Final Project Document

Maanav Choubey

Computer Vision CS-6680

Steps to run my code:

1. Download the dataset that is attached along with my source code. The folder's name is "Dataset," and it contains a portion of the coloured FERET dataset.
2. You will need to add the folder that Google Drive will use to execute Google Colab to your Google Drive.
3. You are now able to execute all of the cells by clicking on runtime (It is probably going to ask you to give access to your google drive).
4. The execution of my code takes approximately twenty minutes to complete.

Summary of Important code sections and Functions

- The LocalBinaryPatterns class: In this section, I have developed the holistic Local Binary Pattern histogram by making use of Python's scikit image package and the feature function that comes with it. Using the function local binary pattern, I was able to produce the histogram as well as the lbp image. Once we have the histogram, the next step is to normalize it so that it can be trained.
- The Binarypattern Function: In this function, I perform the mathematical calculation necessary to determine the Local Binary Pattern of our photos in order to obtain the Histogram and the LBP image. In order to calculate LBP, I will be using a neighborhood that is three by three. We are going to use this function independently so that we can compare the outcomes of both of the different implementations and determine which one produces better results.
- LinearSVC: To train and test our model, rather than using K means classification, we are using scikit learn Linear Support Vector Classification. This is in place of using K means classification. It allows for a greater range of customization in terms of the penalties and loss functions that may be applied, and it should be easier to scale up to large numbers of samples.
- The OpenCV LBPHFaceRecognizer requires the input of four parameters: namely, radius, neighbors, gridX, and gridY. We have used an eight neighborhood and a radius of one in our calculations. The value 8 has been selected as the default for either the grid or the window. This function is what we use to construct an improved version of the Local Binary Pattern Histogram. The same function is used for both the training and the testing of our model.
- Scikit Learn Classification Report: We utilize this to determine how well our model did by obtaining the accuracy, the macro average, and the weighted average of the entire testing set. The confusion matrice was selected as the matrix to be used in the calculation of the accuracy.

Test Case -
Training was done with frontal faces of the Feret database images, testing is done on the all the other images of the person with side view, expression and low light.

Changes and Improvements

- While working on the hLBPH, I discovered that mathematically calculating the LBP of each image contributed to an increase in accuracy of one percent. Additionally, employing the SVC classification proved to be useful due to the fact that this class is able to handle dense and sparse input, and the multiclass support is managed using a one-vs-the-rest strategy.
- In the eLBPH technique, the reason to utilize LBPHFaceRecognizer was because it implements the method specified in the paper, but this algorithm also lends weight to the edges of each image. In other words, this eLBPH method pays more importance to the edges. This results in a more accurate selection of features from our photos, which considerably improves the accuracy of the model.
- Despite the fact that the study indicated comparing the accuracies of our system, we really execute facial recognition on a real-world test image to see our findings.

Experimental Results

- Holistic Local Binary Pattern Histogram(scikit learn)- 30%
- Holistic Local Binary Pattern Histogram(BinaryPattern Function)- 31%
- Enhanced Local Binary Pattern Histogram- 74%

Comparison and Analysis

- • When conducting hLBPH in the two different ways that were discussed, we were able to observe in the implementation that the LBP picture that was generated by the BinaryPattern function performed a better job of choosing features out of our image, and it also had a better histogram with fewer variances. The built-in package's generation of the LBP Image and histogram was far more comparable to the findings presented in the research. Surprisingly, the histograms that were produced by our approach were surprisingly consistent with photographs from the front, the side, and other perspectives. The only two bins in our histogram that showed significant variation were the 0 and 255 bins. In all of the other bins, the variation was minimal.
- Both of our hLBPH had more accurate results than those described in the study for both the neighborhood of 8 and the radius of 1.
- The findings that we obtained through the deployment of eLBPH were superior to those that were given in the paper. Before beginning the training process with the LBPHFaceRecognizer, we gave our photos a preliminary processing (resize and interpolation). Additionally, the weight that was given to the edges in LBPHFaceRecognizer was able to assist in improving the feature selection made out of photos.
- Instead of simply comparing the accuracies of each algorithm, we actually implemented it for face recognition in order to acquire a better grasp of the efficiency and effectiveness of our method. This allowed us gain a better idea of how well it performed.

Notes

The link to access the whole FERET dataset is https://www.nist.gov/programs-projects/face-recognition-technology-feret.

Difficulty and Issues

- It was a tough assignment to calculate and extract the Local Binary Pattern from the photos. Because there was no standard function or technique to obtain the histogram out of each image, it was a very interesting process to discover the histogram and plot it out. Also, there was no predefined function or method to get the histogram out of each image.
- While I was training the model, I was required to validate the shape and size of our data at each step, which required a significant amount of testing work and took a lot of time.
- Due to the fact that the code required a significant amount of time to execute, checking the results after each error was an extremely time-consuming process.