AIP Assignment 1

Manogna S

Q1. SIFT Keypoint detection

SIFT Keypoint detection is implemented on following images by constructing a DOG pyramid with 4 octaves and 3 scales per octave. Eliminating weak key points is observed to be crucial to detect good features which are repeatable when image transforms are applied. Keypoints are eliminated based on value at extrema in dog pyramid. Also keypoints are thresholded based on the ratio of eigen values of hessian at extrema inorder to eliminate keypoints along edges.

Image1 (223X324)

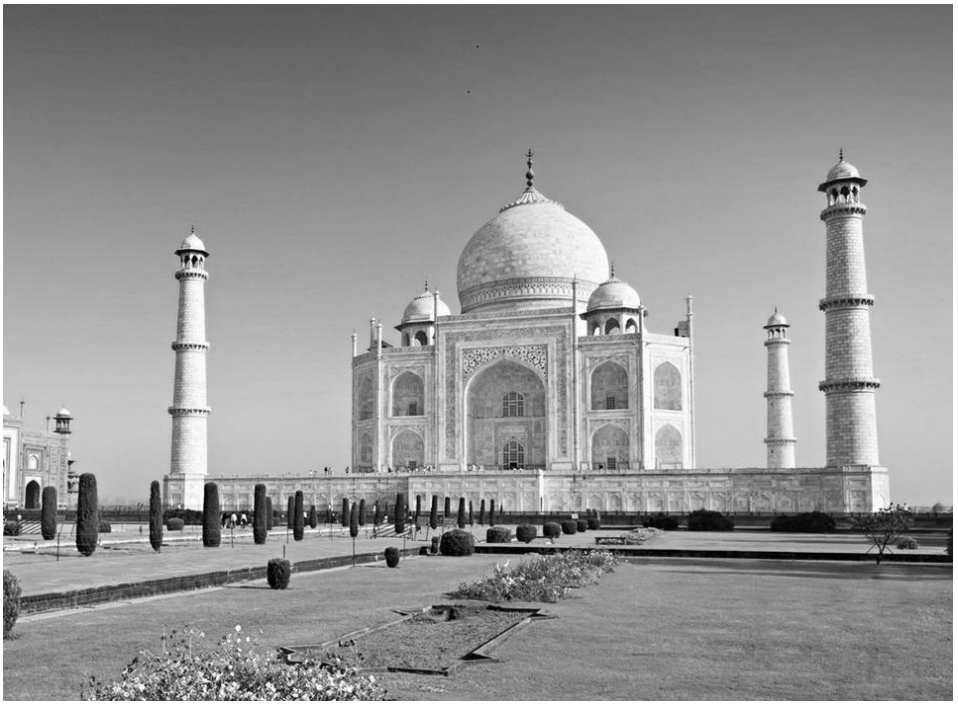
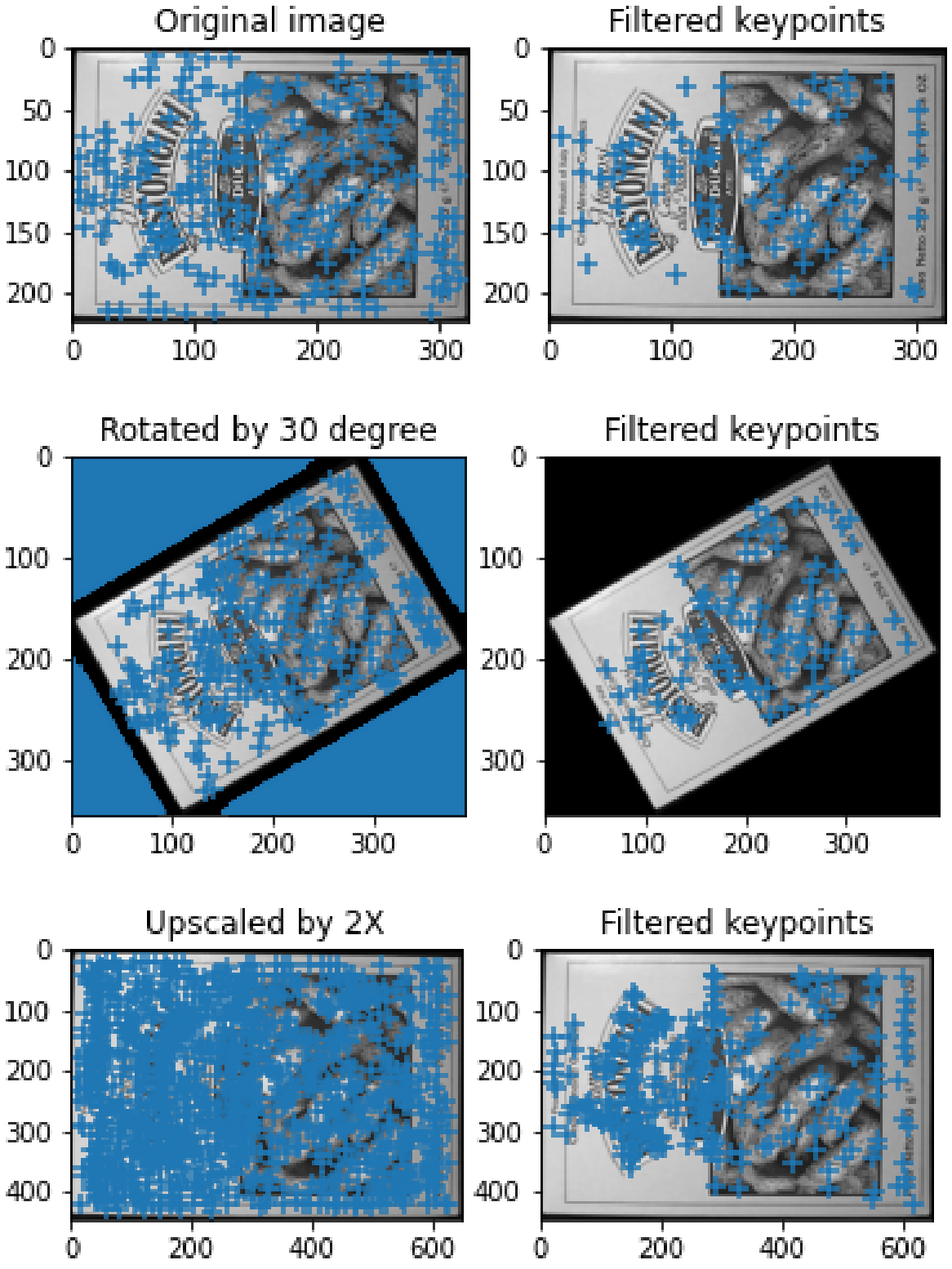


Image2 (706x959)

The following are the scale space extrema detected(along column 1) and final keypoints(along column 2) after eliminating weak keypoints based on threshold and eigen values of hessian at extrema.

Image1 Results:



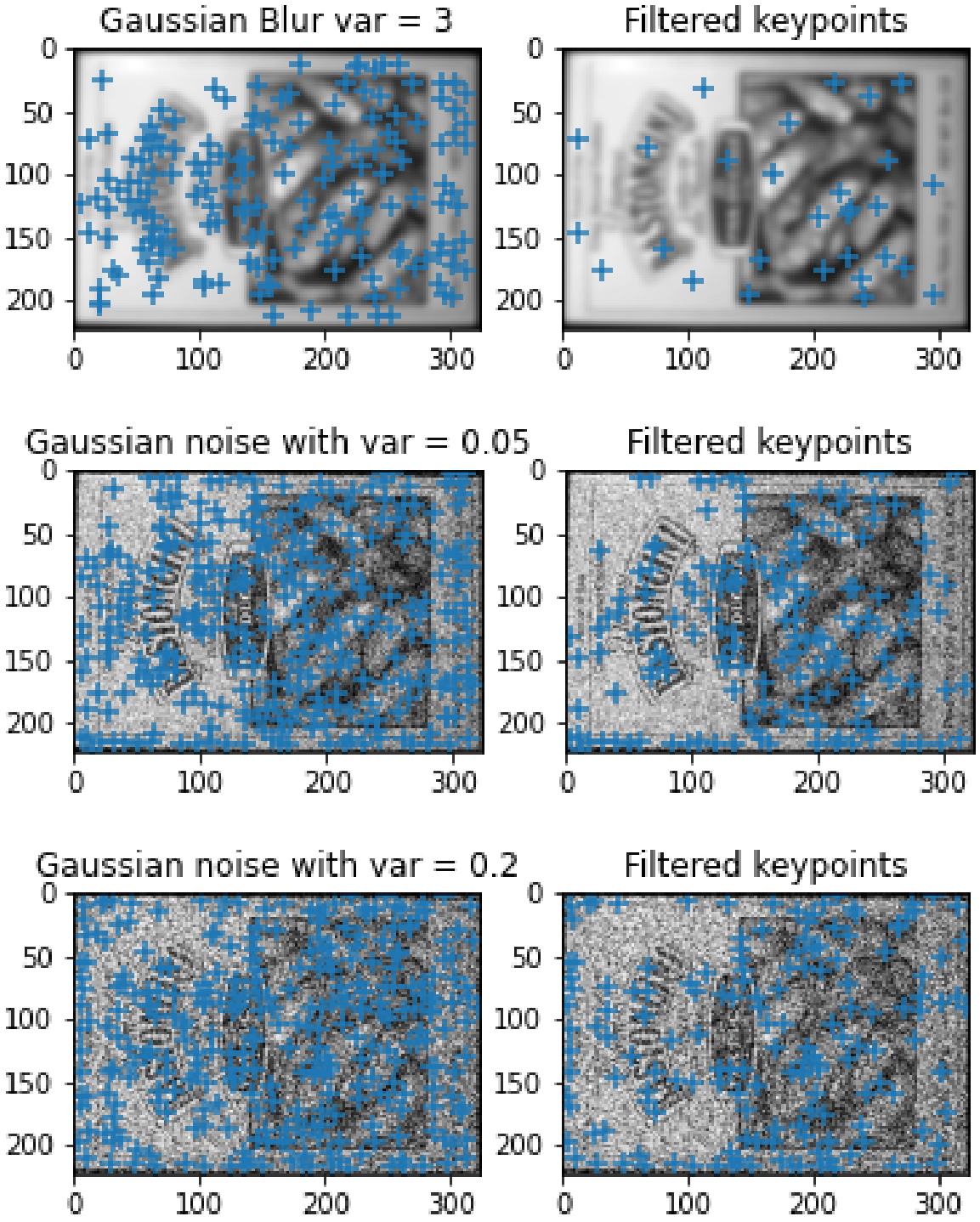
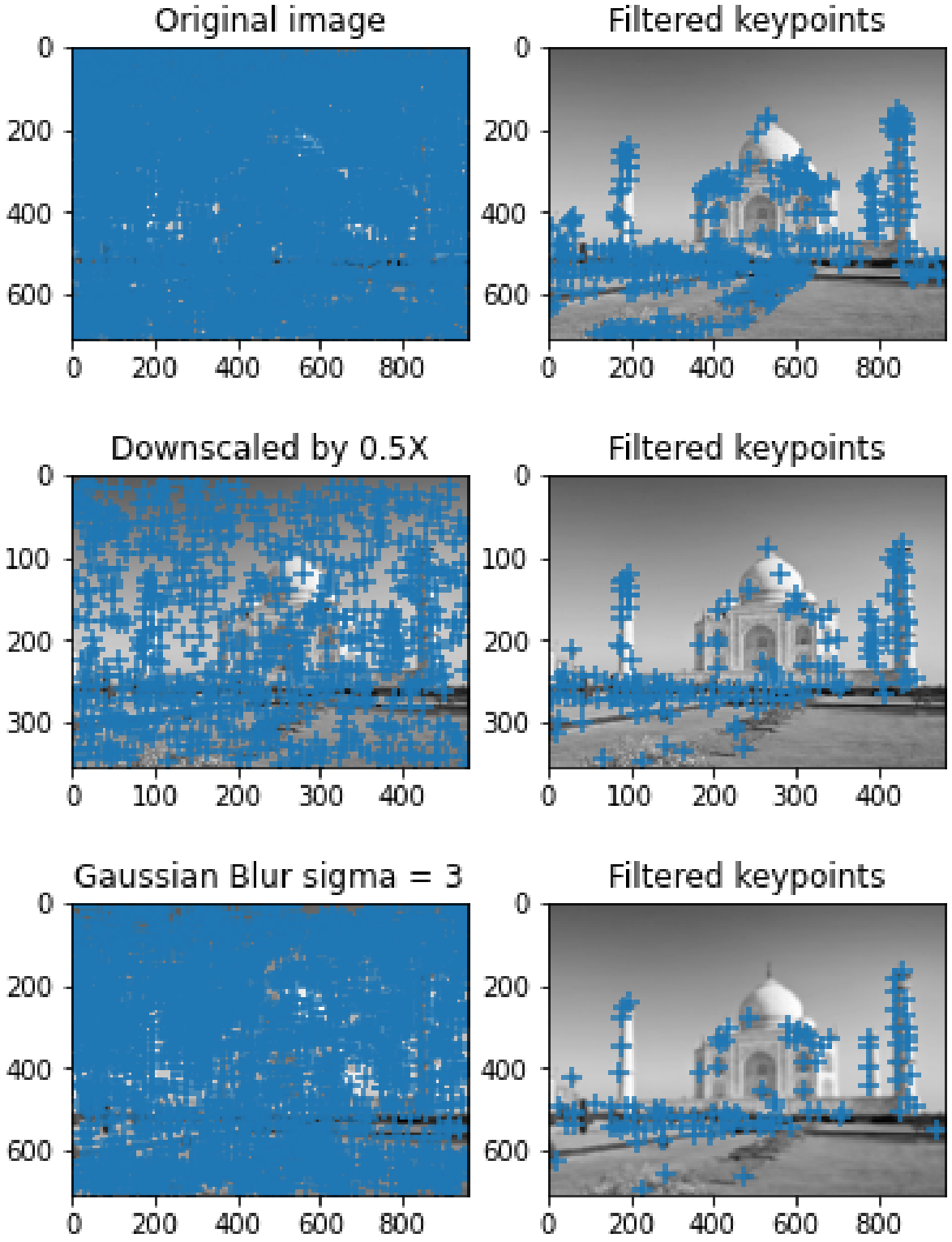
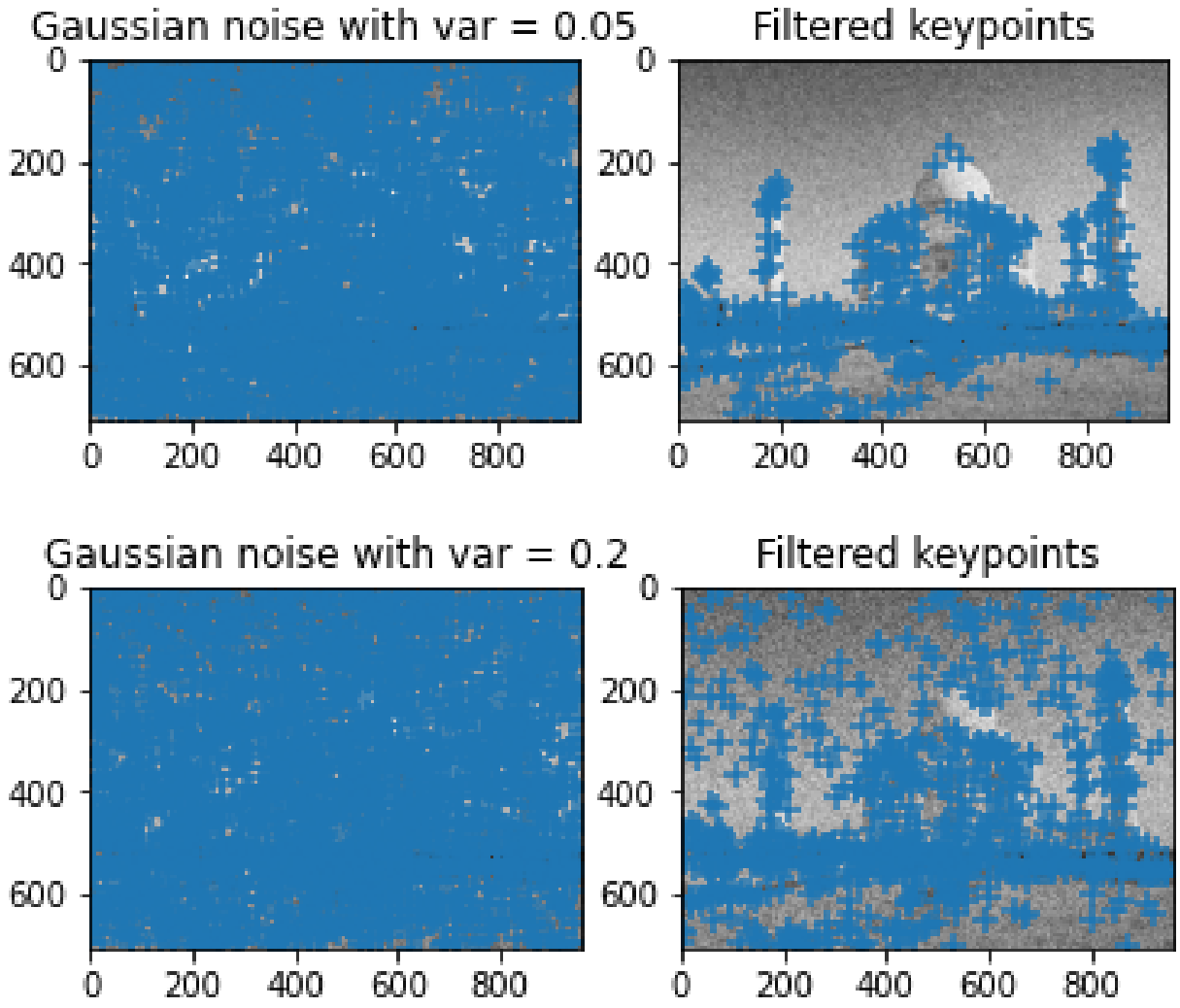

Gaussian Blur var = 3
Filtered keypoints
Gaussian noise with var = 0.05
Filtered keypoints
Gaussian noise with var = 0.2
Filtered keypoints
0
50
100
150
200
0
100
200
300

Image2 Results:




Gaussian noise with var = 0.05
Filtered keypoints
Gaussian noise with var = 0.2
Filtered keypoints
0
200
400
600
800

Observations on SIFT Results:

No. of keypoints:
In Image1, 238 keypoints were initially detected and 98 remained after eliminating weak keypoints. In Image2, ~3000 keypoints were initially detected and resulted in 382 keypoints after eliminating weak keypoints. Higher resolution images result in a lot of scale space extrema.

Rotation:
Rotating the images results in almost the same no. of keypoints and most of them can be matched one on one with the keypoints originally detected.

Scaling:
Upscaling an image(Image 1) results in more dense keypoints(even after filtering keypoints) but they can still be matched with the keypoints detected in the original image and hence SIFT keypoints appear to be stable for scale changes as expected. Downscaling an image(Image 2) results in detecting stable keypoints which can be matches with that of the original image.

Blurring:
Blurred images have very smooth edges and poor gradients. The keypoints detected after scale space extrema can be matched with the actual keypoints but using the same threshold to eliminate weak keypoints rejects most of them. So keypoints should be carefully in such images. Keypoint detection is more sensitive to parameters in such images.

Noise:
Adding gaussian noise results in detecting more keypoints. When gaussian noise with variance 0.05 is added, the keypoints matching the true keypoints in clean image appear to be detected. But more keypoints are densely detected in those regions. In the case of noise variance of about 0.2, a lot keypoints are detected even in smooth regions due to the noise. So SIFT keypoints are stable to noise to a certain extent.

Q2. CNN Features

Pretrained model: VGG16 trained on ImageNet data

Test Accuracy (in %)	k-NN Classifier (k=5, 10, 15)	VGG based Classifier		Model trained from scratch (Model 1)
		'fc1' layer	'fc2' layer	
Airplanes	100	100	100	85
Bikes	100	95	100	100
Cars	100	100	100	95
Faces	100	100	100	95
Total	**100**	**100**	**100**	**93.75**

Table 1: Comparison of different model accuracies

Observations for kNN and CNN based classifier:
As the pretrained VGG16 model is trained on ImageNet which includes similar classes, the model seems to have learnt discriminative features for the given classes and the kNN classifier accurately classifies all the test images. For part (B), the VGG16 feature extractor was frozen and 4 node classification layer was added to the fully connected layer(fc1 or fc2) and trained. This model also resulted in ~100% accuracy on the test set. The kNN results show that the features are discriminative and hence the model would learn good decision boundaries.

Models trained from scratch:

Two models were trained from scratch and the results for the same are reported. It is a sequence of 3x3 convolution and 2x2 pooling layers with no. channels increasing 2x. Following these spatial convolution layers, Model 1 has one a flatten layer(resulting in a 4096 dimensional feature vector) followed by 4 node classification head. Model 2 has an average pooling layer(resulting in a 256 dimensional feature vector) followed by 4 node classification head.

Model 1		Model 2	
Layer	**Output shape**	**Layer**	**Output shape**
Input	128x128x3	Input	128x128x3
Conv (3x3x16)	128x128x16	Conv (3x3x16)	128x128x16
Pool (2x2)	64x64x16	Pool (2x2)	64x64x16
Conv (3x3x32)	64x64x32	Conv (3x3x32)	64x64x32
Pool (2x2)	32x32x32	Pool (2x2)	32x32x32
Conv (3x3x64)	32x32x64	Conv (3x3x64)	32x32x64
Pool (2x2)	16x16x64	Pool (2x2)	16x16x64
Conv (3x3x128)	16x16x128	Conv (3x3x128)	16x16x128
Pool (2x2)	8x8x128	Pool (2x2)	8x8x128
Conv (3x3x256)	8x8x256	Conv (3x3x256)	8x8x256
Pool (2x2)	4x4x256	Pool (2x2)	4x4x256
Flatten	**4096**	**Global Average Pool**	**256**
Dense	**4**	**Dense**	**4**

Table 2: Model definitions

Test Accuracy (in %)	**Flatten + Dense layer (Model 1)**		**Global average pool + Dense layer(Model 2)**	
	epochs = 20	**epochs = 40**	**epochs =20**	**epochs = 40**
Airplanes	75	85	80	60
Bikes	100	100	80	95
Cars	100	95	65	100
Faces	95	95	85	85
Total	**92.5**	**93.75**	**77.5**	**85**

Table 3: Comparison of accuracy for models trained from scratch

Training details	
Input resolution	128x128
Data augmentation	• Random horizontal flip • Random zoom(upto 20%) • Random rotation(upto ±10% x 2π)
Optimizer	Adam
Loss	Cross entropy loss

Observations for models trained from scratch:
Model 1 with 4096 dimensional feature vector attains a test accuracy of 92.5% in 20 epochs and saturates at about ~94% training for more epochs. Model 2 attains a test accuracy of 77.5% in 20 epochs and saturates at about 85-88% on training for more epochs. This is possibly because of losing spatial detail due to global average pooling layer resulting in a 256(<<4096) dimensional feature vector. It would be better to use flatten/fully connected layers instead of global average pooling (at least for shallow networks) to learn good discriminative features for classification.
The same models were trained for 224x224 input image resolutions and were observed to produce similar results. Hence, results are reported only for image resolution of 128x128.

Links for Code:
1. SIFT
2. CNN classifier

The same has also been submitted.