Visual Recognition Assignment 2

Vignesh Bondugula - IMT2019092

March 3, 2022

Instance Matching and Category Recognition

1 Instance Matching

1.1 Introduction

This part of the assignment involves image panorama stitching using instance matching. I have collected two images at IIITB and was able to stitch them and make a panorama using python and opencv. The key techniques involved are,

- **Key Point Detection:** Apart from edges or corners all the images have few key points which are unique and helps to identify the image. In this process we will be able to find all the key points of the given images. These features are invariant to rotation and scaling.
- **Descriptors:** At each key point we can obtain a 64 or 128 dimensional feature which describes the key point. SURF or SIFT is used to obtain the key points and descriptors.
- **Feature Matching:** We obtain a large number of features from both images. Now, we would like to compare the 2 sets of features and stick with the pairs that show more similarity. We use Brute Force Matcher in open cv to match the pairs.
- **Homography estimation:** Now with the set of common pairs we try to find the transformation matrix that will stitch the 2 images together based on their matching points. Such a transformation is called the Homography matrix. We use RANSAC in opencv to obtain this.
- **Perspective warping:** Once we have the estimated Homography, we need to warp one of the images to a common plane. Here, we are going to apply a perspective transformation to one of the images. Basically, a perspective transform may combine one or more operations like rotation, scale, translation, or shear. The idea is to transform one of the images so that both images merge as one. We use warpPerspective() function in opencv to obtain this.

1.2 SIFT vs SURF

SIFT	**SURF**

• Scale invariant feature transform
• SIFT algorithm can be decomposed into four steps:
  – Feature point (also called keypoint) detection
  – Feature point localization
  – Orientation assignment
  – Feature descriptor generation
• SIFT is 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 be used for object recognition.
• For finding the keypoints, it finds maxima and minima in DoG images and then finds sub-pixel minima and maxima from them using Taylor's series expansion.
	• Speeded up robust features. SURF is a speeded up version of SIFT.
	• SURF algorithm operates very similar to SIFT but it follows a different approach for processing in all the steps. In SURF algorithm can be decomposed into three steps:
	  – Interest point detection
	  – Local neighborhood description
	  – Matching
	• SURF is the speed up version of SIFT. 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. One big advantage of this approximation is that, convolution with box filter can be easily calculated with the help of integral images.
	• For finding the keypoints, It uses Hessian Determinant for this purpose, which helps in expressing the local changes. Then the Haar Wavelet responses are calculated again depending upon the scale similar to SIFT.

1.3 Principles of RANSAC and FLANN Matching

RANSAC:
RANSAC is an algorithm used for estimation of model parameters in the presence of outliers i.e., noise in the data set. RANSAC is mainly used to find the Transformation matrix from different matches which contains many bad matches. RANSAC has the following steps to get the good matches from a

given set.
1. Select a subset from the matches randomly
2. Fit a model on selected Subset
3. Determine the number of outliers
4. Do the given operation for given number of times.

FLANN Matching:
This matcher takes a training description set and calls its nearest search methods to find the approximate match in higher dimensional space. It is faster than BF matcher it the set is large. These methods project the high-dimensional features to a lower-dimensional space and then generates the compact binary codes. Using the binary codes, fast image search can be carried out by using Hamming distance. This reduces the computational cost and optimizes the efficiency of the search.

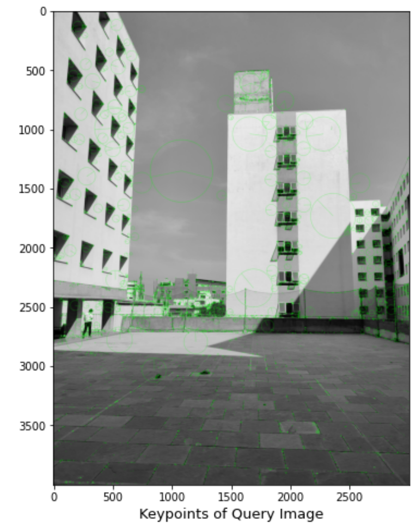
1.4 Implementation

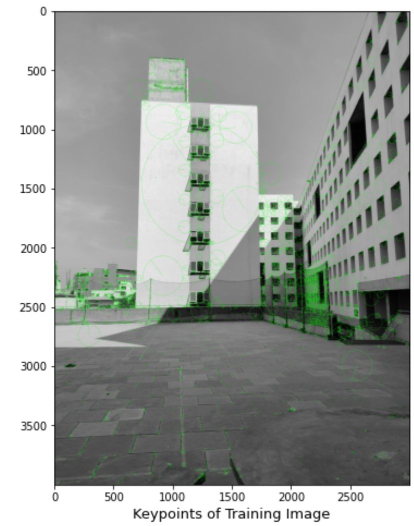
- Loading the images that need to be transformed into panorama using imread() function in opencv.

Query image

Train image (Image to be transformed)

- Extracting key points and descriptors of the images using SIFT and plotting key points on images using drawKeypoints() function in opencv. The below image shows the keypoints in the images.





- Matching the features using BFMatcher in opencv and then plotting the matches using drawMatches() function in opencv. Below image shows the matching keypoints in the images.

- Finding Homography Matrix using findHomography() function in opencv with RANSAC parameter.
- Warping the both images with help of homography matrix obtained using warpPerspective() function in opencv.

1.5 Results

We finally obtain a panorama images which is a stitching of both the input images. The final image is almost accurate but can be made better with post-processing techniques such as cropping, correcting edges etc.

2 Category Recognition

2.1 Bikes v/s Horses - Binary Classification

2.1.1 Introduction

This part of assignment involves binary classification of bikes and horses from the given data set using bag of key points technique. The keys steps in this process are,

- **Key points and Descriptors extraction:** For all the labelled train images we try to detect key points and descriptors using SIFT, SURF in opencv.
- **Constructing Set of vocabularies:** From all the descriptors we extracted, now we try to create a set of vocabulary, where each is a cluster center obtained from Kmeans.
- **Extracting bag of key points:** From the set of vocabularies we created, now for every image we count number of patches that are assigned to each cluster. We can also apply tfidf transformation after extracting the bag of visual words.

- **Applying classifiers:** Applying classifiers such as logistic regression, treating the bag of key points as the feature vector, and thus determine which category to assign to the image.

2.1.2 Implementation

- Read the images and label. Then perform a train-test split using train_test_split() function in sklearn.
- Extracting all the key points and descriptors of the train images using SIFT_create() function in opencv.
- Run kmeans on all the descriptors extracted with 200 clusters. This creates a vocabulary.
- Form bag of visual words by finding descriptors for images and increasing the count to the assigned cluster using kmeans.
- Apply tfidf-transformer on bag of visual words and tried classifying these vectors using KNN, SVM and logistic regression classifiers.

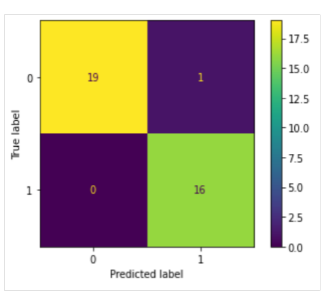
2.1.3 Results and analysis

- I have used SVM, KNN and Logistic Regression classifiers and obtained the following results. Out of all, SVM classifier after running TFIDF transformation gave the highest accuracy.

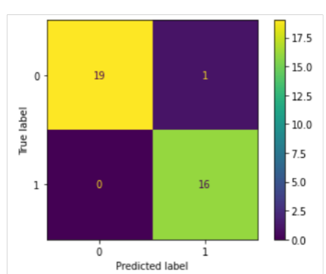
	SVM	Logistic Regression	KNN
Without TFIDF	0.972	0.972	0.944
With TFIDF	1.0	0.972	0.778

Table 1: Accuracies obtained for bike v/s horses classification

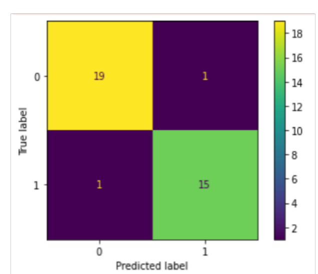
- The following are the confusion matrices obtained by three classifers without TFIDF transformation.



Logistic Regression

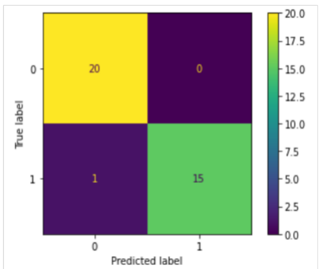


SVM

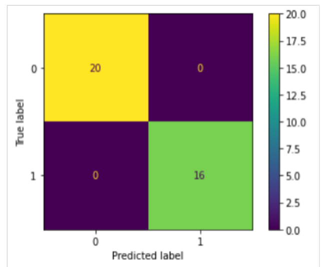


KNN

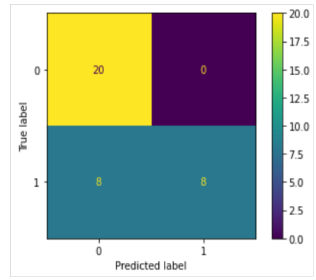
- The following are the confusion matrices obtained by three classifiers after TFIDF transformation.



Logistic Regression



SVM



KNN

- I have also tried testing the model with some images of bikes and horses taken from internet. 5 out of 6 were predicted correct by the best model.

2.2 CIFAR 10 - Classification

2.2.1 Introduction

In this section the techniques used for Bike v/s Horse classification are extended to 10 different classes in CIFAR 10 dataset. The CIFAR 10 dataset contains images of airplane, horse, truck, automobile, ship, dog, bird, frog, cat and deer. It provides train images as well as test images. These images are loaded and similar techniques and steps followed in binary classification are followed to perform classification.

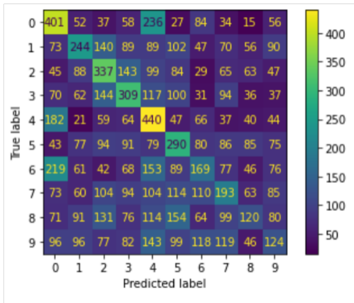
2.2.2 Results and analysis

- I have used SVM, KNN and Logistic Regression classifiers and obtained the following results. Out of all, SVM classifier after running TFIDF transformation gave the highest accuracy.

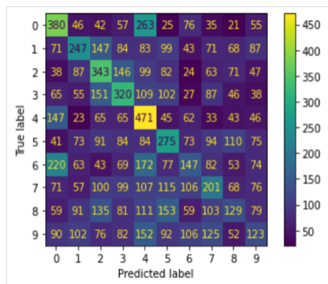
	SVM	Logistic Regression	KNN
Without TFIDF	0.2636	0.2627	0.1874
With TFIDF	0.2653	0.2613	0.1874

Table 2: Accuracies obtained for CIFAR-10 classification

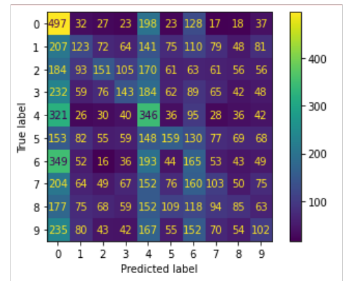
- The following are the confusion matrices obtained by three classifiers without TFIDF transformation.



Logistic Regression

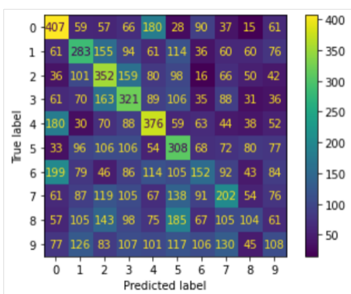


SVM

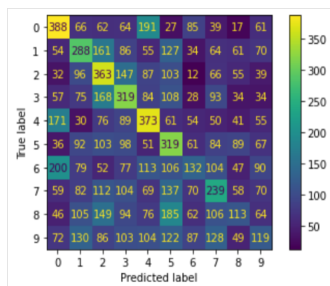


KNN

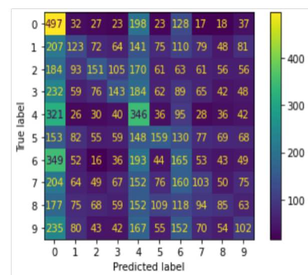
- The following are the confusion matrices obtained by three classifiers after TFIDF transformation.



Logistic Regression



SVM



KNN

- From the confusion matrix we can tell that the model is predicting airplane, truck and ship more accurately than others. Also the results obtained are not accurate enough. The accuracy can be improved exponentially if we use neural networks.