Parallelized Local Binary Pattern Approach for Face Recognition System

Abstract

The evolution and advancement in the field of technology over the last few years have been immense and the importance of computer vision and object detection has grown ever since. In computer vision, face recognition is one of the areas which has had prolific growth specifically because of its flexibility and ability to be applied to numerous domains in real world. There are many types of algorithmic implementations for the execution of a face recognition system. However, in order to obtain an extremely accurate and efficient performance, massive training datasets are needed and based on the number of epochs the training runs for, the time required for completion of the process increases drastically. This paper proposes a face recognition algorithm which is called the Local Binary Pattern (LBP) approach. And in order to overcome the major time bound challenge mentioned earlier, the system proposed in this work is parallelized using multiple processes, i.e., a pool of processes, to train the model simultaneously. The paper also gives insights on some of the other existing algorithms with regards to face recognition, and also compares the proposed parallelized LBP model with the conventional algorithms based on its accuracy and performance.

Keywords – face recognition, parallelization, local binary pattern, processes

1. Introduction

With technology advancing at reaching heights of success with respect to numerous applications in various domains, computer vision, face recognition and object detection have played major roles. Face recognition is one of the highly used applications because a human face portrays a number of details and personal information to the viewer and is one of the major points for identification of an individual [1]. In simple terms, face recognition is one of the techniques used to confirm an individual's identity using their face and this can be done either from a static image, a real time video feed or a normal video. In many cases, this technique is even used as tool for biometric security along with other security measures like fingerprint scan, retina scan, etc. The emergence of Convolutional Neural Network (CNN) has truly enhanced the computer vision domain to a great extent [2]. The two main steps involved in face recognition include training the model with subsequent dataset and then implementing and testing the model [3]. However, the existing systems also face a number of minor/major challenges which adversely affects the final system implementation and efficiency. Especially in cases where extremely large datasets are used, the amount of time taken for training the model is massive. In this paper, we have focused on Local Binary Pattern (LBP) approach for face recognition which overcomes one of the major challenges faced with regards to the efficiency, accuracy and speed of some of the existing systems [4]. Furthermore, in order to enhance the algorithm and introduce a unique feature, the algorithm has been parallelized. The LBP approach was initially proposed for its use in the form of an image texture analyzer but currently it's use has become versatile and it's being used in a plethora of computer vision applications. Some of the other techniques which were initially used for face recognition include Principal Component Analysis, Linear Discriminant Analysis, Bayesian Classifier, etc. Majority of the techniques use different type of facial recognition methods like

those of feature analysis, neural network, eigen faces, automatic face processing, etc. with the aim of comparing and contrasting a human face with that of a digital form of face extracted through videos, images, etc. [5] Fig. 1 describes the major technical steps involved in conventional face recognition algorithm. The input frame from the image or video feed is extracted and preprocessed and converted to gray scale. Further, using Haar Cascade featural detections, the necessary facial features are extracted from the detected face which is then classified based on the dataset with which it was trained earlier and any new images are then stored in the database for further learning. These are the commonly used steps for a generic facial recognition system and hence can be enhanced and improvised further for better accuracy and efficiency and faster results [6].

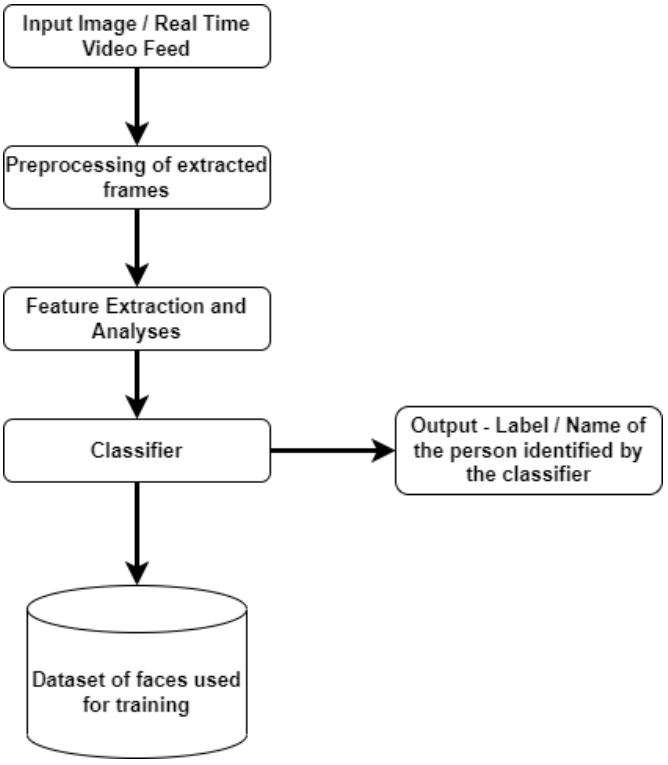

Fig. 1 Work flow of Face Recognition Algorithm

2. Literature Review

Facial recognition is one of those applications which has evolved over the years with the development and evolution of various algorithms and technical enhancements. There are different techniques for implementing a face recognition algorithm and each of these make use of various statistical, stochastic and similar approaches for the implementation [7]. One of the early face recognition algorithms was the Eigenfaces algorithm which completely relies on statistical approaches and it works by pulling out the principal aspects and elements which are likely to introduce a drastic change to the image in their absence. Each class in the training dataset represents a particular individual who is required for identification and every pixel in the picture is considered as a dimensional representation [8]. One of the other face recognition algorithms is

Fisher face which uses a holistic approach for identification and detection by modifying the above mentioned Eigenfaces algorithm and hence, it also makes use of Principal Component Analysis. This algorithm mainly makes use of the linear discriminant analysis to differentiate the various images and frames from a different class [9]. One of the other algorithms commonly used for face recognition is Deep Neural Networks which makes use of the neural networks being divided into various layers with an input layer and a corresponding number of hidden layers followed by a final output layer. These neural networks are used for clustering and gathering the data together based on the similarities [10]. Another extension of this sort of algorithm is the Convolutional Neural Networks which is used for detecting and analyzing 2D images in the form of inputs. Each pixel in the picture is represented by a number between -1 and 1. -1 is a dark pixel, and 1 is a bright pixel, 0 is gray. By observing figure 8 which represents an "X" we can notice three features which are used to draw the "X". There is a pattern representing a backward slash, another one a forward slash and the last one is a cross [11]. These patterns are used as filters and will try to be matched through the pictures. Then comes the next face recognition algorithm which is Rectified Linear Units Layer which is highly recommended for any sort of activation functions and extractions. Apart from these, there also exists many other detection algorithms like that of You Only Look Once, Single Shot Detector, etc. [12]. Elastic Bunch Graph matching is one of the other commonly used face recognition techniques which completely relies on the estimation of a bunch of features using the bunch graph data structure wherein the features are retrieved by making use of various filters called face meshes or face graphs. Similarly, template matching is one of the other recognition algorithms which works on similar grounds and it utilizes the geometrical based approaches for the same [13]. The interesting thing about this algorithm is that it is quite simple and flexible to handle. Last but not the least, the geometrical feature matching algorithm is also used for the very same objective of achieving face recognition. It focuses on analyzing and detecting the minute characteristic aspects of facial areas on the basis of which a Bayesian classifier is used for further identification and recognition [14]. The above-mentioned algorithms are the ones we found to be some of the notable and highly used face recognition techniques and are the ones which are used in existing applications and professional domains.

3. Proposed Work

Based on the survey and analysis done so far, we were able to move forward with the LBP approach. However, in order to tweak it and increase the efficiency of the existing LBP algorithm, the system proposed in this paper is implemented in a parallelized manner with multiple processes executing the training and detection of faces simultaneously. One of the major pointes to be noted her is that face detection and face recognition are often interchangeably used even though they have significant differences between them. Face detection focuses on locating and detecting the face from an extracted frame of the image. On the other hand, face recognition is an approach in which the facial images are preprocessed and converted to gray scale for further application of the face recognition algorithm. The LBP approach is used for the facial recognition concept and is a simple yet powerful and efficient textural operator which makes use of labelling the pixels in the extracted image frame by threshold analysis of the neighboring and adjacent pixels and by optimizing the results using a binary statistical perspective. There are a number of steps involved in the implementation of this facial recognition algorithm. The first step is to consider the various

parameters it makes use of for analysis and implementation. One of the parameters is the radius which is used to curate the circular binary pattern for the localized aspect and it is conventionally set to 1. The next parameter is the neighboring pixels followed by the parameters of Grid X and Grid Y which indicate the number of cells to be considered in the horizontal and vertical axes respectively. The next step involves training of the algorithm which involves the process of sourcing a suitable dataset with the facial images of all those people who are required to be identified using the algorithm. An ID and name are set for each of the images according to the class pertaining to the respective individual. Following this is the application of the LBP operation. The very first step involved in this process is develop an intermediate pixelated image which can be used to enhance the original image further by highlighting the features and characteristics of the detected faces. In order to accomplish this, the above-mentioned parameters are made use of. After acquiring the enhanced gray scale image of the original image, the picture is converted into a nxn matrix of pixels. Each of the cells in the matrix represents the intensity of the pixel values which lies between the range of 0 – 255. Once this has been computed, the next step is to identify the central most value as the threshold. Using this threshold, the each of the adjacent cells of the pixelated matrix will be set to binary values. One is set in those cells which have its pixel values greater than or equal to the threshold and zero for those cells which have a lower value than the threshold. From the binary matrix which has been created, the cells are concatenated together line by line to achieve a new binary value which is considered as the final output for further computation [15]. The computed binary value is then converted to decimated integer and is the value of the centralized cell in the matrix which represents a unit pixel from the original image. The same LBP operation is carried out for each and every pixel of the gray scale image and the resultant image obtained from the set of newly computed pixels would have highly enhanced and better features with regards to the characteristics and features than the original plain image. Fig. 2 represents the computational process of converting a grayscale image to the final decimated matrix. The final step is to compute the face recognition using the newly created matrix. Using the previously trained model, the input frame which is extracted from the image or video feed is firstly compared and matched with the images in the dataset and the class label which closely matches the input image is returned as the identified result. The algorithm also returns the distance which is the difference between the input image and the image in the dataset and this acts as the base for measuring the confidence. Confidence measure is used to indicate the probability with which the predicted result is close to the actual result. In order to improve accuracy, the confidence threshold used is 0.5, i.e., 50%. So only if the confidence level is greater than 50%, will the predicted class label be displayed and if not, it will be demeaned.

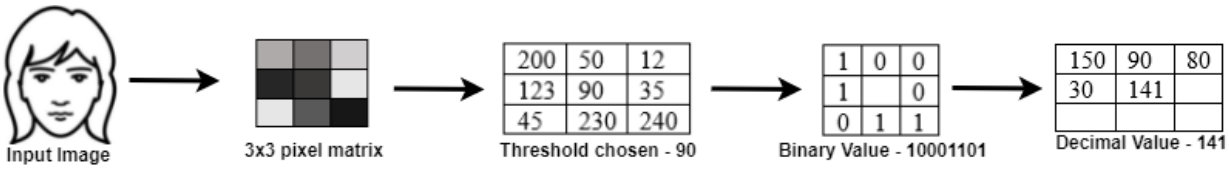

Fig. 2 Computation process in the LBP approach for Face Recognition

One of the major challenges in the beginning was to train the model with sufficient dataset for accurate identification and this took a lot of time as it had to be run for at least thirty epochs to

avoid and reduce the possible lossless training. The time taken to trin huge datasets was one of the major hurdles because of the time constraint. In order to overcome this bottleneck, we have parallelized the above-mentioned LBP code [16]. Parallelization is the essence of curating an algorithm or implementation which would process the data and dataset parallelly or simultaneously. Conventionally, programs are executed sequentially in a serial fashion. The main objective of parallelizing the algorithm is to ensure that it breaks down the large chunk of processing task into smaller pieces which each of which would be carried out by independent processes at once. The main advantage of parallelization is to reduce the idle time and to increase the optimizations and speed of execution of the tasks. We have used Python as our base language for implementing the face recognition algorithm and hence have used the libraries and packages in python for parallelizing the training process which would help the simultaneous execution of various parts of the code. The *multiprocessing* package is used to import *process.* The number of processes is set based on the load of the dataset to be trained and a pool of processes are created accordingly. In order to acquire maximum utilization of multiprocessing, we have taken care of the following pointers. To get the full speed benefit from parallelization, we divided the program into discrete chunks of work, each of which can then be delegated to different threads or processes. We also structured our application so that the parallel tasks don't step over each other's work nor create contention for shared resources such as memory and input/output channels. Fig. 3 and Fig. 4 indicate the complete flow of the above-mentioned parallelized LBP approach.

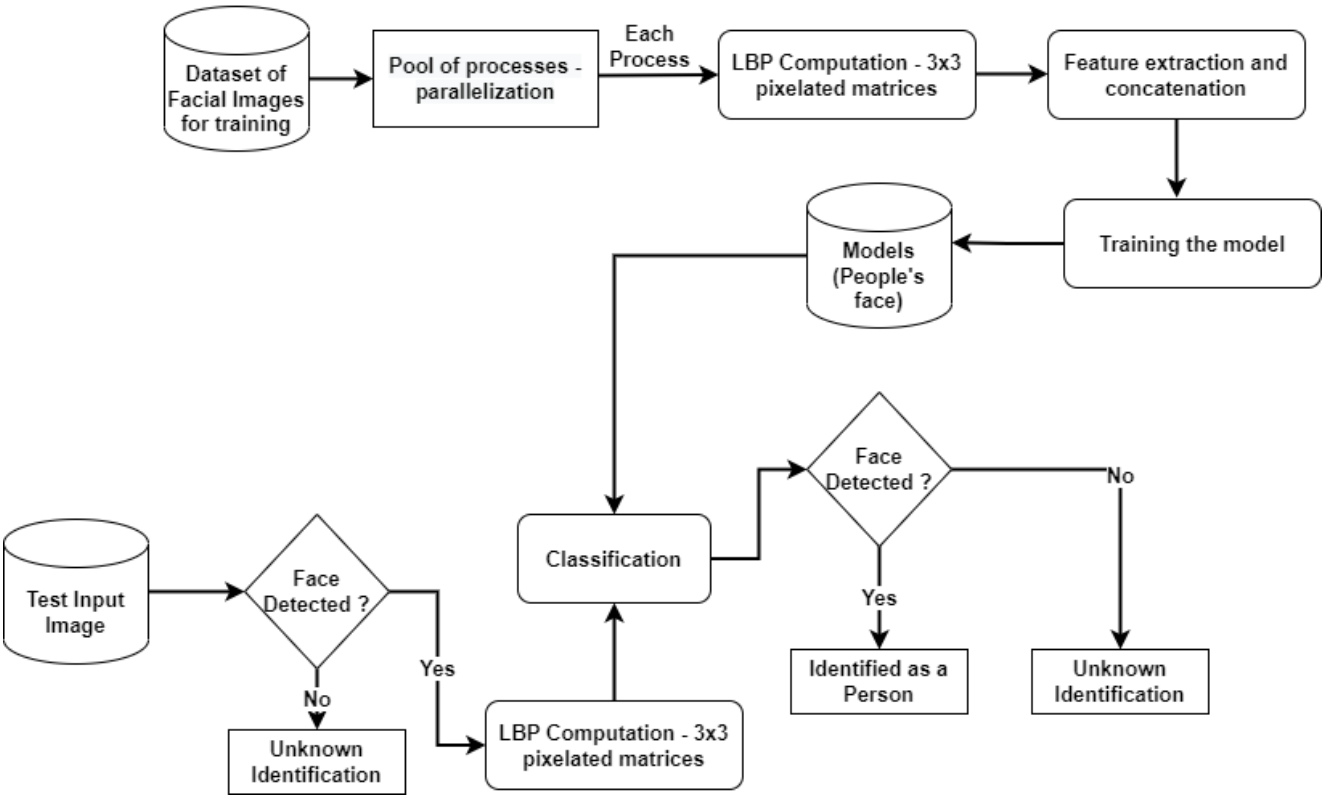

Fig. 3 Work flow of Parallelized LBP Face Recognition Algorithm

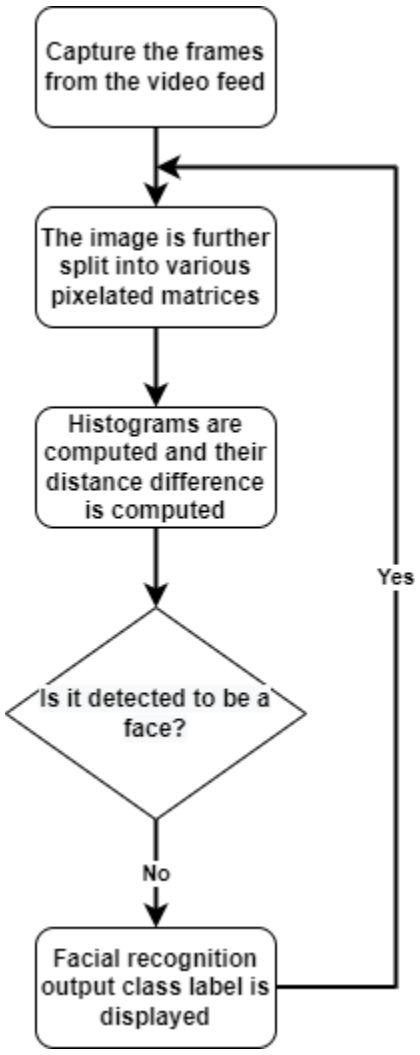

Fig. 4 Work flow of computation in LBP algorithm

```
def trainimg():
    num_processes=len(names)
    p = mp.Pool(num_processes)
    p.map(process_video_multiprocessin, range(num_processes))
    for i in p:
        i = multiprocessing.Process(target=trainimg)
        recognizer = cv2.face.LBPHFaceRecognizer_create()
        global detector
        detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
        try:
```

```
        global faces,Id
        faces, Id = getImagesAndLabels("dataset")
    except Exception as e:
        print(e)
    i.recognizer.train(faces, np.array(Id))
    try:
        i.recognizer.save("model/trained_model2.yml")
    except Exception as e:
        print(e)
combine_output_files(num_processes)
res = "Model Trained"  # +",".join(str(f) for f in Id)
```

Algorithm used for parallelized LBP face recognition approach

4. Results

The proposed algorithm mentioned above was implemented with a sufficiently large dataset for identifying and recognizing the faces of the authors of this paper. On implementing the parallelized version of the code, it was estimated to have trained in very less time with hardly any data losses. In order to compare the efficiency and performance of LBP face recognition with the other algorithms, we have implemented some of the other face recognition algorithms mentioned earlier in the paper which were surveyed. The efficiency and performance of these algorithms have been consolidated with the results obtained from that of LBP and are mentioned in the table below.

Face Recognition Algorithm	Recognition Accuracy (%)
PCA	89.4
LDA	91.25
Eigen Face	88.44
R-CNN	92.5
LBP	93.4

Table 1

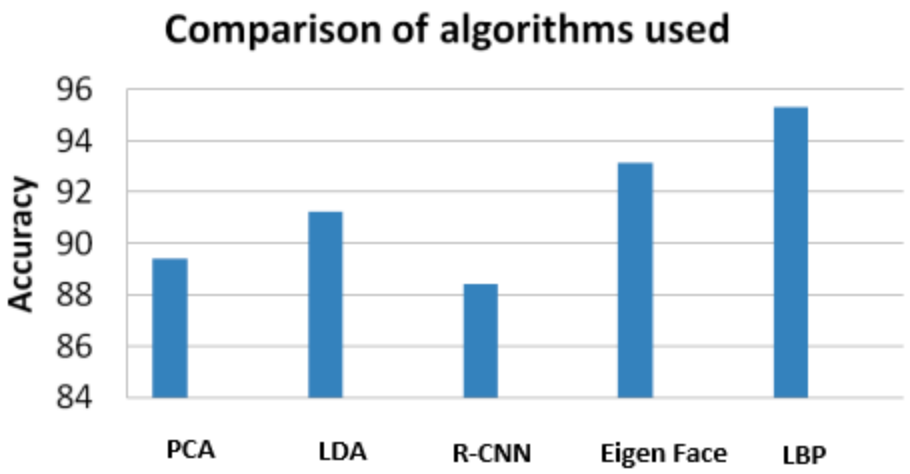

Fig. 5 Comparative accuracy of the various algorithms surveyed

From Table 1, it can be clearly observed the LBP is the algorithm with the highest recognition accuracy. Furthermore, during implementation the frames per second was comparatively higher for LBP when compared to PCA, LDA, and the convolutional neural network algorithms. Furthermore, the graph portrayed in Fig. 5 indicates the comparative accuracy of the various algorithms which have been surveyed so far. Given below in Fig. 6 and Fig. 7 are snapshots of the implementation of the LBP algorithm developed. Fig. 6 is the snapshot of the UI which opens up on running the train_model.py file. We enter the id of the person who's going to be trained followed by the name of the person to be display on confident identification. Then the "Take Images" button is clicked which would start a video capture and take sufficient number of images from the real time video feed frames and will store it in the dataset folder with the respective id and name of the person. Then the "Train Images" button is clicked which would train the model with all the acquired dataset images and frames. Once the training has been done for around thirty epochs, a notification pops up saying "successfully trained the model" if the model has been trained without too much of a lossy training. The training is done parallelly using multiple processes and hence takes hardly a minute or two to finish the training process completely. The next step is to run the face recognition code. Fig. 7 displays the output of the face recognition where it has correctly identified the people as Meghna and Harika and has bounded the faces with green boxes respectively.

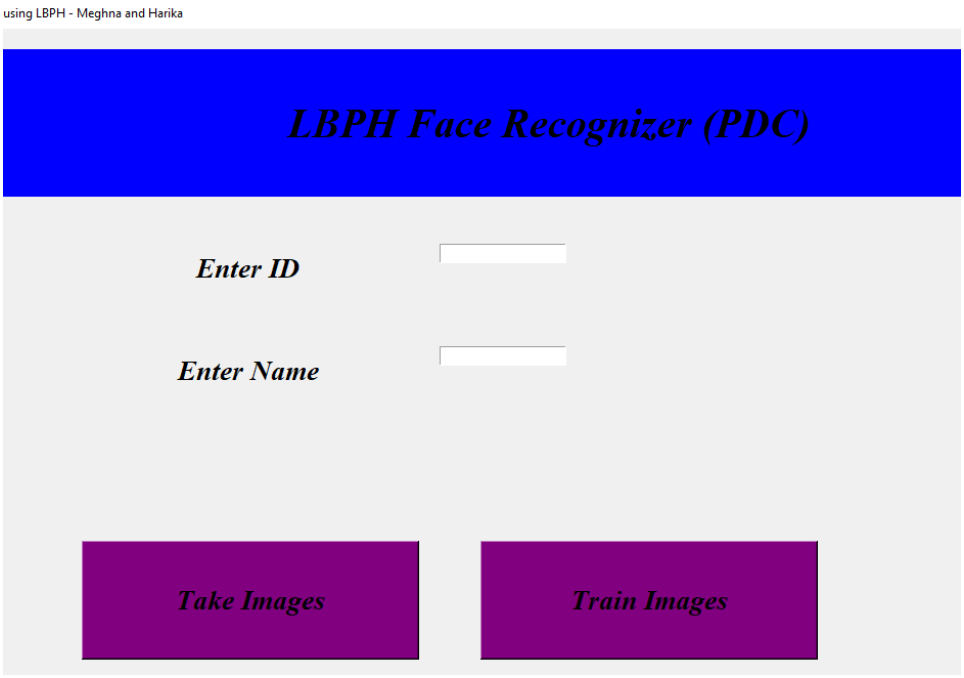

Fig. 6 Training the model

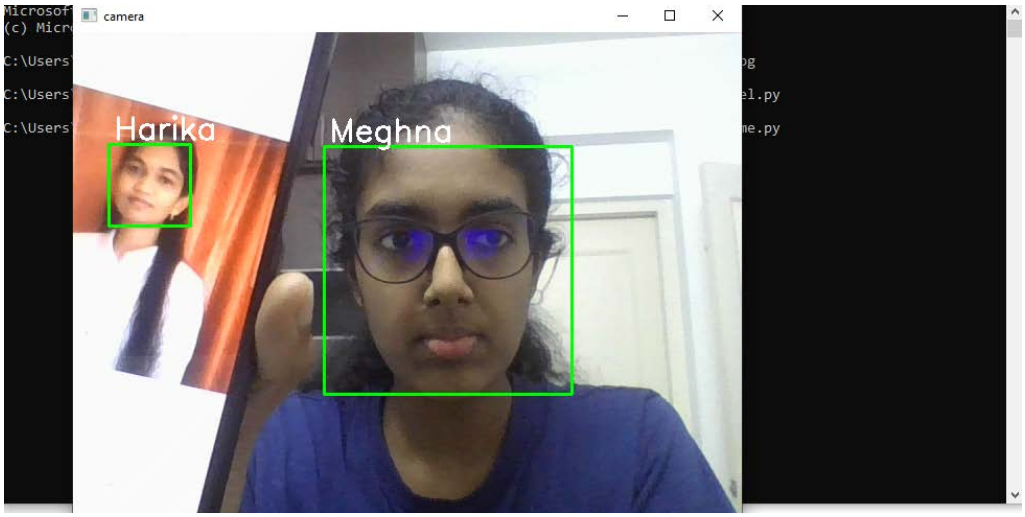

Fig. 7 Implementing the parallelized LBP recognition algorithm

5. Conclusion

From the implementation of the parallelized LBP face recognition algorithm and on comparing it with some of the existing face recognition algorithms we could infer that LBP is one of the most flexible and versatile algorithms which is easy and simple with regards to the aspects of computation and processing [17]. The fact that it can represent the local features of the detected face in an enhanced and highlighted way further imposes its accuracy and performance. It's highly robust and strong when compared to the traditional PCA, LDA or Eigen Faces algorithms especially because of its ability to run well against monotonic gray scale transformations.

Furthermore, the OpenCV library in python even offers built in modelling function for LBP Histogram approach implementation which can be used for easing out the coding and modularity of the code. The proposed face recognition algorithm has its application and use case in a plethora of domains. One of the major areas in which it can be implemented is in the area of criminal surveillance and related identification. The efficiency and accuracy of the algorithm can help concerned authorities detect and identify criminals in country/state border checks, airports, etc. It can be extensively used for forensic investigations and aid the concerned professionals and experts in easing out their tasks [18]. With an increasing use of social media platforms and virtual communities, this algorithm can also support the approach of identifying people in various posts and videos. And most importantly, it can even be used to track the activities and attendance of students in a university or college. The fact that the proposed parallel LBP algorithm can detect multiple faces and recognize them simultaneously is one of the major features of attraction for the applications mentioned above. Hence, it can be concluded that the implemented algorithm has numerous domains of applications and has been successfully implemented [19].

References/Bibliography

[1] He, X., Yan, S., Hu, Y., Niyogi, P. and Zhang, H.J., 2005. Face recognition using laplacianfaces. IEEE transactions on pattern analysis and machine intelligence, 27(3), pp.328-340.

[2] A.U. Batur and M.H. Hayes, “Linear Subspace for Illumination Robust Face Recognition,” Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition, Dec. 2001.

[3] Parkhi, O.M., Vedaldi, A. and Zisserman, A., 2015. Deep face recognition.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, pages 1106–1114, 2012

[5] C. Lu and X. Tang. Surpassing human-level face verification performance on lfw with gaussianface. AAAI, 2015.

[6] Lahdenoja, O., Maunu, J., Laiho, M. and Paasio, A., 2006, May. A massively parallel algorithm for local binary pattern based face recognition. In 2006 IEEE International Symposium on Circuits and Systems (pp. 4-pp). IEEE.

[7] Delbiaggio, N., 2017. A comparison of facial recognition’s algorithms.

[8] Amos, B., Ludwiczuk, B., & Satyanarayanan, M. (2016). OpenFace: A general-purpose face recognition library with mobile applications, (June).

[9] Barnouti, N. H. (2016). Improve Face Recognition Rate Using Different Image PreProcessing Techniques Nawaf Hazim Barnouti American Journal of Engineering Research (AJER), (4), 46–53.

[10] Geitgey, A. (2016). Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning. Retrieved 11 May 2017

[11] Nair, V., & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. Proceedings of the 27th International Conference on Machine Learning, (3), 807–814. https://doi.org/10.1.1.165.6419

[12] Wilson, P. I., & Fernandez, J. (2006). Facial Feature Detection Using Haar Classifiers. Journal of Computing Sciences in Colleges, 21(4), 127–133.

[13] King, D. E. (2009). Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research, 10, 1755–1758. https://doi.org/10.1145/1577069.1755843

[14] http://ijcttjournal.org/Volume5/number-4/IJCTT-V5N4P136.pdf

[15] K.; Subban, R.; Krishnaveni, K.; Fred, L.; Selvakumar, R.K Nallaperumal, "Human face detection in color images using skin color and template matching models for multimedia on the Web", Wireless and Optical Communications Networks, 2006 IFIP International Conference on 11-13 April 2006, pp. 5-10

[16] Jigar M. Pandya, Devang Rathod, Jigna J. Jadav," A Survey of Face Recognition approach", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 1, January -February 2013, pp.632- 635

[17] A. S. Tolba, A.H. El-Baz, and A.A. El-Harby, " Face Recognition: A Literature Review", International Journal of Signal Processing 2:2 2006

[18] Jyoti S. Bedre ,Shubhangi Sapkal, "Comparative Study of Face Recognition Techniques: A Review", Emerging Trends in Computer Science and Information Technology – 2012(ETCSIT2012) Proceedings published in International Journal of Computer Applications® (IJCA) 12

[19] I.J. Cox, J. Ghosn, and P.N. Yianios, "Feature-Based face recognition using mixture-distance," Computer Vision and Pattern Recognition,1996