E9 246 Advanced Image Processing

Assignment 01

Name: **Niladri Dutta**

Course: **MTech AI**

SR No.: **23112**

1. PCA-SIFT

a) Scaling

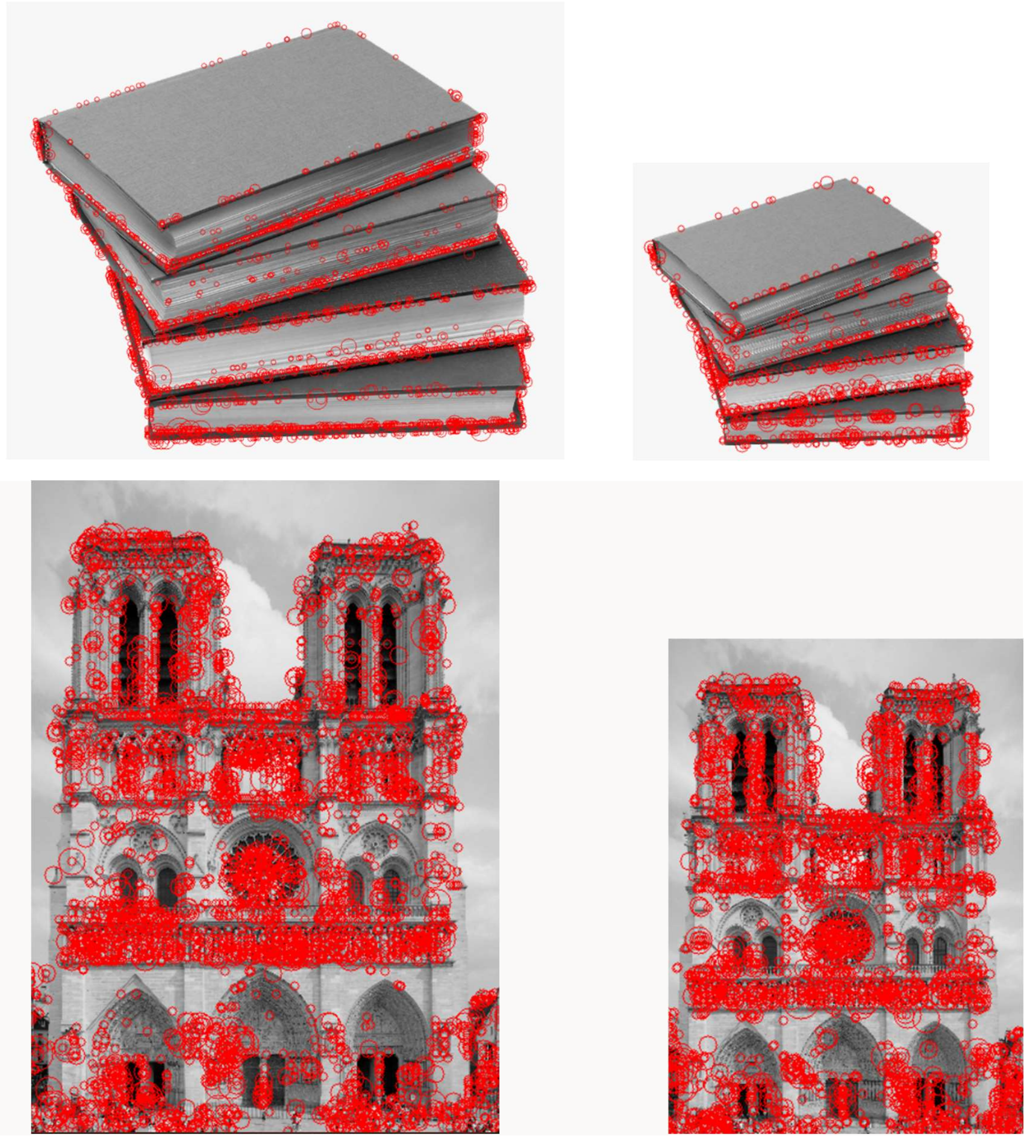


Fig1. Key points in Image-1 and Image-2 and corresponding subsampled version.

In the subsampled version of given image there is a drop in the image quality due to data loss hence there will be fewer key points detected, intuitively. But key points are located at same locations as evident from given images. Also as shown in images above, in the subsampled images more circles of larger radii are present indicating the detection at a higher octave.

Number of detected key points:

Image 1- Original:15786 Subsampled (by factor 2):7252

Image 2- Original: 2448 Subsampled (by factor 2): 743

b) Rotation

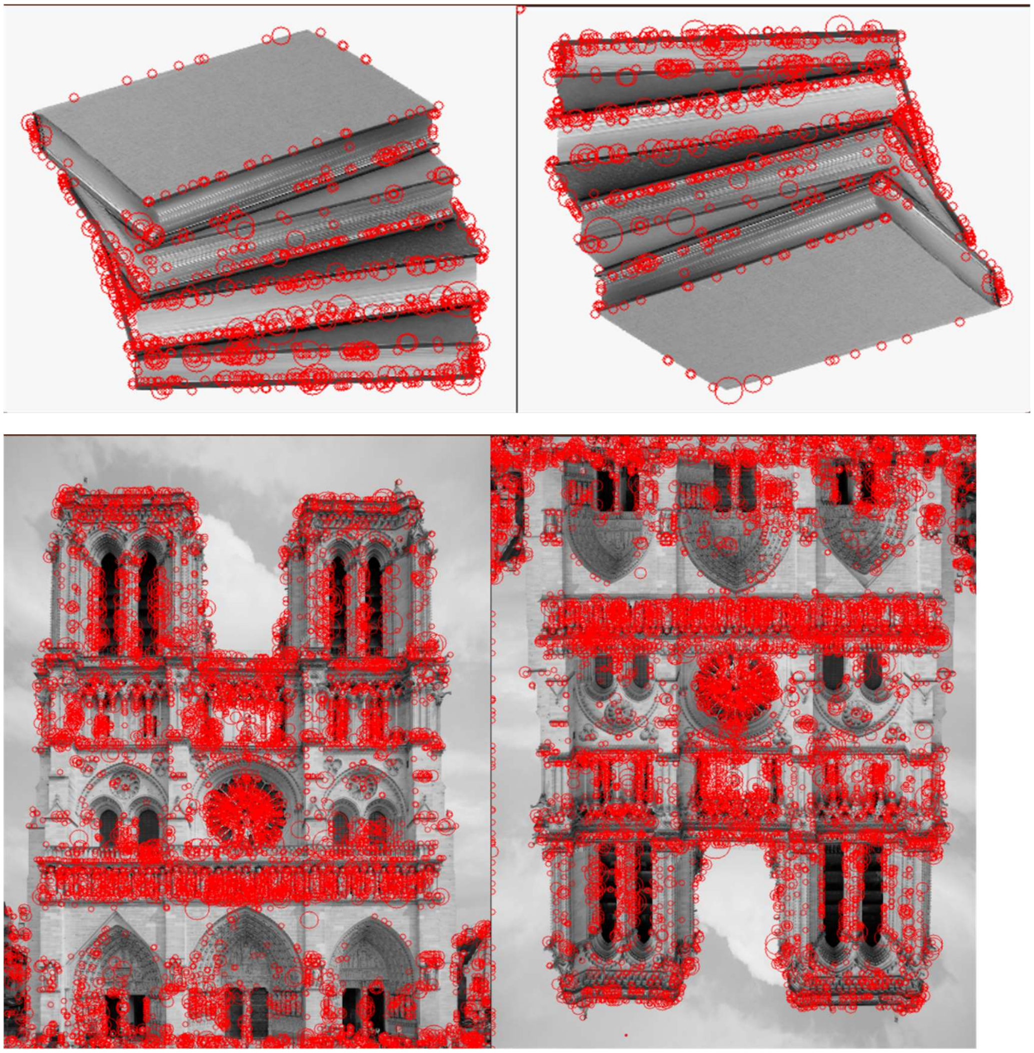


Fig2. Key points in Image-1 and Image-2 and corresponding 180 degrees rotated version.

It can be observed from the images that the key points in the original and their rotated versions are located at similar locations i.e., the key points stand invariant to rotation operation as can be seen from their counts too, as given below.

Number of detected key points:

Image 1- Original:15786 Rotated:15651

Image 2- Original: 2448 Rotated: 2444

As evident from the image, key points are mostly located on the corners, edges and also the carvings on the building (image 1) which are representative of the features of the image. The count doesn't change significantly after rotation because the image is effectively the same with exactly the same features preserved, just rotated.

c. Gaussian Blur

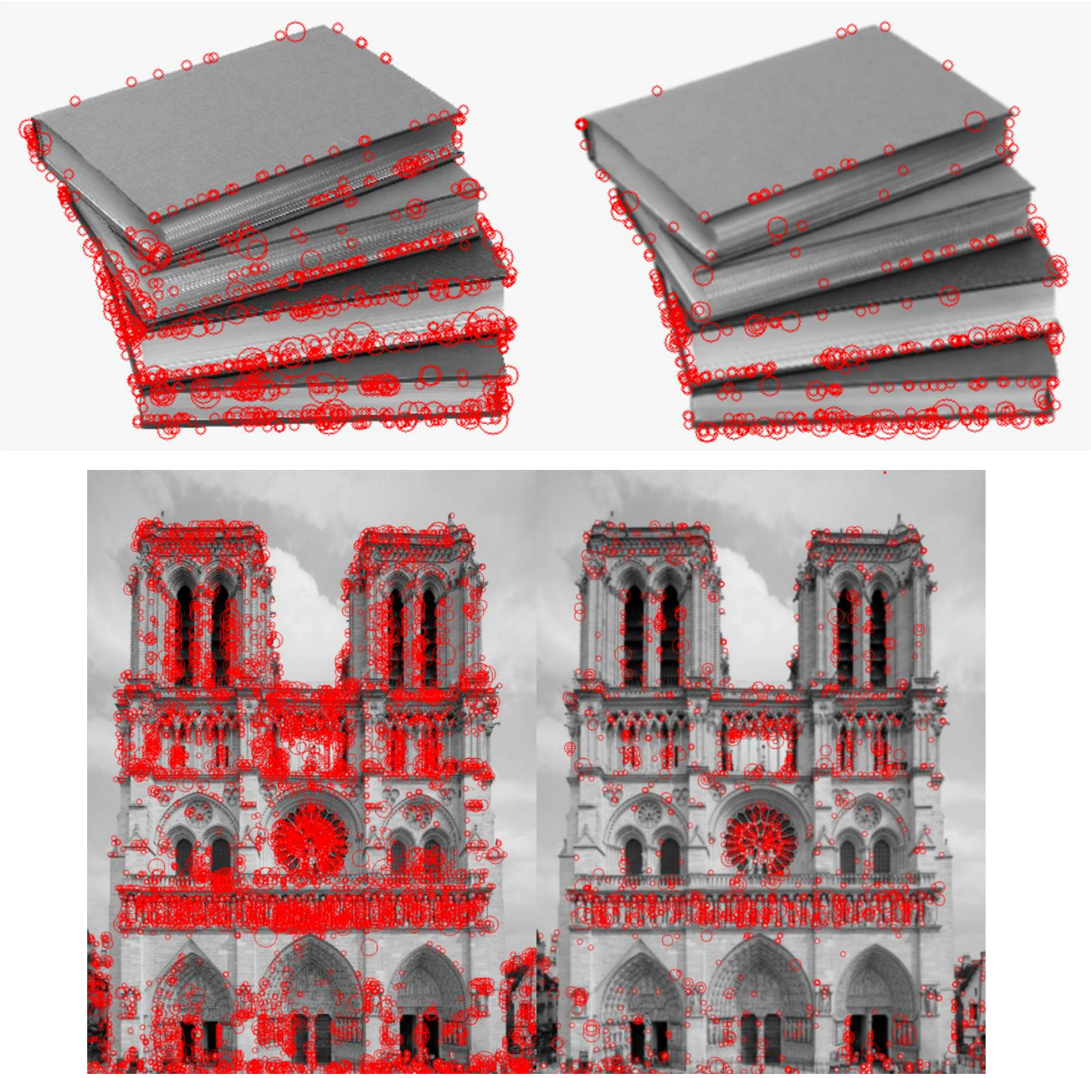


Fig3. Key points in Image-1 and Image-2 and corresponding Gaussian blurred version (3x3 kernel with sigma=3).

From the above images it is evident that the number of key points detected in the blurred image drops significantly as given below:

Number of detected key points:

Image 1- Original:15786 Blurred:5530

Image 2- Original: 2448 Blurred: 861

As we increase the blur in an image, it loses the finer details, small sized blobs, etc as key points because while convolving with the DoG (Difference of Gaussian), scale-space extremum will no longer occur at that point. Thus, overall, the bigger blobs get assigned as key points as can be seen from image 1 where the finer designs in the building no longer remain key points in the blurred version, only the bigger carvings prevail. In image 2 the points where gradient has a relatively larger magnitude prevail even if they are corners or edges.

Additional Points:

1. Since, in this implementation key point localisation and extremum thresholding steps were not added hence initially key points detected were unacceptably huge. In order to mitigate this issue, I made the condition for hessian check even stricter than given in the paper and obtained the results as presented.

2. Another possible method to mitigate the above could have been (although not implemented here) is to take strides of more than one for extrema detection. This will eliminate the extremely closely located key points and thus the number of detected key points will reduce.

3. Several parameters had to be tuned in order to get the presented results which are: Variance (2.7) and size of the Gaussian filter (5x5), scale factor per octave (k=2), threshold for the hessian step (r=4) and subsampling rate (x2) of images after every octave.

4. Overall, by performing all these operations we can conclude that SIFT is indeed quite robust in detecting key points and can be used for feature matching, this was also verified using the Lena image as shown below.

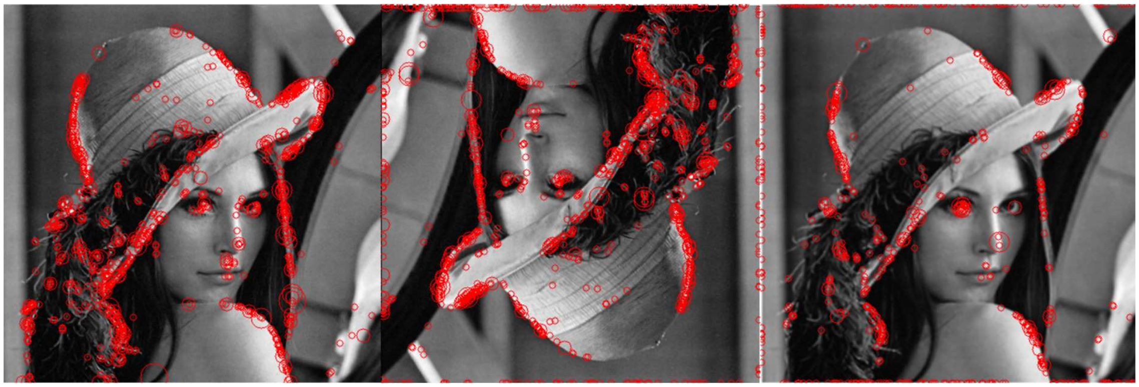


Fig 4. Key points in Lena original, rotated and blurred respectively.

5. For every key point detected PCA was performed to reduce the dimension of the descriptor vector associated with every key point to n=20.

2. Image Classification

CNN with same structure but using two different activation functions i.e. sigmoid activation in one and ReLU in the other network. They were trained on a custom dataset of 30000 images with labels and their performance was measured on a custom additional test dataset.

A total of four convolutional layers and a MaxPooling coupled with dropout layers to prevent overfitting in the network after every pair of convolutional layers were used. This was then connected to a couple of dense layers which was ultimately connected to a SoftMax layer. This architecture was selected after multiple hit and trials.

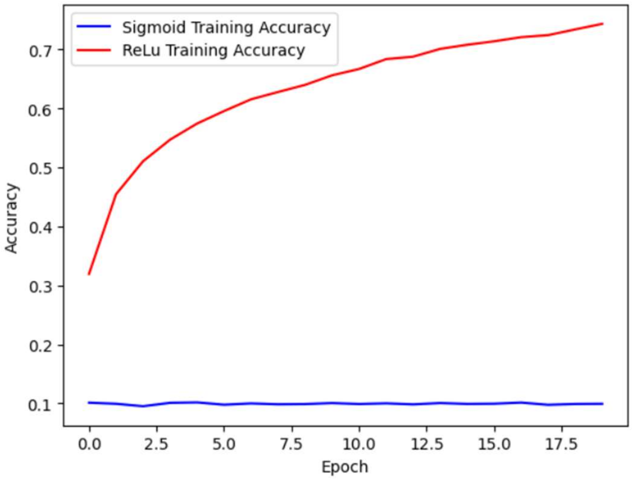
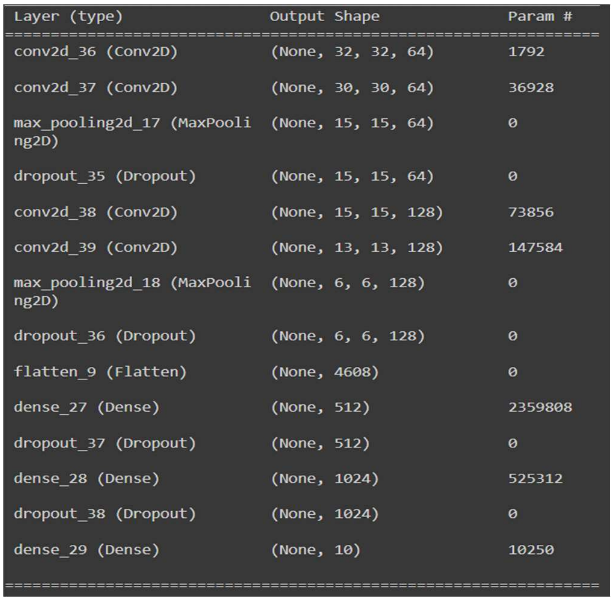


Fig 5. Graph showing the training accuracy for ReLu and sigmoid activations after every epoch.

ReLu out-performed the sigmoid network on the training and test data by a large margin, while on additional tests the accuracy was almost equal. The training as well as testing accuracies for sigmoid case didn't improve which is due to its popular disadvantage of vanishing gradients and absence of zero centred outputs.

	Test Data Accuracy (%)	**Additional Test Data Accuracy (%)**
Sigmoid	10.00	10.00
ReLU	64.25	9.83

It can be inferred from the table above that the performance of both the models are not up to the mark, this could be potentially due to lack of training data (30k instead of the standard 50k). When trained on the whole CIFAR10 dataset both the models performed much better.



```
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_36 (Conv2D)          (None, 32, 32, 64)        1792

 conv2d_37 (Conv2D)          (None, 30, 30, 64)        36928

 max_pooling2d_17 (MaxPooli  (None, 15, 15, 64)        0
 ng2D)

 dropout_35 (Dropout)        (None, 15, 15, 64)        0

 conv2d_38 (Conv2D)          (None, 15, 15, 128)       73856

 conv2d_39 (Conv2D)          (None, 13, 13, 128)       147584

 max_pooling2d_18 (MaxPooli  (None, 6, 6, 128)         0
 ng2D)

 dropout_36 (Dropout)        (None, 6, 6, 128)         0

 flatten_9 (Flatten)         (None, 4608)              0

 dense_27 (Dense)            (None, 512)               2359808

 dropout_37 (Dropout)        (None, 512)               0

 dense_28 (Dense)            (None, 1024)              525312

 dropout_38 (Dropout)        (None, 1024)              0

 dense_29 (Dense)            (None, 10)                10250

=================================================================
```

Fig 6. Graph showing the training accuracy for ReLu and sigmoid activations.

The other hyperparameters are Batch size (64), Filter size(3x3), epochs (20), optimizer used: Adam.