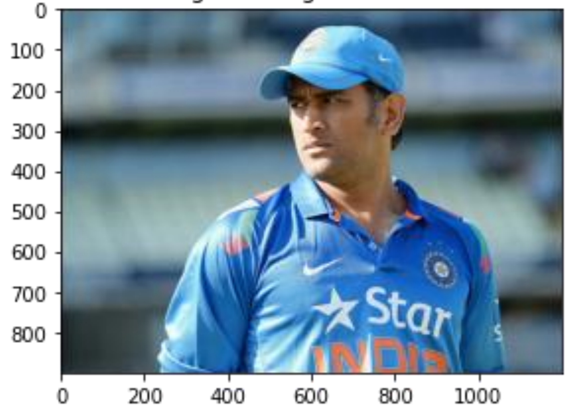
```
modi = cv2.imread("modi.jpg")
ax7 = fig.add_subplot(4,2,1)
ax7.imshow(cv2.cvtColor(modi, cv2.CV_32S))
plt.title('Original Image (Modi)')

train_img1=  to_grayscale(modi) # train image
kp1,desc1= sift.detectAndCompute(train_img1,None) # find the keypoints and descriptors with SIFT
train_img_kp1= cv2.drawKeypoints(train_img1,kp1,None,(0,0,255),4) # draw keypoints of the train image
ax1 = fig.add_subplot(4,2,2)
ax1.imshow(train_img_kp1)    # show the train image keypoints
plt.title('Train Image (Modi) -- detected Keypoints')
```

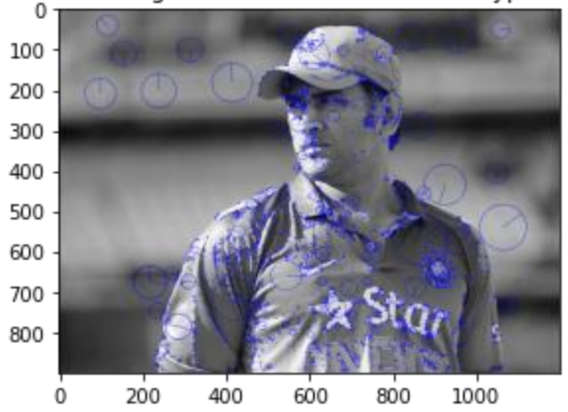
Similarly we have implemented the code for the rest of the 4 images which you can find in our submitted code file.

Observation after running this code is shown on the following page:

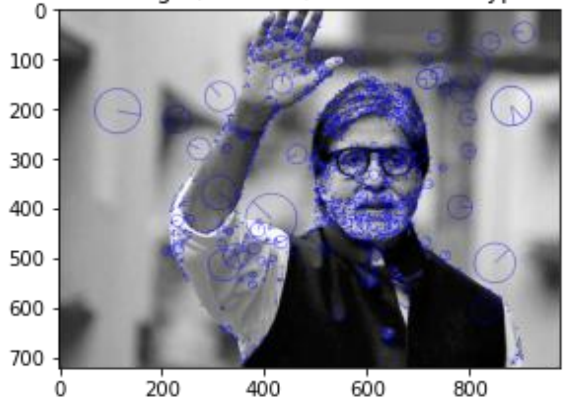
Original Image (Modi)

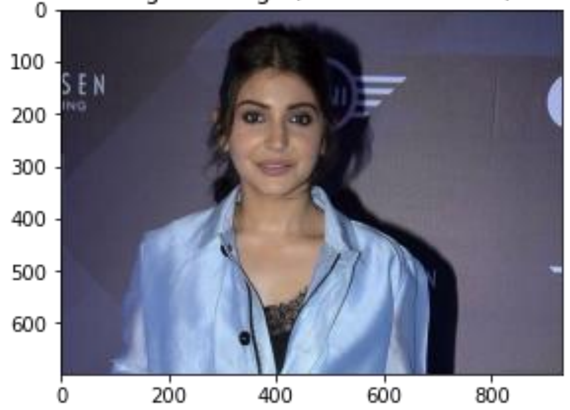

Train Image (Modi) -- detected Keypoints

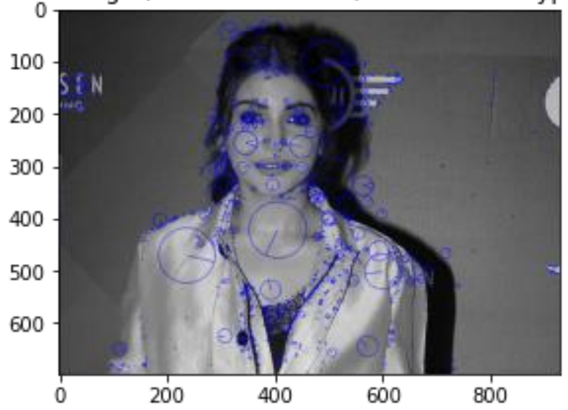

Original Image (MS Dhoni)


Train Image (MS Dhoni) -- detected Keypoints

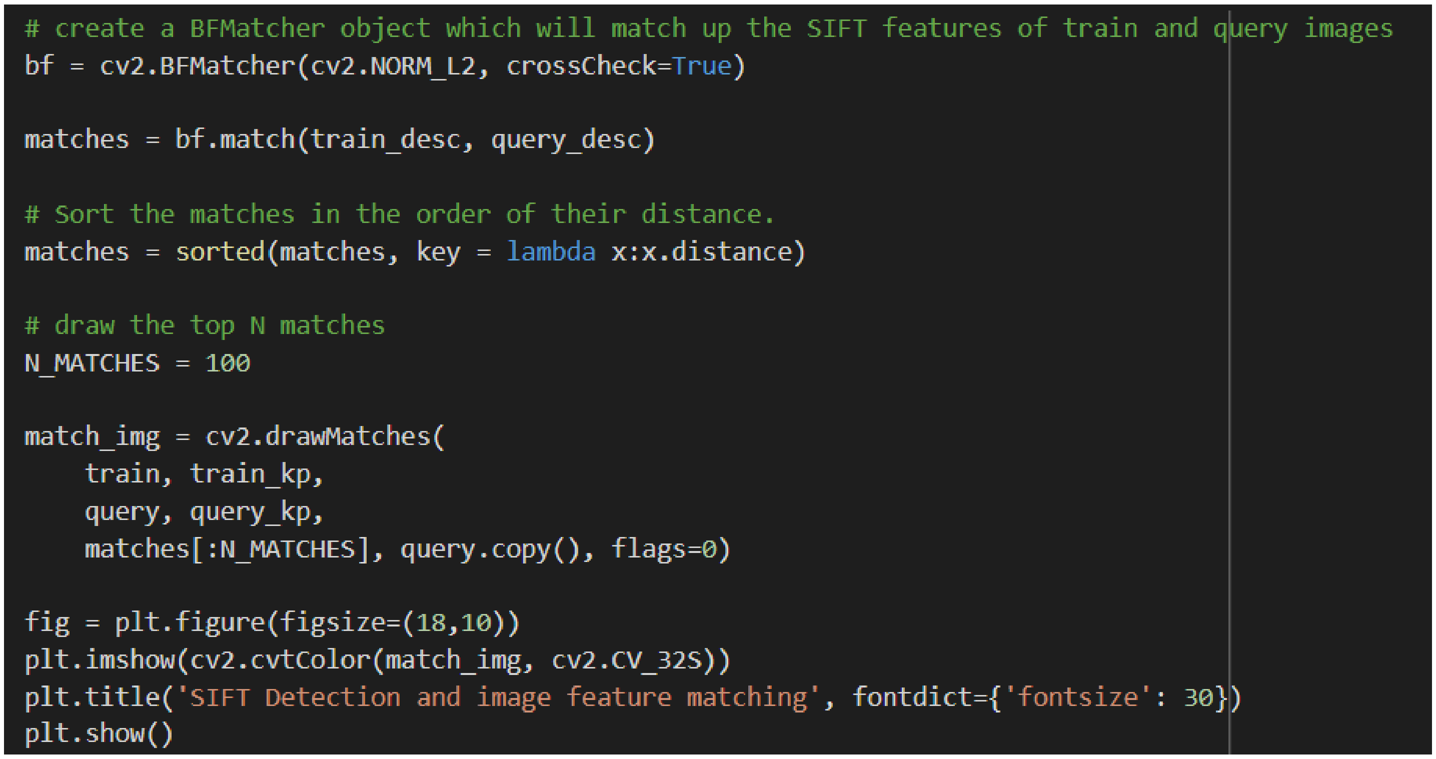
Original Image (Amitabh)


Train Image (Amitabh) -- detected Keypoints

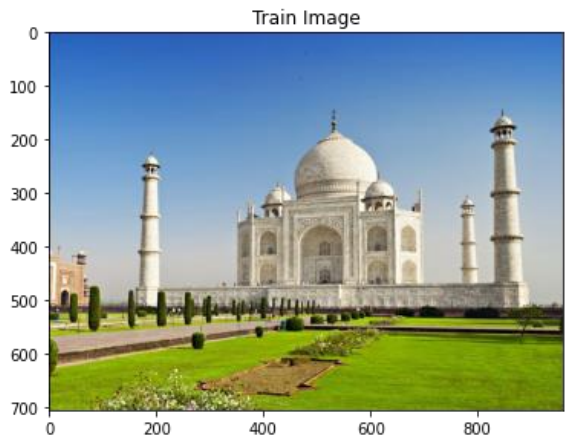

Original Image (Anushka Sharma)

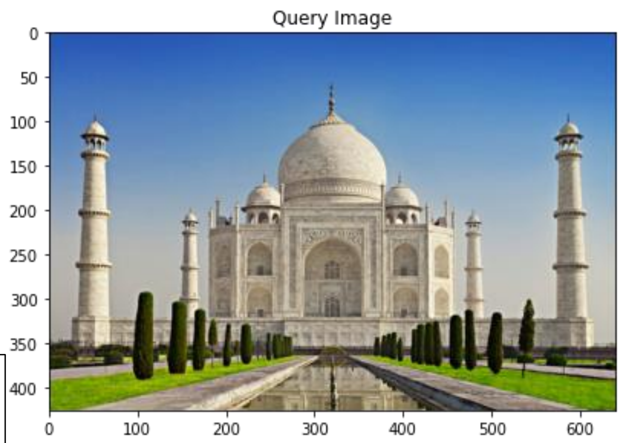

Train Image (Anushka Sharma) -- detected Keypoints

- Now we are going to take 2 images of **Taj Mahal** – train image and query image and the same for **Humayun's tomb** to implement the algorithm and the results obtained are shown below:
- Till the step of extracting features process, code remains similar but for the matching part as you can see following this code explanation, we are creating a BFmatcher object which will match up the SIFT features of train and query images. After that it sort the matches in the order of their distance and finally draw top N matches of the keypoint matching and plot the calculated matched features of both the images as shown in the below presented code.

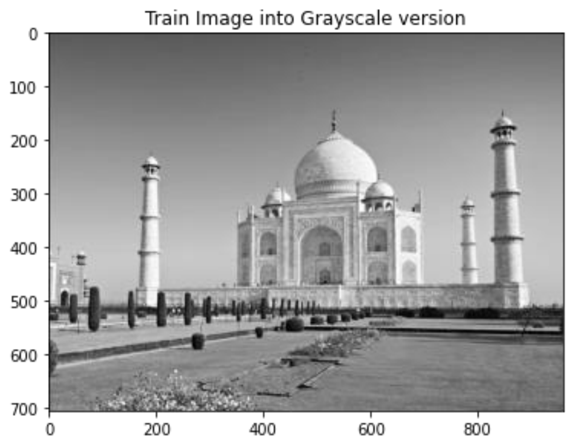


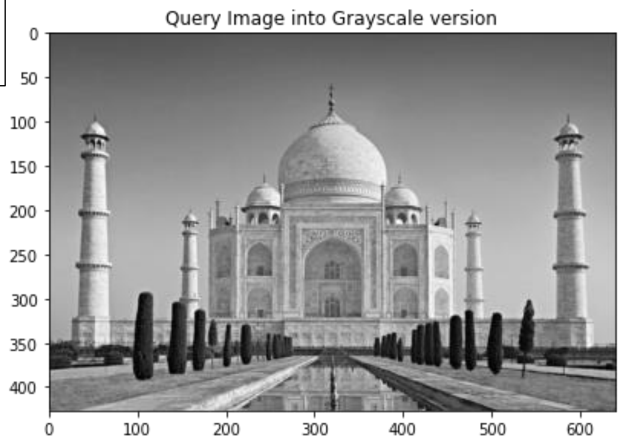
```
# create a BFMatcher object which will match up the SIFT features of train and query images
bf = cv2.BFMatcher(cv2.NORM_L2, crossCheck=True)

matches = bf.match(train_desc, query_desc)

# Sort the matches in the order of their distance.
matches = sorted(matches, key = lambda x:x.distance)

# draw the top N matches
N_MATCHES = 100

match_img = cv2.drawMatches(
    train, train_kp,
    query, query_kp,
    matches[:N_MATCHES], query.copy(), flags=0)

fig = plt.figure(figsize=(18,10))
plt.imshow(cv2.cvtColor(match_img, cv2.CV_32S))
plt.title('SIFT Detection and image feature matching', fontdict={'fontsize': 30})
plt.show()
```



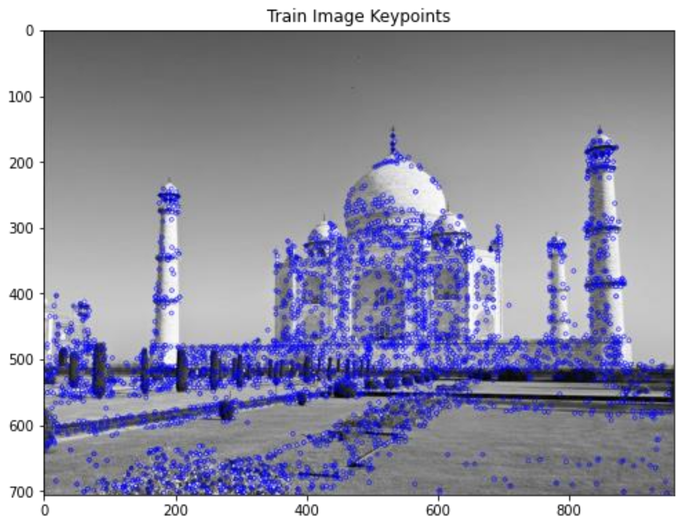


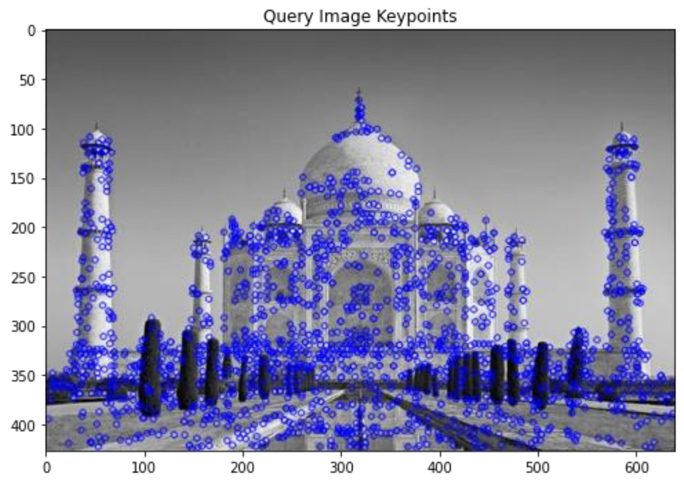
Input image to grayscale version



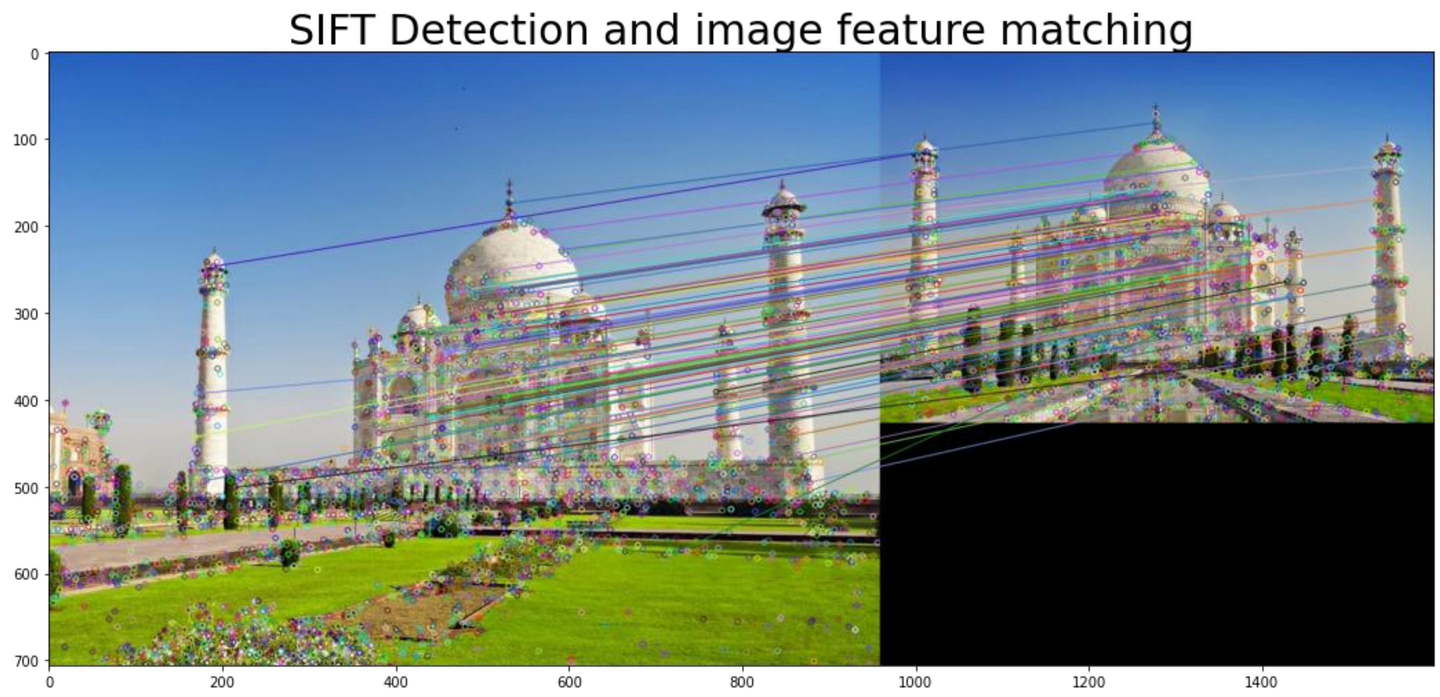


Extracting keypoints and descriptors

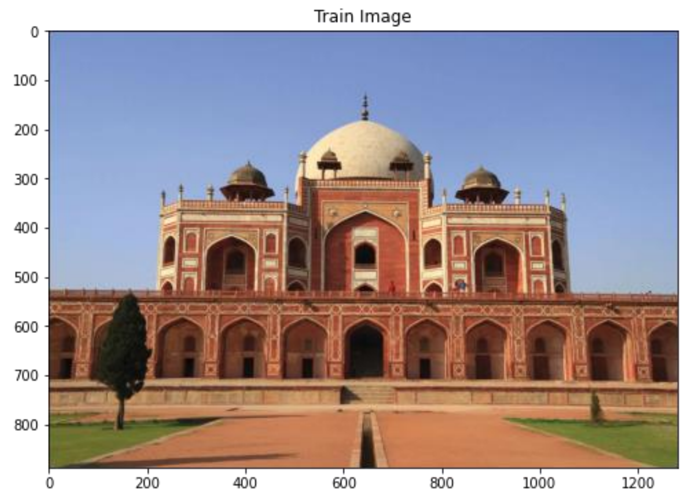


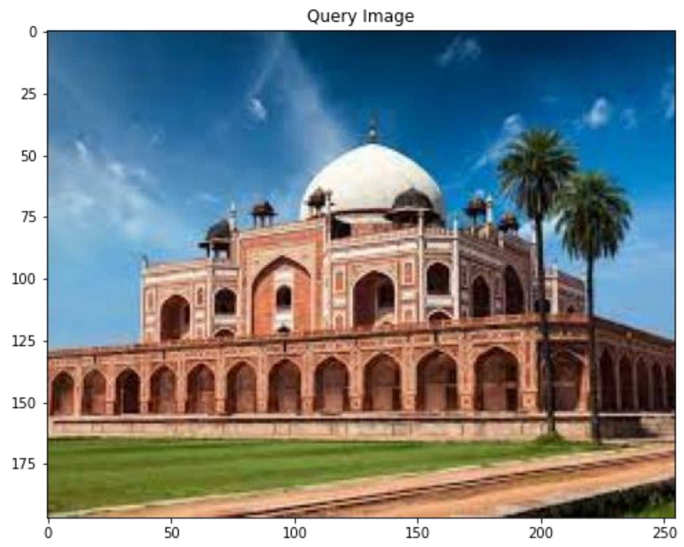


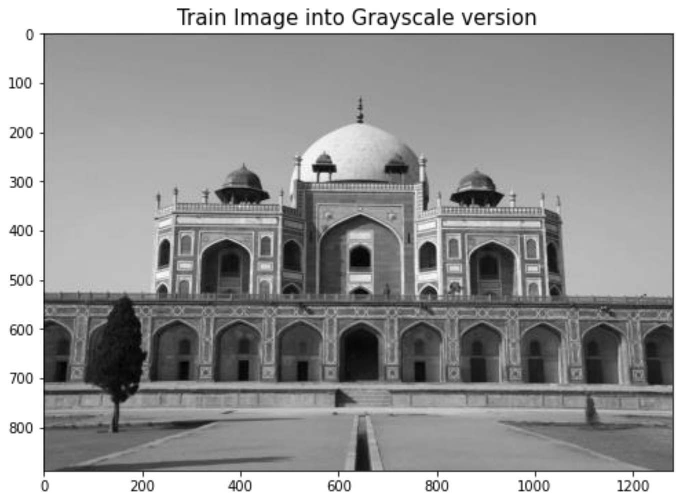
Finally, we are matching both the SIFT features of train and query images

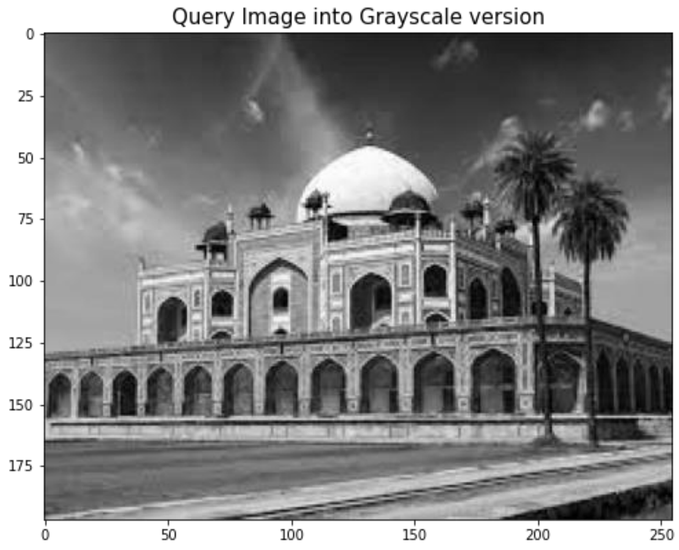


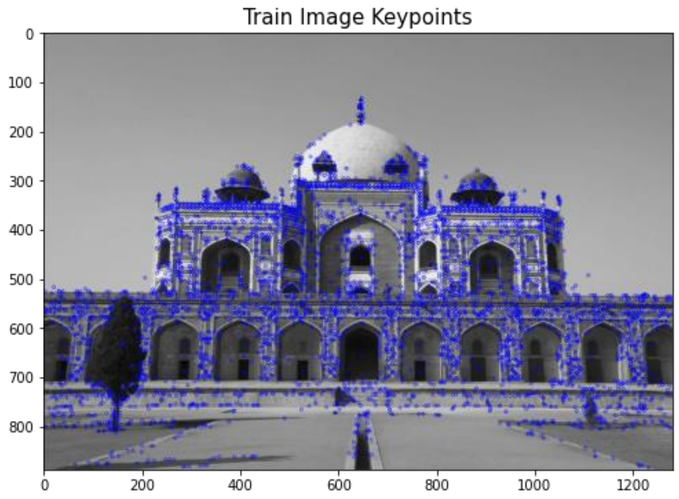
Now for the images of **Humayun's tomb** we went through the similar procedure and obtained the following results:

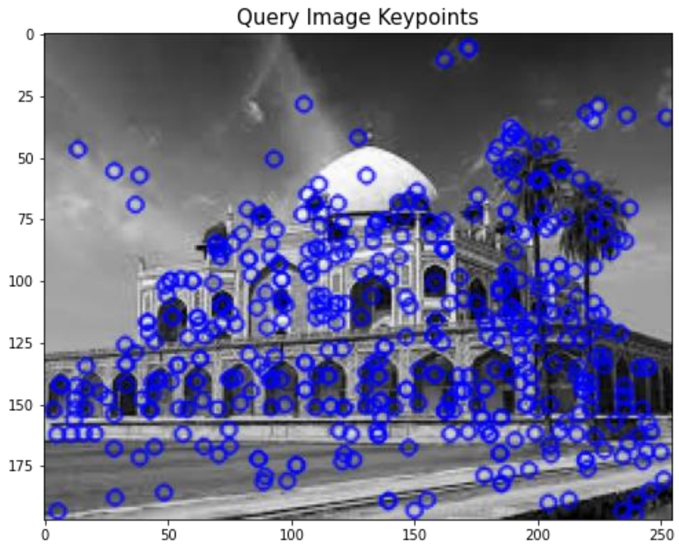


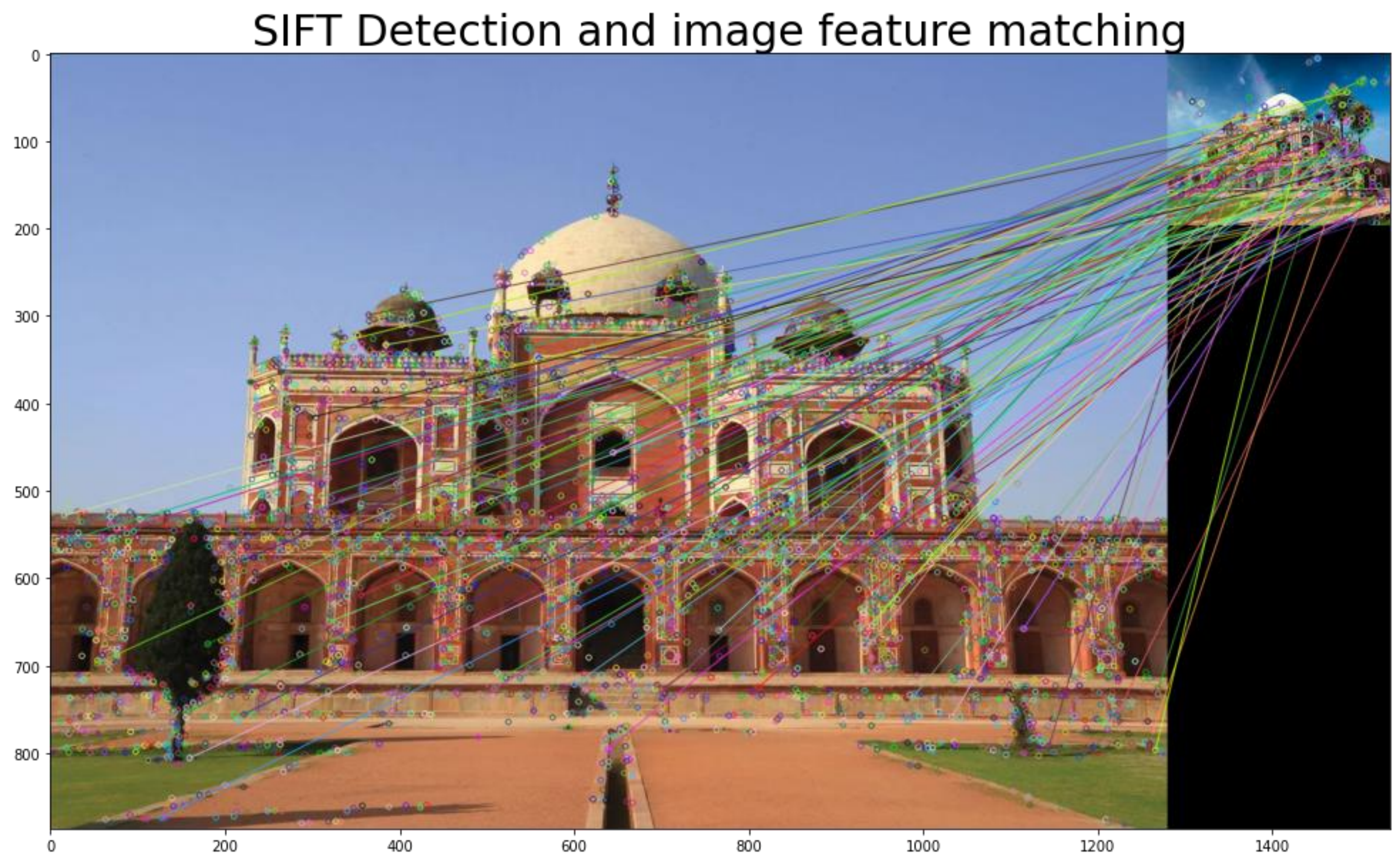



Train Image into Grayscale version
0
100
200
300
400
500
600
700
800
0
200
400
600
800
1000
1200


Query Image into Grayscale version
0
25
50
75
100
125
150
175
0
50
100
150
200
250


Train Image Keypoints
0
100
200
300
400
500
600
700
800
0
200
400
600
800
1000
1200


Query Image Keypoints
0
25
50
75
100
125
150
175
0
50
100
150
200
250



Here we extracted the keypoints and distractors after converting them to grayscale and then matches these SIFT features of train and query images.

Conclusion and Applications

We have successfully implemented SIFT feature extraction and done image matching with 2 examples of Taj mahal and Humayun's tomb with good accuracy. Initially we showed the keypoint extraction over celebrity images. We then modified and used that code structure to do feature matching between train and query images.

Applications of SIFT include object recognition, image stitching, 3D modeling, gesture recognition, video tracking, individual identification of wildlife and match moving

Link of the ipynb file implementing the above code:

https://colab.research.google.com/drive/1fRb1UaI0NoZPkIFcRqG399HLT8DuVOYO?usp=sharing

Contributions

1. **Guneesh Vats –** Designed the code for implementing SIFT, Literature review, Theory of SIFT, prepared observations and results and the final project report.
2. **Abdul Hadi –** Literature Review, Theory of SIFT, Implementation strategy, code structure, final project report.
3. **Venkata Ramana –** References and Applications
4. **Priyansh Khunger –** References and Conclusion

References

1. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110. https://doi.org/10.1023/b:visi.0000029664.99615.94

2. Cong Geng and X. Jiang, "*Face recognition using sift features*," 2009 16th IEEE International Conference on Image Processing (ICIP), 2009, pp. 3313-3316, doi: 10.1109/ICIP.2009.5413956.

3. Cong Geng and X. Jiang, "SIFT features for face recognition," 2009 2nd IEEE International Conference on Computer Science and Information Technology, 2009, pp. 598-602, doi: 10.1109/ICCSIT.2009.5234877.

4. X. Wangming, W. Jin, L. Xinhai, Z. Lei and S. Gang, "Application of Image SIFT Features to the Context of CBIR," 2008 International Conference on Computer Science and Software Engineering, 2008, pp. 552-555, doi: 10.1109/CSSE.2008.1230.

Reference 1 is the major portion contributing for our implementation.