Visual Recognition

Udith Sai .M
IMT2018081

Assignment-2

February 27, 2021

Introduction

In the previous Assignment, we got some basic idea regarding OpenCV library and some image processing algorithms Gaussian Blur,canny edge detection, convolution, Hough Lines, Contour figures. In this assignment, we have applied some of the previous functions like bgr to grayscale image, resizing. And we learn new topics like Homography, Matching lines, Warping of images, BoW, VLAD and also about the CIFAR-10 dataset. There are two parts in this assignment. One is stitching two proper adjacent images to make a panorama and the other is classification of the CIFAR-10 dataset using BoW and VLAD.

Goals

1. **Creating Panorama:** In this part, two adjacent images of a place are given and when we run the Panorama.py program we get the panorama image by stitching the two images in a proper way.
2. **Classification of the CIFAR-10 dataset:** In this part we use Bow,VLAD algorithms to classify the CIFAR-10 dataset.

Creating Panorama

First we need to collect two adjacent images which overlap some parts of the image. Then we need to do a lot of preprocessing before stitching the images. For a test case, we have taken two images of place.

Firstly, Generally i resize the images and convert them to gray scale images. Then we need to find the homography of the images. We do homography using a homography matrix. To find the homography matrix we need to find a minimum of 4 key points in the image. We extracted key points from the two input images using SIFT.The scale-invariant feature transform (SIFT) is a feature detection algorithm in computer vision to detect and describe local features in images. Then after finding key points we draw the points on the original image to check the points.

Then we need to match the descriptors between two images. For that we used FlannBasedMatcher and drew the lines in green colour to identify the matches. We used the

cv::FlannBasedMatcher interface in order to perform a quick and efficient matching by using the Clustering and Search in Multi-Dimensional Spaces module.

Then using Key points and matching lines, we use findHomography() and perspectiveTransform() to get the overlapped image of both. Here we used the RANSAC algorithm for cleaning the noise in the datasets. Then we displayed the overlapped image in grayscale.

Applying a warping transformation using the homography matrix obtained above using cv2.warpPerspective() function. Then we will get the stitched image of two images without cropping.

Then we crop the image sides where there is an empty part. And we will get the final image.

SURF and SIFT

Feature detection and matching are used in image registration, object tracking, object retrieval etc. There are a number of approaches used to detect and match features as SIFT (Scale Invariant Feature Transform), SURF (Speeded up Robust Feature), FAST, ORB etc. SIFT and SURF are most useful approaches to detect and match features because they are invariant to scale, rotate, translation, illumination, and blur. SURF is fundamentally faster, by a larger amount, than SIFT if you were to count FLOPS of two well written implementations. SIFT computes an image pyramid by convolving the image several times with large Gaussian kernels, while SURF accomplishes an approximation of that using integral images. The SIFT algorithm has 4 basic steps. First is to estimate a scale space extrema using the Difference of Gaussian (DoG). Secondly, a key point localization where the key point candidates are localized and refined by eliminating the low contrast points. Thirdly, a key point orientation assignment based on local image gradient and lastly a descriptor generator to compute the local image descriptor for each key point based on image gradient magnitude and orientation. SURF approximates the DoG with box filters. Instead of Gaussian averaging the image, squares are used for approximation since the convolution with squares is much faster if the integral image is used.

FLANN

FLANN(Fast Library for Approximate Nearest Neighbors) is a library for performing fast approximate nearest neighbor searches in high dimensional spaces. It contains a collection of algorithms we found to work best for nearest neighbor search and a system for automatically choosing the best algorithm and optimum parameters depending on the dataset. We used the cv::FlannBasedMatcher interface in order to perform a quick and efficient matching by using the Clustering and Search in Multi-Dimensional Spaces module. FLANN performs a hierarchical decomposition of the search space by successively clustering the input dataset and constructing a tree in which every non leaf node contains a cluster center and the leaf nodes contain the input points that are to be matched.

RANSAC

The RANdom SAmple Consensus (RANSAC) algorithm is a predictive modeling tool widely used in the image processing field for cleaning datasets from noise. RANSAC could be used as a “one stop shop” algorithm for developing and validating QSAR models, performing outlier removal, descriptor selection, model development and predictions for test set samples using applicability domain. Random Sample Consensus (RANSAC) which produces the right result even in the presence of a large number of bad matches.

Classification of CIFAR-10 dataset using Bow and VLAD

Image Classification is a fundamental task that attempts to comprehend an entire image as a whole. The goal is to classify the image by assigning it to a specific label. Typically, Image Classification refers to images in which only one object appears and is analyzed.BoW, VLAD are some of the basic algorithms that are used in image classification.

1. BoW(Bag of Visual Words)

   This is similar to NLP’s BOW. To represent an image using the BoW model, an image can be treated as a document. Similarly, "words" in images need to be defined too. To achieve this, it usually includes the following three steps: feature detection, feature description, and codebook generation. A definition of the BoW model can be the "histogram representation based on independent features".We train and test the model using histogram using KNN. We use SIFT. SURF or any feature extraction algorithm to extract key points and descriptors. Descriptors are the representation of features in the image. Then using K-means we make clusters of descriptors, where the centroid of each cluster represents a visual word.

2. VLAD(Vector of Locally Aggregated Descriptors)

   In simpler terms, we match a descriptor to its closest cluster, then for each cluster, we store the sum of the differences of the descriptors assigned to the cluster and the centroid of the cluster. Recalling BoW, it involved simply counting the no. of descriptors associated with each cluster and creating a histogram for each set of descriptors from an image, thus representing the information in an image in a compact vector. VLAD is an extension of this concept. We accumulate the residual of each descriptor with respect to its assigned cluster.

Firstly we need to load the CIFAR-10 data using Keras library and then extracting descriptors using SIFT and then i passed the descriptors list to Kmeans operator with different k values to get more accuracy and then i applied BoW, VLAD respectively for their file. And these features are passed into KNN for classification.

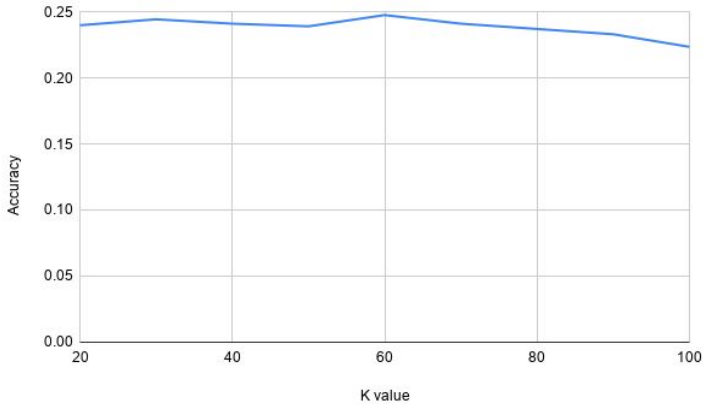
Accuracy for SIFT+K Means (k= 60) +BOW+KNN - 0.2475

Accuracy for SIFT+K Means (k = 30) +VLAD+KNN - 0.3221

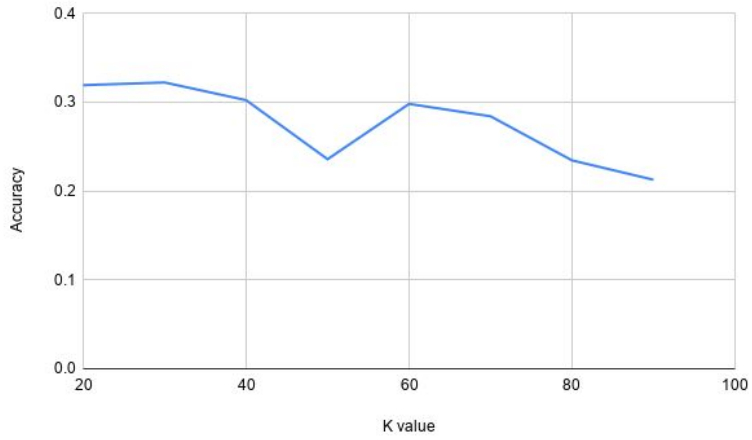
Tuning of k for Bow and VLAD

I considered a different number of clusters in the beginning and then converted them to 10 clusters. Different accuracies for different k values are below.

Bow chart:



VLAD chart:



Conclusion

After completing the assignment, we got an idea about Homography using keypoints and descriptors and wrapping transformation using homography matrix. We also learn about SIFT, SURF, FLANN, RANSAC and also about K Means, KNN algorithms. With each assignment we are getting familiar with visual recognition features.

References

- https://learnopencv.com/image-alignment-feature-based-using-opencv-c-python/
- https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/
- https://medium.com/analytics-vidhya/panorama-formation-using-image-stitching-using-opencv-1068a0e8e47b
- https://www.researchgate.net/publication/314285930_Comparison_of_Feature_Detection_and_Matching_Approaches_SIFT_and_SURF#:~:text=SIFT%20and%20SURF%20are%20most,translation%2C%20illumination%2C%20and%20blur.&text=SIFT%20is%20better%20than%20SURF,integral%20image%20and%20box%20filter.
- https://pub.towardsai.net/what-is-perspective-warping-opencv-and-python-750e7a13d386
- https://ameyajoshi005.wordpress.com/2014/03/29/vlad-an-extension-of-bag-of-words/

- https://medium.com/@shehan.a.perera/a-comparison-of-sift-surf-and-orb-333d64bcaaea
- https://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Machine_Learning_Classification_K-nearest_neighbors_k-NN.php

Acknowledgements

We would like to thank our Professor Dinesh Babu sir for giving us a great headstart in this world of Visual Recognition.