ASSIGNMENT1, ADVANCED IMAGE PROCESSING

Rishubh Parihar

CDS, Indian Institue Of Science, Bengaluru

1. SIFT

We performed the first step of SIFT algorithm i.e. scale space minima detection for a set of images. The following are the hyper-parameters used for the implementation: number of octaves-1, number of scales-6, Gaussian kernel size-15x15. For any input image, we first convert it into gray scale and apply a gaussian blur (with kernal size - 5 and sigma - 3) as a pre-processing step. We experimented with multiple image transformations applied to the input image e.g. rotation and scaling. Fig.1, 2 and 3 show results for SIFT feature points detected by out implementation. For each of the figure, we used two images of the same object/scene.

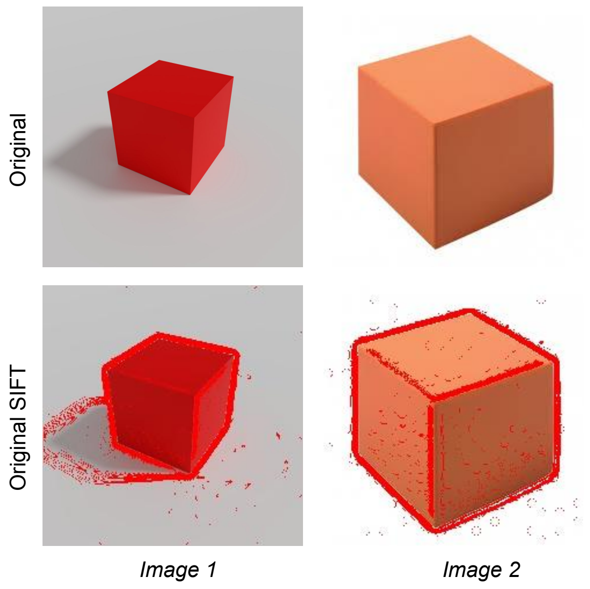

Fig. 1. SIFT feature extraction on a pair of images. The extracted features are marked in the red

Fig. 1: SIFT is able to detect most of the edge points of the cube in both the images. However given the background is smooth, few key-points are also detected in the background region. Also note that in *Image 1*, the key-points are also detected in the shadow of the cube which is expected as the shown also has edges and corners.

Fig. 2: We used the SIFT feature detector to detect key-points on two real world image of India Gate. We can observe that most of the key-points in the *Image 2* are detected on the boundaries of the monument and the background trees. Also, many key-points are detected in the interior region of the monument. For the *Image 1* more false positive points are detected in the sky region. Additionally, we performed the zoom operation on the original images and compute the SIFT key-points to evaluate the robustness of detection. In the zoomed version, we can see that most of the key-points lie on the borders and edges present in the monument. These detected key-points are much cleaner as compared to the key-points detected in the original image. As we used only a single octave for our implementation, our implementation is not scale invariant which is quite evident in this result.

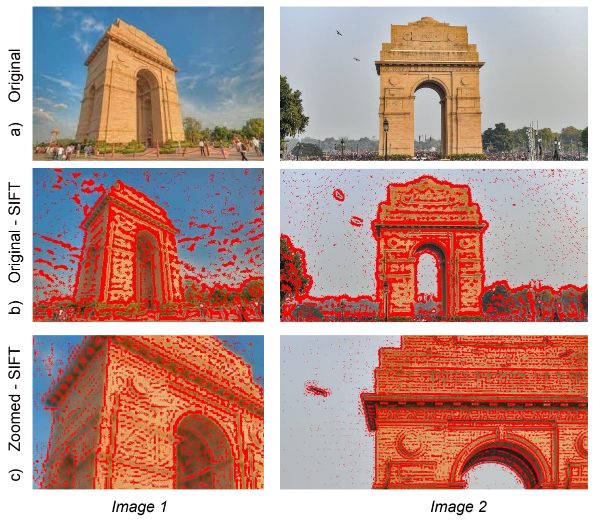

Fig. 2. SIFT feature extraction on a pair of images. The extracted features are marked in the red

Fig. 3: We used the SIFT key-point detection on a more complex scene with large illumination variations. We can observe that although many key-points are detected on the pillars of Pantheon, but a large number of key-points are also detected in the sky and the background regions. Further, we performed rotation by 45 degrees on the original image and applied SIFT on the image as shown in Fig.3 c). It can be observed that the most of the detected key-point location is similar to the locations in the original image. This suggest that the SIFT key-point detector is rotation in invariant.

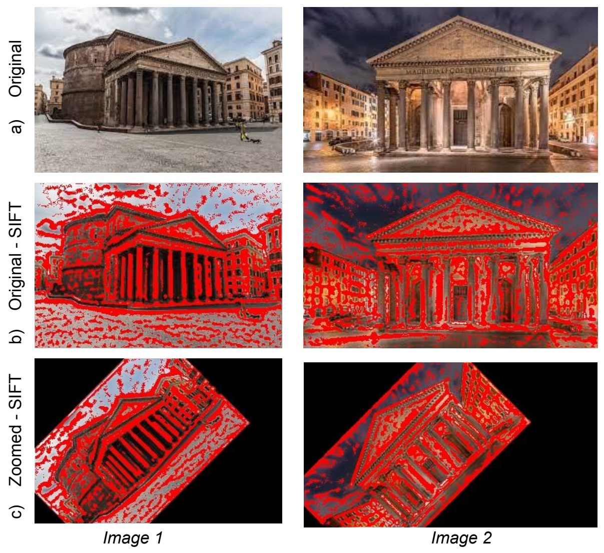

Fig. 3. SIFT feature extraction on a pair of images. The extracted features are marked in the red

2. CNN

In this section, we will discuss about out implementation for 6 class classification problem. In the given dataset, the training set consists of 240 images and the test set consists of 120 images. We have used three classifiers and compared them for the task of classification. The classifiers are:

1) **Nearest Neighbour classifier:** We used pre-trained VGG16 model on ImageNet dataset and extracted features of 4096 dimensions before the last fully connected layer. We used these features to perform nearest neighbor classification on the test set and acheived 0.97 accuracy. This finding suggest that the pre-trained representations are of very high-quality and they are enough to perform this classification. One of the primary reason for this could be that these 6 classes are a part of 1000 imageNet classes resulting in clear separation between then in the feature space.

2) **VGG16 classifier:** We used the same pre-trained VGG16 model and trained the last fully connected layer of input dimensions 4096 and output dimension of 6. The model was converged only in the first 10 epochs as can be seen in Fig. 4. The fast convergence can be accredited to the pre-training of the model. With experimentation, we used learning rate of 0.0001 and a batch size of 32 for training.

3) **Custom model:** We trained a custom CNN model with three convolution layers and one fully connected layer of the same size as of VGG16 last fc layer. The training progression is shown in Fig. 5, where we can observe that the model took 35 epochs to convergence. This is as expected as we are not using any pretrained weights and the weights are learnt from scratch. Also note that for custom model the accuracy is least among the three models, enforcing the fact that to train CNN model either we need good pre-trained weights or large number of data points.

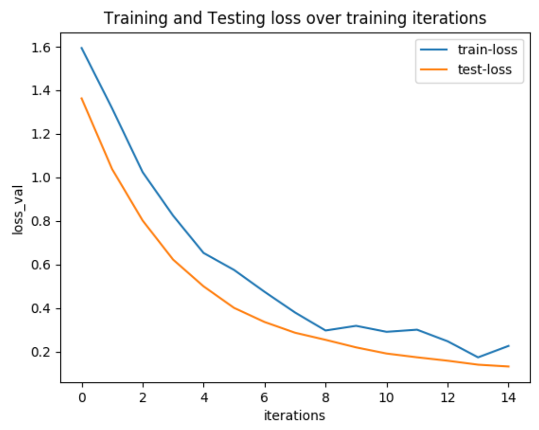

Fig. 4. Plot of training and test losses for VGG16 model with epochs.

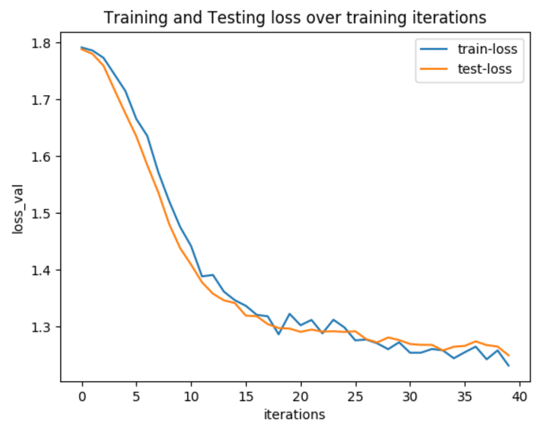

Fig. 5. Plot of training and test losses for Custom model with epochs.

Table 1. Results for NN, VGG and Custom model on 6-way classification task) & LSTM(SSL-2

	NN	VGG16	Custom
test acc.	0.97	1.00	0.80