Project Name : Face identification

Application : If a person uploaded his photograph for a job application when

he comes for the interview we have to specify that he is same one who

applied for the job by comparing the presently taken image by web

camera with the uploaded image

Algorithm’s : SIFT (Scale invariant feature transform) for feature extraction and

Used BFM (Brute-Force matcher) for matching the extracted features.

Programming : Python (Open CV , NumPy , matplot lib)

Language

Metric Used : Accuracy

Model

Description : SIFT which is an inbuilt algorithm in open source computer vision

formally known as OpenCv library in Python which is generally used for

object detection and image identification by extracting the features in

the form numerical vector which are scale invariant.

Algorithm’s Flow Chart :

Creating difference of Gaussian pyramid

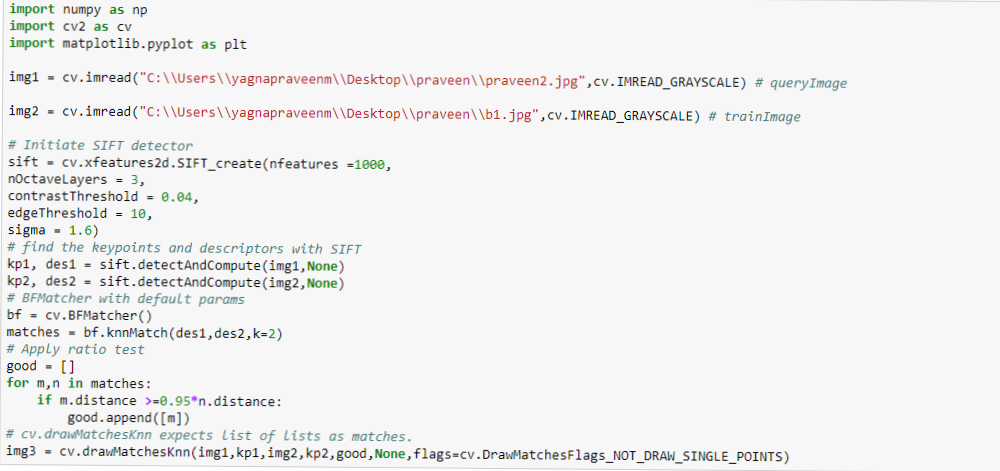
Extrema detection

Noise elimination

Orientation assignment

Descriptor Computation

Key point matching



Above image shows the simple code for our model in which we loaded required libraries first and then we read query image which is taken by web camera and then we read train image which is uploaded by the applicant.

And then we initiated SIFT detector for feature extraction by calling the cv.xfeature2d.SIFT\_creat

Object which contains the hyperparameters like

* **nfeatures** – The number of best features to retain. The features are ranked by their scores (measured in SIFT algorithm as the local contrast)
* **nOctaveLayers** – The number of layers in each octave. 3 is the value used in D. Lowe paper. The number of octaves is computed automatically from the image resolution.
* **contrastThreshold** – The contrast threshold used to filter out weak features in semi-uniform (low-contrast) regions. The larger the threshold, the less features are produced by the detector.
* **edgeThreshold** – The threshold used to filter out edge-like features. Note that the its meaning is different from the contrastThreshold, i.e. the larger the edgeThreshold, the less features are filtered out (more features are retained).
* **sigma** – The sigma of the Gaussian applied to the input image at the octave #0. If your image is captured with a weak camera with soft lenses, you might want to reduce the number.

In this model if the ratio between extracted feature vector of the query images and trained image is greater than 0.95 we consider those features and we sum up those n no no.of features

Based on the requirement we can set up these threshold’s

When I tested this model on different images model gives me 92% accuracy

And I found following pro’s and cons of this model

Pro’s :

* As the name of the portrays the query and trained image need not be equally scaled
* Rotation of the image won’t affect the feature extraction and feature matching
* Even though format of the images is different we can able to compare both images without changing the format
* Low illumination of the image won’t effect the feature extraction.

All of above are key points in our application

Con’s:

* The only time model fails multiple times is when the first uploaded image i.e train image is presently taken one while query image is older one, model is not able to match specified threshold no.features, but in our case this situation can’t arise so no problem with this
* Model throughs error when trained image is less than 8kb for this we can set the limit of image as greater than 8kb.