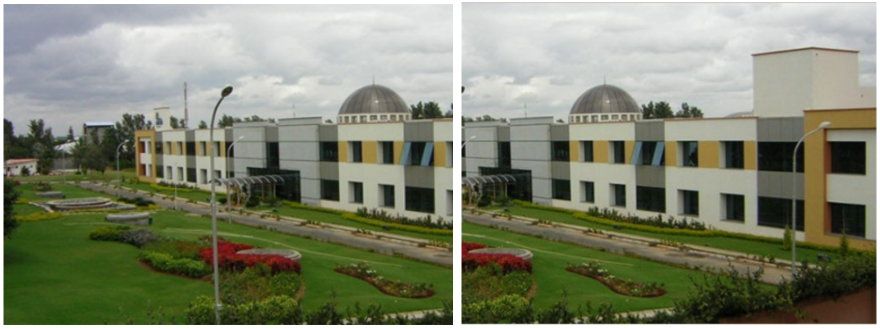
Assignment 2 Report

Ajay RR (IMT2017502) **Ronak Doshi (IMT2017523)** **Ram S (IMT2017521)**

Visual Recognition Course
International Institute of Information Technology Bangalore
February 24 2020

1 Image Panorama Stitching

Image stitching refers to the creation of a composite image from a group of images which correspond to a scene. Two images of IIIT-Bangalore were stitched together to form a panorama, using Opencv and Python. The two images that were stitched are shown in Figure 1.



(a) Image 1 (b) Image 2

Figure 1: Original images to be stitched

The approach adopted to successfully stitch these images is described below :

1. **Keypoint Detection and Descriptor Computation :** The first step towards matching the features of the images being stitched is the detection of keypoints. The keypoints were detected and their descriptors computed, using the SIFT algorithm available in Opencv. The scale invariant property of SIFT is very important as the two images that need to be stitched may vary slightly in scale, which is the case with the images shown in Figure 1.
2. **Feature Matching :** In order to join the two images, the overlapping keypoints have to be detected. These overlapping points give an idea of the orientation of the second image with respect to the first. In the result shown, we use the DoG (Difference of Gaussian) keypoint detector and SIFT feature extractor. The result of brute force matching between the two images is shown in Figure2. We also apply Lowe's ratio test (ensuring the distance is within a certain ratio of each other) on the matching. For computational reasons, FLANN matcher could also be used.

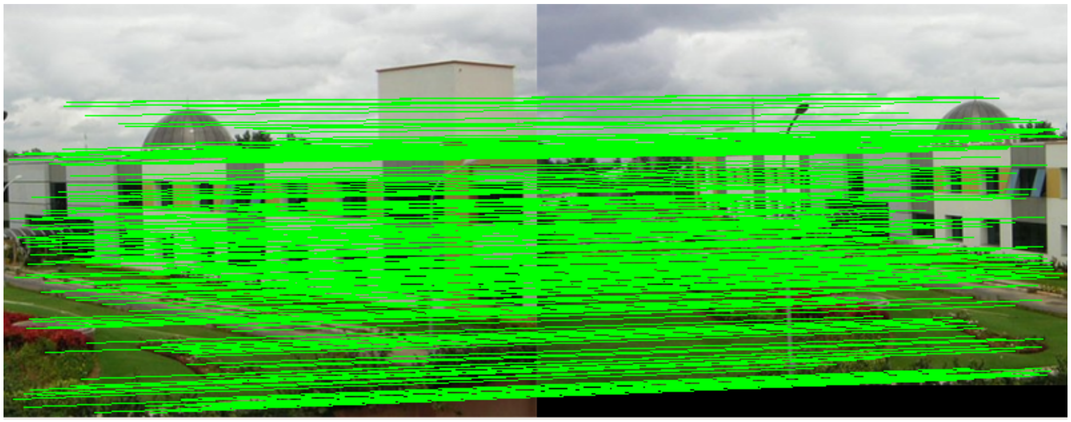


Figure 2: Feature Matching

3. **Homography Matrix Estimation using RANSAC Algorithm:** A Homography is a transformation that maps the points in one image to corresponding points in the other image. Computing the homography between two sets of points requires at the bare minimum an initial set of 4 matches. However, for a more reliable estimation, we should have substantially more than 4 matches. More details about RANSAC and FLANN are mentioned in Section 3.

4. **Perspective Warping :** Given our Homography matrix, we can stitch the two images together through a perspective transform. We call cv2.warpPerspective which accepts 3 arguments:

   (a) The image we wish to warp

   (b) The Homography Transformation Matrix

   (c) The shape of the output image. We consider the shape as the sum of the widths of both the images and height as height of the second image.

   After the perspective wrap, the resulting image is shown in Figure 3.

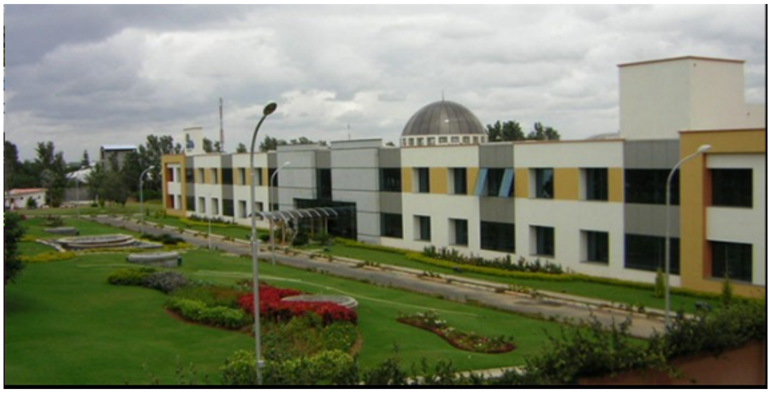
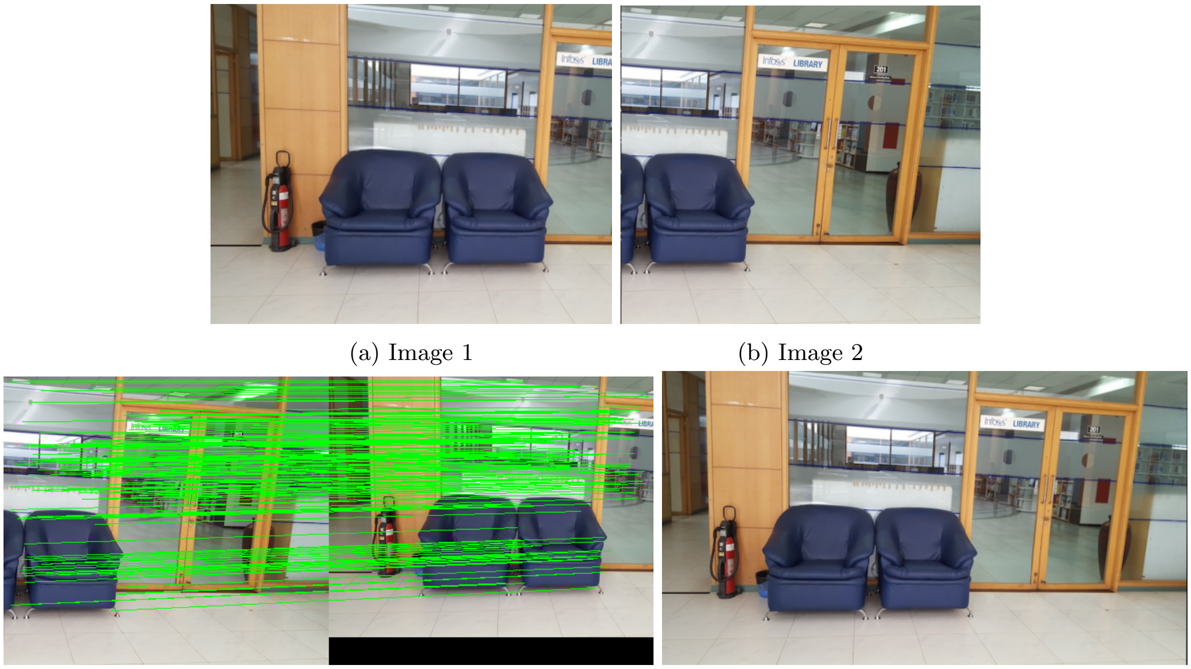


Figure 3: Panorama

The results of stitching for a different pair of images is shown in Figure 4.



(a) Image 1

(b) Image 2

(c) Keypoint matching

(d) Panorama

Figure 4: Results

2 SIFT vs SURF

2.0.1 SIFT (Scale Invariant Feature Transform)

- SIFT estimates a scale space extrema using the Difference of Gaussian (DoG). The DoG is convolved with different size images with same filter size.
- It uses local extrema detection, applies Non maxima suppression and eliminates edge response with Hessian matrix.
- Keypoint orientation is assigned based on local image gradient. Image gradient magnitude and orientations are sampled around the key point location, using the scale of the key point to select the level of Gaussian blur for the image.
- Descriptors are computed for each keypoint based on the gradient magnitude and orientation.
- 128 dimensions for each interest point.

2.0.2 SURF (Sped Up Robust Feature)

- SURF approximates the DoG with box filters. Instead of Gaussian averaging the image, squares are used for approximation since the convolution with square is much faster. This is done in parallel for different scales.
- It determines the key points with Hessian matrix and Non Maxima suppression.
- Wavelet responses in both horizontal and vertical directions are used with adequate gaussian weights for orientation assignment. A sliding orientation window detects the dominant orientation of the Gaussian weighted Haar Wavelet responses at every sample point with in a circular neighbourhood around the interest points.
- A neighborhood around the key point is selected and divided into sub regions and then for each sub region the wavelet responses are taken and represented to get the feature descriptor.
- 64 dimensions for each interest point.

3 FLANN and RANSAC

3.1 FLANN : Fast Library for Approximate Nearest Neighbors

FLANN is a library for performing fast approximate nearest neighbor searches. The nearest neighbor search problem can be defined as follows :
Given a set of points P = p1, p2, . . . , pn in a metric space X, these points must be pre-processed in such a way that given a new query point q $\in$ X, finding the point in P that is nearest to q can be done quickly. FLANN provides various algorithms such as the kd-tree to preform the nearest neighbor search. A k-d tree is a binary tree in which every leaf node is a k-dimensional point. Searching for the nearest neighbor in a k-d tree will be more efficient as binary tree properties can be exploited to search recursively.

3.2 RANSAC : Random Sample Consensus

RANSAC is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers. Models like linear regression use least squares to fit the best model. This is very sensitive to outliers. The goal of RANSAC is to identify the outliers and eliminate them from the final fit. RANSAC fits the model only on the identified set of inliers. The RANSAC algorithm can be summarized to the following steps :

- Select a random subset of the original data. This forms the hypothetical inliers.
- Fit a model to this subset.
- Test all other data against the model using a specific measure of error (loss function) and choose which points agree well with the estimated model (An error threshold is chosen based on the application and data set). These points form the consensus set.
- The estimated model is considered good if sufficient points are part of the consensus set.
- The model is improved by re-estimations using all members of the consensus set.

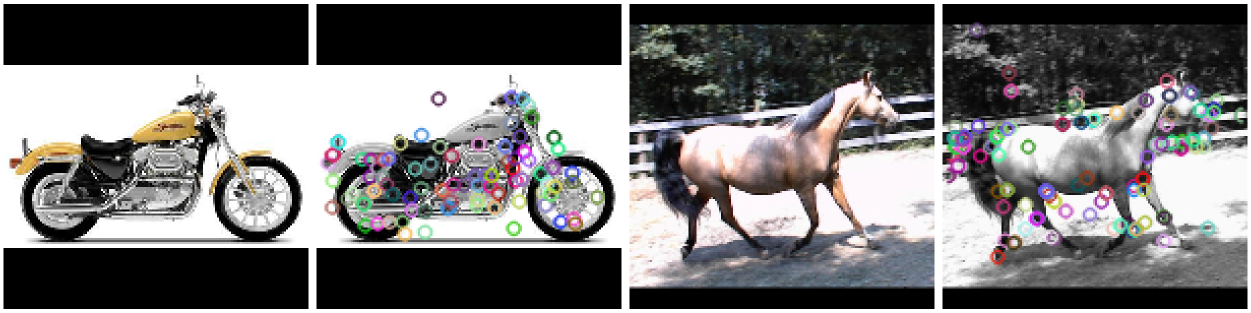
4 Bike vs Horse Classification

Here given an image we try to classify whether it is a bike image or a horse image using SIFT/SURF and bag of visual words technique.

4.1 Bag of Visual Words

The Bag of Visual Words model is an important concept in computer vision and is commonly used in image classification. The method has been adapted from the Bag of Words technique of Text processing, where the frequency of each word is computed, and a set of keywords are identified. These image features comprise of various unique patterns that can be identified from the image. The final idea of the Bag of Visual Words model is to represent every image with a set of features which can be used for image classification. The Bag of Visual Words model was used to classify images of Bikes and Horses. The procedure is described below :

1. **Pre-processing :** The images were converted to gray scale and resized to $100X100$ pixels.
2. **Keypoint Detection and Descriptor Computation :** The SURF algorithm was used to detect keypoints and compute their descriptors. The detected keypoints for both classes of images is shown in Figure 5.



(a) Bike (b) Keypoints (c) Horse (d) Keypoints

Figure 5: Keypoint Detection

3. **Descriptor Clustering :** The descriptors were clustered and the centre of each cluster was used as the visual dictionary's vocabulary. Clustering was performed using the K-means algorithm. The optimal value of K was determined to be 10 using the elbow method as shown in Figure 6. The Elbow method looks at the total Within Cluster Sum of Square (WSS) as a function of the number of clusters. One should choose a number of clusters so that adding another cluster doesn't improve much better the total WSS.

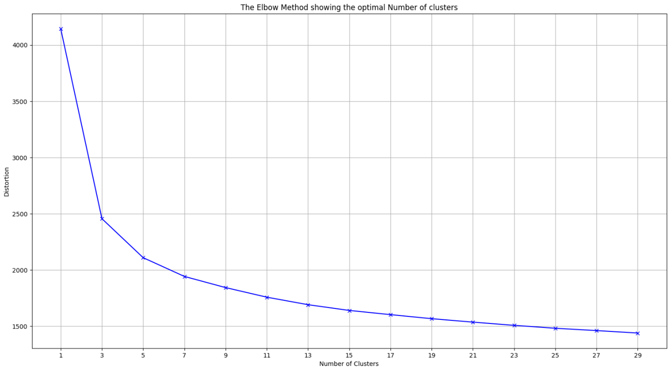


Figure 6: Elbow plot

4. **Histogram Computation :** The frequency histogram of the vocabularies was computed for each image, ie, the number of bins is equal to the number of cluster centers. This forms the feature vector for the image.

4.2 Training and Classification :

After generating feature vector for all the images, the model was trained for classification purposes. Two different classifiers were used :

- K Nearest Neighbors Classifier
- Random Forest Classifier

We trained the model on 70% of the available data-set and tested on the rest 30%. Accuracy that we got for each model is as follows:

Table 1: Accuracy Table

Model	**Accuracy ± variance**
k-Nearest Neighbours with $k = 5$	0.90 ± 0.03
k-Nearest Neighbours with $k = 7$	0.89 ± 0.05
Random Forest Classifier	0.935 ± 0.08

5 CIFAR 10

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

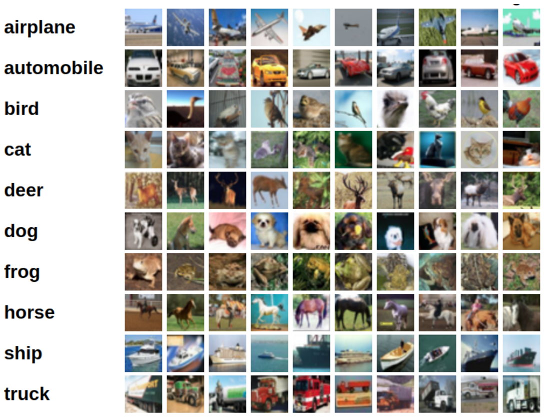


Figure 7: Data Set

The images are of very low resolution and hence very few keypoints can be detected. Thus, the SIFT descriptors are insufficient to be used as feature vectors. The images were resized to a better resolution of $100X100$ but it did not improve the classification accuracy. The results

Table 2: Accuracy Table

Model	**Accuracy**
k-Nearest Neighbours with $k = 5$	21.3%
k-Nearest Neighbours with $k = 7$	20%
Random Forest Classifier	25.5%

References

[1] https://www.researchgate.net/publication/314285930_Comparison_of_Feature_Detection_and_Matching_Approaches_SIFT_and_

[2] https://www.cs.ubc.ca/ lowe/papers/ijcv04.pdf

[3] https://www.vision.ee.ethz.ch/ surf/eccv06.pdf

[4] https://www.vision.ee.ethz.ch/ surf/eccv06.pdf