Assignment 1 : Priyanshu Gupta & Shivam Kumar

Hash: for
Final.ipynb - 9a1de2bf1cd42bea6b9489b13b24b766
Retrive-2425afedf2b02c13bfcf7b8dd876c21e
Retrive_test-2a2e6182d1a1896b134d646f158b7144
Preprocesses-7b279262b34fbd57f7e9e2e0f4b830b8

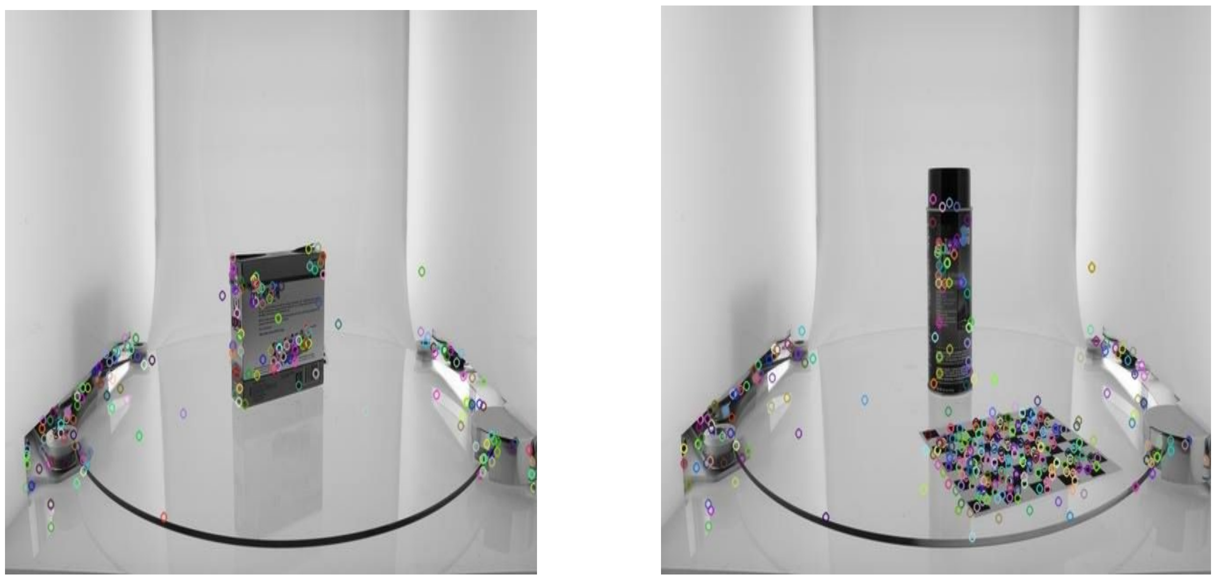
Our current pipeline:

- We build a SIFT descriptor by constructing histogram of frequencies of "visual words".
- We find SIFT features for each image which is a 1D vector of size 128, then concatenate all these 1D vectors into a long 1D vector of the whole training set.
- Use mini batch K-means to cluster these data points from this vector into K groups.
- Now for each image, we have its SIFT features, assigned these features into clusters that we've already clustered before, this represents the histogram representation of frequencies of "visual words" for our image.
- Do this similarly for the rest of the images to form the features representation of our data before feeding into the classifier model.
- Now we use multiclass Support Vector Machines as our classification model.
- Used one-vs-all linear SVMS to operate in the bag of SIFT feature space. The feature space is partitioned by a learned hyperplane and test cases are categorized based on which side of that hyperplane they fall on.
- Now we are combining the results of Tf-idf on train data with SVM score to get the result with more accurate ranking within same classes.

Our progress from beginning:

- Our first and sole inspiration of all we started to think about the pipeline formation was from the slides itself.
- We taught to make a dictionary of words generated from our image. For this we used sift descriptor from OpenCV open source contribute page.
- Earlier we taught the question to be to give proper ranking for all images within same category (that means accounting the orientation of the chessboard present and the recognised object). So we were thinking of not using a classifier as it will not distinguish between it's own subset. But later we realised that we are required to just give higher ranking to relevant images without any weightage within in same class.

- Now our approach was to make bag of visual words from sift descriptor.



Sift descriptor on our dataset.

- We successfully implemented bag of visual words and saved the histogram by using k-means for clustering those descriptors. But we were not sure of what should we choose the value of k for clustering. So we proceeded further by assuming the value of k to be 200.

  Here we were facing 2 major problems.

1. It was taking a lot of time to cluster by K-means so we were being slowed by this and were not able to vary value of k much.
2. The outcomes on sample test case were not satisfying.

- But our problem was solved by using mini batch k-means. It turns out that it is a modified K-Means algorithm which is far more efficient than the original algorithm.It is mostly useful in web applications where the amount of data can be huge, and the time available for clustering maybe limited.

  And now we were able to vary large number of clusters and were getting good accuracy relative to K-means.Now we were using the value of k to be 4000.

- Now as we were thinking , we used SVM to classify the images. But to generate relevance scale ranking we were not able to find any scoring parameters as it was just returning binary output. This problem was solved only when we found a function in sklearn which returned the probability score.
- Till now we were able to generate Tf-idf score and combined both these svm and tf-idf scores to get proper ranking.

- Also there was a problem that our dictionary was storing descriptors randomly and we were able to resolve it by creating a map for each image to its key(Sift Descriptors).