Visual Recognition Assignment-2

Arya Kondawar(IMT2020084)

Assumtions:

- For Q2a I wasn't able to upload the files in submissions, assuming after downloading the images from here, to run the notebook save the images in same folder in which code is present, check for the libraries required from imports and then run the notebook.
- For Q2b download the dataset from LMS, change the folder's name containing Bike and Horse dataset to "Content" and save the "Content" folder in the same folder where the code is saved, check for the libraries required from imports and then run the notebook.
- For Q2b CIFAR-10 classification download the dataset for python(CIFAR-10 python version from this link or from here) and after unzipping/extracting the "cifar-10-batches-py" folder store the "cifar-10-batches-py" folder in the same folder where the code is present, check for the libraries required from imports and then run the notebook.

2a. Play With Panaroma:

SIFT v/s SURF:

SIFT (Scale-Invariant Feature Transform) and SURF (Speeded Up Robust Features) are two popular algorithms used for feature detection and matching in computer vision. Both algorithms are used for object recognition, image registration, and other related applications. Here are some of the differences between SIFT and SURF:

- SURF is faster than SIFT due to the use of a box filter instead of a Gaussian filter. This reduces the number of calculations required and allows for faster feature detection and matching.
- Both SIFT and SURF are scale-invariant, but they use different methods to achieve this. SIFT uses a difference of Gaussians (DoG) filter to detect features at different scales, while SURF uses a Gaussian filter at multiple scales.
- SIFT uses a 128-dimensional vector, while SURF uses a 64-dimensional vector.
- In SIFT, Lowe approximated Laplacian of Gaussian with Difference of Gaussian for finding scale-space. SURF goes a little further and approximates LoG with Box Filter.

- SURF rely on determinant of Hessian matrix for both scale and location.
- SURF uses wavelet responses in horizontal and vertical direction for a neighborhood of size 6s. Both algorithms are rotation-invariant, but SURF is more robust to rotation due to the use of a Haar wavelet response.
- SIFT has good results than SURF in scale area. SURF has good results in rotation, blur, warping, RGB noise, and time consumption than SIFT. In illumination, both have the same effect on find detect feature points.
- SURF is more robust to changes in illumination, noise, and occlusion than SIFT.

In Summary, The scale-inavariant feature transform(SIFT) an algorithm used to detect and describe local features in digital images. It locates certain key points and then furnishes them with quantitative information (so-called descriptors) which can for example be used for object recognition. The descriptors are supposed to be invariant against various transformations which might make images look different although they represent the same objects. A SIFT feature is a selected image region (also called keypoint) with an associated descriptor. Keypoints are extracted by the SIFT detector and their descriptors are computed by the SIFT descriptor. It is also common to use independently the SIFT detector (i.e. computing the keypoints without descriptors) or the SIFT descriptor

SURF is the speed up version of SIFT. SURF goes a little further and approximates LoG with Box Filter. One big advantage of this approximation is that, convolution with box filter can be easily calculated with the help of integral images. And it can be done in parallel for different scales. Also, the SURF rely on determinant of Hessian matrix for both scale and location. For orientation assignment, SURF uses wavelet responses in horizontal and vertical direction. Adequate guassian weights are also applied to it.The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of angle 60 degrees. wavelet response can be found out using integral images very easily at any scale. SURF provides such a functionality called Upright-SURF or U-SURF. It improves speed and is robust . OpenCV supports both, depending upon the flag,upright. If it is 0, orientation is calculated. If it is 1, orientation is not calculated and it is faster.

FLANN MATCHING:

FLANN stands for Fast Library for Approximate Nearest Neighbors. It is an image matching algorithm for fast approximate nearest neighbor searches in high dimensional spaces. These methods project the high-dimensional features to a lower-dimensional space and then generate the compact binary codes. Benefiting from the produced binary codes, fast image search can be carried out via binary pattern matching or Hamming distance measurement, which dramatically reduces the computational cost and further optimizes the efficiency of the search.

It contains a collection of algorithms optimized for fast nearest neighbour search in large datasets and for high dimensional features. It works faster than BFMatcher for large datasets.

However, it will find a good search but not necessarily the best one. The real benefit of FLANN is seen with large data sets.

FLANN uses a tree-based approach to efficiently find the approximate nearest neighbors of a query point in a set of high-dimensional points. It uses hierarchical clustering to build a tree-based index for the feature descriptors in the database. It is able to find a good match for a query descriptor even if an exact match is not present in the database. It also supports a variety of distance measures for comparing feature descriptors, including the Euclidean distance and the cosine distance.

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, when outliers are to be accorded no influence on the values of the estimates. Therefore, it also can be interpreted as an outlier detection method.

The RANSAC algorithm is a learning technique to estimate parameters of a model by random sampling of observed data. Given a dataset whose data elements contain both inliers and outliers, RANSAC uses the voting scheme to find the optimal fitting result. Data elements in the dataset are used to vote for one or multiple models. The implementation of this voting scheme is based on two assumptions: that the noisy features will not vote consistently for any single model (few outliers) and there are enough features to agree on a good model (few missing data). The RANSAC algorithm is essentially composed of two steps that are iteratively repeated:

1. In the first step, a sample subset containing minimal data items is randomly selected from the input dataset. A fitting model with model parameters are computed using only the elements of this sample subset. The cardinality of the sample subset (e.g., the amount of data in this subset) is sufficient to determine the model parameters.
2. In the second step, the algorithm checks which elements of the entire dataset are consistent with the model instantiated by the estimated model parameters obtained from the first step. A data element will be considered as an outlier if it does not fit the model within some error threshold defining the maximum data deviation of inliers. (Data elements beyond this deviation are outliers.)

The set of inliers obtained for the fitting model is called the *consensus set*. The RANSAC algorithm will iteratively repeat the above two steps until the obtained consensus set in certain iteration has enough inliers.

An advantage of RANSAC is its ability to do robust estimation of the model parameters i.e. it can estimate the parameters with a high degree of accuracy even when a significant number of outliers are present in the data set. A disadvantage of RANSAC is that there is no upper bound on the time it takes to compute these parameters . When the number of iterations computed is limited the solution obtained may not be optimal, and it may not even be one that fits the data in a good way. In this way RANSAC offers a trade-off by computing a greater number of iterations the probability of a reasonable model being produced is increased. Moreover, RANSAC is not always able to find the optimal set even for moderately contaminated sets and it usually performs badly when the number of inliers is less than 50%.

RESULTS:

Left Image

Right Image

Stitched Image

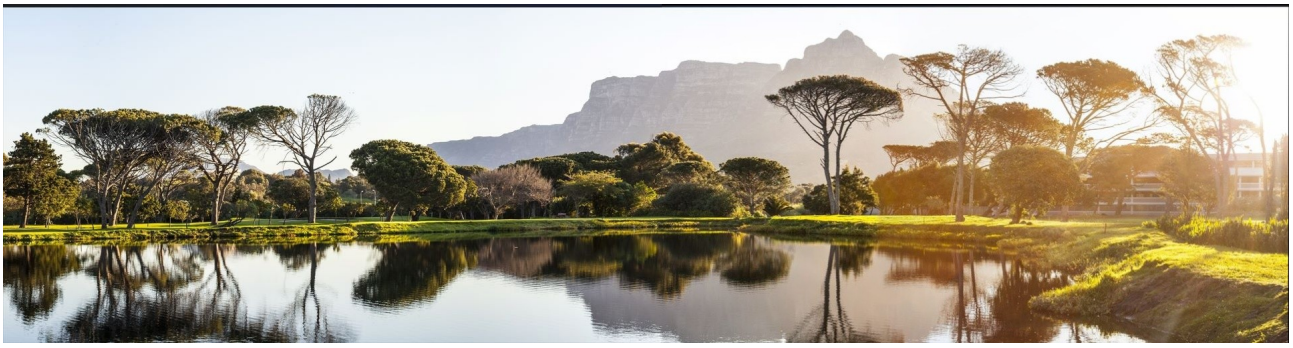
Left Image

Right Image

Stitched Image

Left Image

Right Image



Stitched Image

Since all the test images were very large I couldn't add them in the submission. Hence, you can find them over here : Test Images

2b.

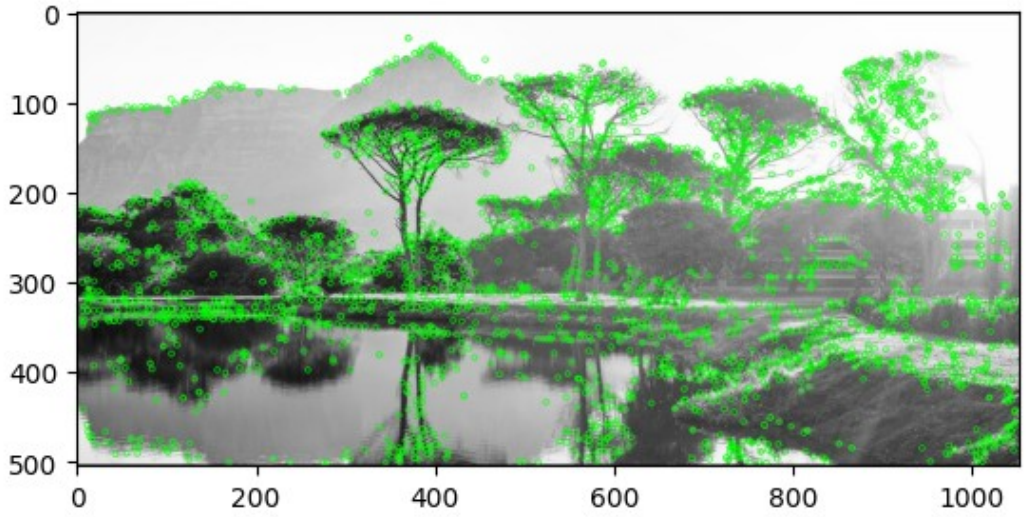
Procedure:

Bag of visual words (BOVW) is commonly used in image classification. Its concept is adapted from information retrieval and NLP's bag of words (BOW). In bag of words (BOW), we count the number of each word appears in a document, use the frequency of each word to know the keywords of the document, and make a frequency histogram from it. We treat a document as a bag of words (BOW). We have the same concept in bag of visual words (BOVW), but instead of words, we use image features as the "words". Image features are unique pattern that we can find in an image.

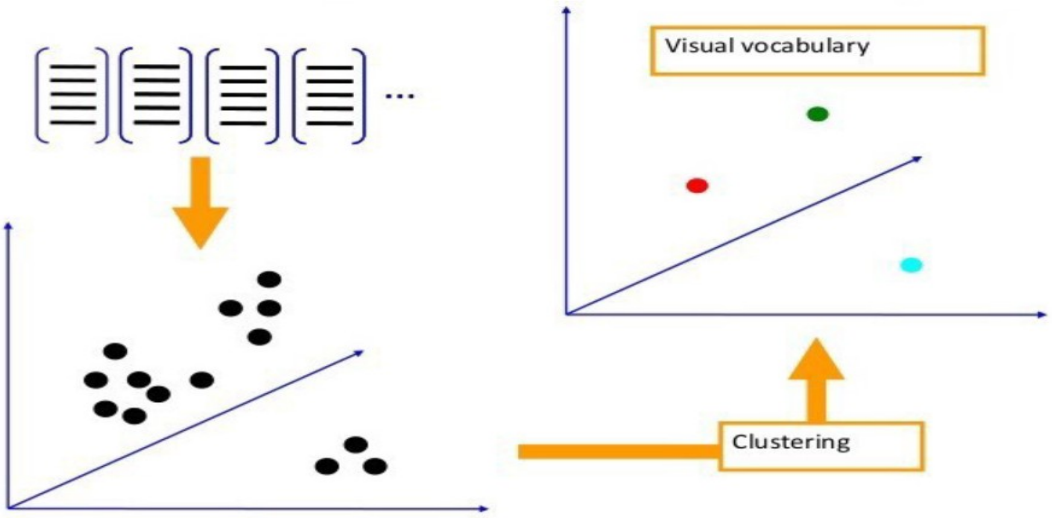
The general idea of bag of visual words (BOVW) is to represent an image as a set of features. The feature of the image consists of keypoints and descriptors. Keypoints are the "stand out" points in an image, so no matter the image is rotated, shrink, or expand, its keypoints will always be the same. And descriptor is the description of the keypoint. We use the keypoints and descriptors to construct vocabularies and represent each image as a frequency histogram of features that are in the image. From the frequency histogram, later, we can find another similar images or predict the category of the image.

Approach:

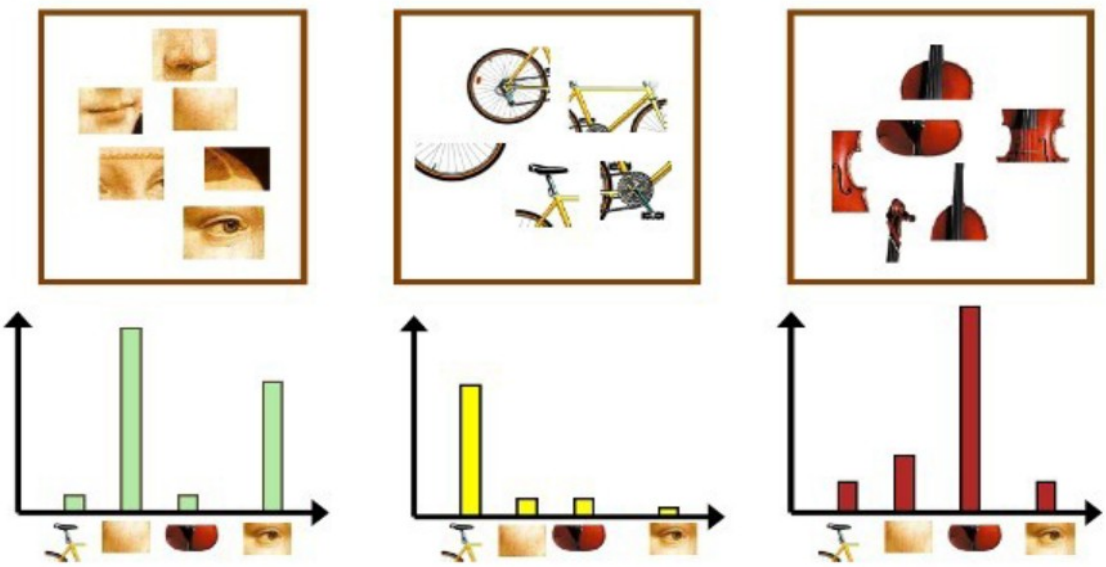
- First Load the Images and their corresponding labels for thr training data.
- Find image descriptors for the images that can be used to identify them.
- Extract local features using SURF/SIFT/ORB descriptors.



- Create a list of descriptors that define a certain class of Images. Then build a Visual Dictionary(a Vocabulary).
- Quantize the feature space i.e. we build the Visual Vocabulary using K-Means to form clusters from the list of descriptors. The center of each cluster is used as the Visual Dictionary's Vocabulary.



- After the vocabulary construction, finally, for each image in both test and training dataset, we make a frequency histogram from the vocabulary and the frequency of the vocabulary in the image by extracting the local features and coparing these features with visual words. Those histograms of training dataset are our bag of visual words (BOVW).



- In this way, an image can be represented by a histogram of codewords. The histograms of the training images can then be used to learn a classification model.

Observations:

Bike V/s Horse Classification:

I have taken k to be 512(had varied it too) in (k means clustering for BOVW). I have used 3 Classification models:

1. SVM
2. Logistic Regression
3. K-Nearest Neighbors

Model	Parameters	Acccuracy
SVM	C = 0.005, kernel = 'linear'	0.9722222222222222
Logistic Regression	max_iter = 1000	0.9722222222222222
K-Neaest Neighbors	n_neighbors = 5	0.9166666666666666

Accuracy Score for corresponding models

CIFAR-10 Classification:

I have taken k to be 512(had varied it too) in (k means clustering for BOVW). I have used 3 Classification models:

1. SVM
2. Logistic Regression
3. K-Nearest Neighbors

Model	Parameters	Acccuracy
SVM	C = 0.01, kernel = ‘linear’, gamma = 0.01	0.2676
Logistic Regression	max_iter = 1000	0.2505
K-Neaest Neighbors	n_neighbors = 5	0.1218

Accuracy Score for corresponding models

- In case of Bike V/s Horse Classification both SVM and Logistic Regression have the same accuracy.
- In Both cases accuracy of K-Nearest Neighbors had less accuracy than both SVM and Logistic Regression.
- Increasing or Decreasing the number of clusters don’t have much impact on the accuracy scores.

Note: For CIFAR-10 Classification I have used only th first batch for training(batch 1), you can change the number of batches from the code.