AI 825: Visual Recognition

Instructor: Prof. Dinesh Babu

Assignment 2

Gousepeer Arella

IMT2020042

Solution 1

Explain how SURF is different from SIFT

SIFT – Shift Invariance feature transform

SURF - Speed up version of SIFT

The scale-invariant feature transform is an algorithm used to detect and describe local features in digital images. It locates certain key points and adds descriptors to them to aid in object detection. The descriptors are supposed to be invariant against various transformations which might make images look different although they represent the same objects. These Keypoints are scale and rotation invariant.

The SIFT process is done in the following parts:

- **Constructing a Scale Space:** To make sure the features are independent.
- **KeypointLocalisation:** Identifying the suitable features or keypoints.
- **Orientation Assignment:** Ensure the keypoints are rotation invariant.
- **KeyPoint Descriptor:** Assign a unique identifier to each KeyPoint.

The main features of why **SIFT** is preferable is because the detected local features are Scale invariant, rotation invariant, illumination invariant, viewpoint invariant, and affine invariant. However, they are partially invariant to change in occlusion, clutter or noise.

SURF is the speed-up version of **SIFT**. In **SIFT**, we approximate Laplacian of Gaussian with the Difference of Gaussian for finding scale-space. SURF goes a little further and approximates LoG with Box Filter. One significant advantage of this approximation is that convolution with a box filter can be easily calculated with the help of integral images. SURF is good at handling images with blurring and rotation, but not good at handling viewpoint change and illumination change.

Briefly explain the main principles of FLANN matching and RANSAC

FlANN - Fast Library for Approximate Nearest Neighbors

RANSAC - Random sample consensus

FLANN contains a collection of algorithms optimized for fast nearest neighbor search in large datasets and for high dimensional features. **FLANN** builds an efficient data structure which is a KD-tree that will be used to search for an approximate neighbor. These methods project the high-dimensional features to a lower-dimensional space and then generate compact binary codes. The produced binary codes perform fast image searches to get better computation time and better output.

RANSAC approach is to avoid the impact of outliers, so we look for inliers. If an outlier is chosen to compute the current fit, then the resulting line won't have much support from rest of the points. The RANSAC algorithm works in the following manner: It first estimates the parameters of a mathematical model from a set of observed data which contains outliers. Then, a model is fitted using hypothetical inliers. If other data fits to the model, it is added to the inliers. Then, the model is re-evaluated with the new set of inliers.

PANORAMA CREATION

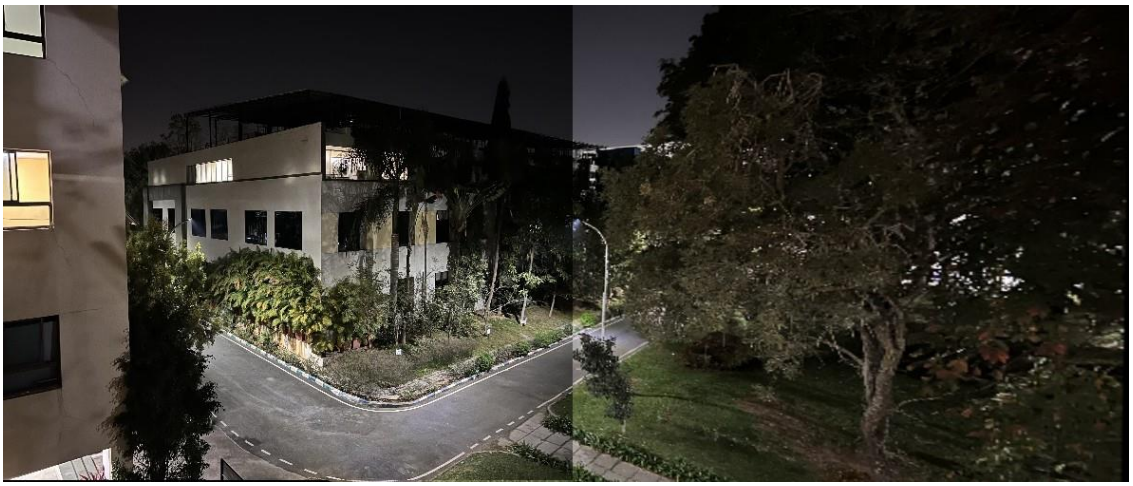
Panoramic photography is a technique that combines multiple images from the same rotating camera to form a single, wide photo. This is done by using a method called image stitching which combines many images by using the common area between them both.

Procedure

Detect keypoints and descriptors: Detecting Keypoints and descriptors is done by using one of the extractors including the sift,surf,orb etc. In this we will use the corresponding extractor's detectAndCompute method to get the required number of keypoints and descriptors of the image. If we want to draw them on the image, we can do this by using cv2.drawKeypoints.

Detect a set of matching points that is present in both images: To detect overlapping areas, we compare the descriptors of the first image with the descriptors of the second image. We also do the ratio test to get the best matches. The ratio test gets rid of the points that are not distinct enough. Basically, we are discarding these matches where the ratio of the distances to the nearest and the second nearest neighbor is greater than a certain threshold.

Stitching two images together: To stitch two images together we use feature-based image alignment. It is the transformations that map features in one image to another. This technique consists of two steps. The first step is to apply RANSAC algorithm to evaluate a homography matrix. Then, we will use this matrix to calculate the warping transformation based on matched features.



Solution 2

Bikes vs Horses

The general idea of bag of visual words (BOVW) is to represent an image as a set of features. Features consists of keypoints and descriptors. We extract all the features of images and then construct a histogram. Then with the histogram we construct the bag of keypoints. By using bag of visual words representation from our dataset, we can compute the test image's nearest neighbors.

I started by putting a SIFT-SURF extractor in place to obtain keypoints and descriptors. By altering the method in the Bag of Keypoints class, we may change the extractor. We next use k-means to obtain the vocabulary of the closest Neighbours using the extracted keypoints and descriptors. Following the execution of the k-means algorithm, the histogram for the retrieved image characteristics is created and used to produce the bag of keypoints.

Using the bag of keypoints, I am using SVM, Logistic Regression, KNN and Naive Bayes models, to perform model training. The obtained model after doing several experiments predicts the test dataset in an accuracy range of 95-100 percent.

There is a bag of keypoints class which has the corresponding methods to perform kmeans, get the vocabulary, and construct the histogram and bag of keypoints.

Observations

SVM with 64 clusters, c = 0.005 and linear kernel – accuracy : 0.9333

LR model with 64 clusters, C = 0.01 - accuracy: 0.9777

KNN model with 64 clusters and 5 neighbours - accuracy: 0.9111

CIFAR-10 Dataset

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.

Since the photos in this case are included in batch files, we will use the pickle module to extract them. The loadBatch method performs this. After that, the batch photographs are reshaped and transposed to place them in the appropriate image files.

We then use sift or surf detectors and extract the keypoints and descriptors. Using the extracted features, we then vertically stack them and perform kmeans on them. From the nearest neighbours output, we create a bag of keypoints which we can then use to train models. The models used are SVM, logistic regression, KNN and Naive Bayes Model. The accuracies are in the range of the 20-30 percent.

Observations

SVM with 256 clusters, c = 0.005 and linear kernel – accuracy: 0.2875

LR model with 256 clusters, C = 0.01 - accuracy: 0.2853

KNN model with 256 clusters and 5 neighbours - accuracy: 0.1472