LOW-COST REAL-TIME FACE RECOGNITION SYSTEM USING LOCAL BINARY PATTERN HISTOGRAM ALGORITHM

***Dr.Shailendra Kumar Dhakad*[1]*., Dr.I.Selvamani*[2] *., K.Kavitha Rani*[3]*.,K.Tabitha*[4]*., P.Padma Poojitha*[5]*., P.Sathvika*[6]**

1, 2 Professor, Department of ECE., Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India (i.selvamani@gmail.com)

3, 4, 5, 6 B.Tech ECE, (19RG1A0479, 19RG1A0482, 19RG1A04B3, 18RG1A0494),

Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India

Abstract:

This research presents a novel method for real-time face recognition, one that is sensitive to variations in pose, expression, and even picture size (as little as 15 pixels). Both LRD200 and LRD100, two of our datasets, have been used in training and classification efforts. After the face image has been detected using the Viola-Jones algorithm, it sent on to the recognition stage, where it is recognised using the Local Binary Pattern Histogram (LBPH) algorithm after some reprocessing is performed using contrast limited adaptive histogram equalisation (CLAHE) and face alignment. Our dedicated Android app makes it possible to refresh the system's face database, at which point the system will resume its training and recognition procedures with the new data. Our suggested technique has achieved 78.40% accuracy at 15 pox and 98.05% accuracy at 45 pox when applying it to the LRD200 database, which comprises 200 images per individual. With a database of 100 images per person, an accuracy of 0.60 percent was achieved at 15 pox and 95 percent at 45 pox (LRD100). With a facial variation of around 30 pox on each side from the front face, the average face recognition accuracy was between 72.25 and 81.85 percent. This facial recognition technology may be employed by law enforcement even when a person is too far away for a security camera to take a clear photograph. It might be used as a surveillance system in public areas to deter criminal activity, such as airports, bus terminals, and other transportation hubs.Some keywords that might be used to locate this article are low-resolution, feature extraction, and face detection or recognition. The article will be published in the April 12, 2021 issue of Optics & Photonics Journal (volume 63, issue 14; DOI: 10.4236/opj.2021.114005).

Introduction

The process of identifying a specific face or faces in a photograph or collection of photographs is known as face recognition. Humans are much superior to machines in facial recognition and detection. Video surveillance, security, access control, law enforcement, general identity verification [1], gender recognition [2], missing person identification, and so on all depend on facial recognition technology. Face recognition is a primary focus throughout the development of such programmes. Facial recognition may be used for either verification or identification purposes. "Face verification" refers to a technique that might

potentially tell whether two images of the same person are of the same person. Yet, in order to successfully identify a face, it must be compared to a library of labelled images. Face recognition may be accomplished in one of three ways: 1) using a holistic matching methodology; 2) using a structural method; or 3) using a hybrid approach. Methods for comprehensive face recognition include Eigenfaces, principal component analysis, and linear discriminant analysis [4, 5, 6]. In contrast, structural methods of identification make use of native statistics and more fundamental features like a person's nose, mouth, and eyes. An example of a hybrid strategy would be one that combines structural and holistic features. A face recognition system works by comparing an uploaded photo of a face to a library of "trained" photos. During the previous several decades, many methods for recognising human faces have been created. The Viola-Jones algorithm [7], one of the most well-known face detection algorithms, is a possible ancestor of many distinct approaches to face recognition. This technique makes use of a face detecting algorithm that makes use of Haar-like rectangular features. Their results for a 320x240 pixel image are dismal in compared to the software-based system operated on CPUs. Using Haar features with the AdaBoost algorithm, Chakras Ali and Kitale [8] were able to successfully identify human faces. Every facial recognition system relies heavily on its users' ability to identify individuals in the system's database. Many different recognition systems have used holistic facial recognition techniques. In [9], Principal Component Neural Networks (PCNNs) for recognising faces are shown. This device is perfect for access control and monitoring since it can recognise up to 1400 individual faces in a single photograph. With just 32x32-pixel pictures, they demonstrated a solution for real-time face recognition. While the cited source [10] can process low-resolution photos, no mention of a real-time application is made (60 by 60 pixels). Endure et al. reported a PCA-based real-time embedded facial recognition system in which the minimum image size required for recognition was 320x340. This system can only support a maximum of two users at once. Recent work by Schaffer et al. [12] developed a system for face recognition that uses a hybrid of the Viola-Jones face identification approach and the PCA methodology, both of which are implemented in software (MATLAB), and on an FPGA, respectively. The company claims it can do a real-time facial recognition accuracy rate of 95% and analyse 13,026 faces per second. There are already 153 people in the system's database, and 20 images of their faces have been reviewed [13, 14]. A MLBPH-based real-time face recognition system was presented by Zhao and Wei [15], who showed that it could recognise faces in different poses and orientations with an accuracy of 48% to 55% for facial regions at 30 degrees of rotation. The LBPH framework was used by Ahmed et al. [16] to achieve low-resolution facial recognition (35 pox). Yet, they claim 94% efficiency at 45 pox and 90% efficiency at 35 pox, with an average of 500 photographs per person, in the identification process. A GPU-based LBPH algorithm was proposed for facial recognition in [17]. We recommend combining the LBPH algorithm with face alignment and CLAHE for improved recognition precision in real time. As a result, we achieve higher accuracy in face recognition while using a smaller database. Our results show that 15 pixels of the input image is enough to improve the accuracy of face identification regardless of the subject's mood or degree of facial deviation.

Detailed Plan for Human Face Recognizability

In this section, we introduce the Viola-Jones [7] face detection algorithm. As can be seen in Figure 1, the Viola-Jones technique employs Haar-like rectangular features in the construction of its classifiers. The brighter area just above the cheekbones in Figure 2 may be used to distinguish the darkish area that covers a human's eyes. As the nasal junction is brighter than the two chick sides, the Haar characteristic f shown in Figure 1 may be used to locate the nose (Figure 2). Integral to this method is the calculation of rectangle features for the whole image. There's a certain way the whole thing is put together, and that's how

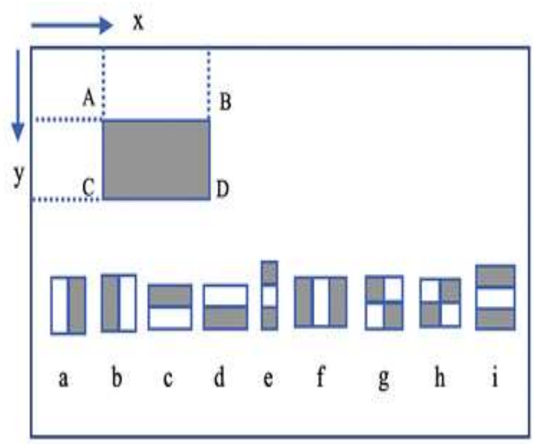


Figure:Block diagram of the system.

Figure 2. Relevant Haar feature for face detection.

This indicates that at the coordinates (x, y), the whole image includes all of the pixels to its left and up above (x, y). A cascade classifier, constructed in stages, is used in this method. At each stage, a set of selected rectangular features is pushed over a smaller window to identify whether or not a face is present (Figure 3). A threshold test is used to decide whether or not a more condensed region of the window should be disqualified as a face candidate. The method involves superimposing a pyramid of ever smaller images over the original until all faces are found; these images have the same set of rectangular properties but are all progressively smaller. All the faces are marked off in red squares in the original test image.

Traits Disconnection Example of the LBPH Algorithm in Use

A powerful feature for texture classification in computer vision, Local Binary Pattern (LBP) is a simple yet incredibly fast operator to describe a pixel's contrast information with regard to its surrounding pixels. Labels are assigned to image pixels by calculating a threshold from the values of their neighbouring pixels and then using that number as a binary value. Histograms of Oriented Gradients (HOG) is a descriptor that dramatically improves LBP's performance on certain datasets. An image of a person's face may be represented as a simple data vector for use in a face recognition system thanks to the combination of LBP and the HOG descriptor. Figure 4 shows that the Local Binary Pattern Histogram (LBPH) face recognition algorithm operates on a grayscale version of the facial image. In order to extract features from the grayscale image, as shown in Figure 5, the image is first divided into 3x3 window cells, with the central pixel of each cell being compared to its neighbours in both the clockwise and anticlockwise directions. When the surrounding pixels are bigger than the core pixel, the central pixel is changed to a 1, and vice versa when the surrounding pixels are smaller. An 8-bit binary number is created by replacing the centre pixel of the original cell with the decimal representation of the binary number obtained by counting clockwise around the resultant 3x3 window. This number represents the local texture. The 8-bit LBP code for the central pixel at coordinates (x, y) may be computed by plugging in the values for go and the neighbouring pixels (g0, g1, g2,, gp1) into Equation 1. (1).

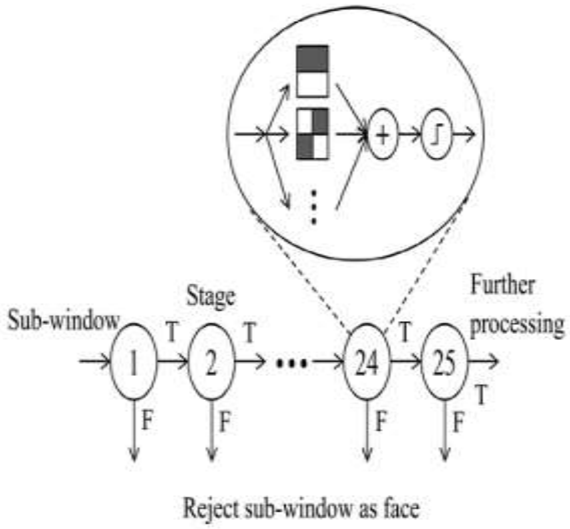
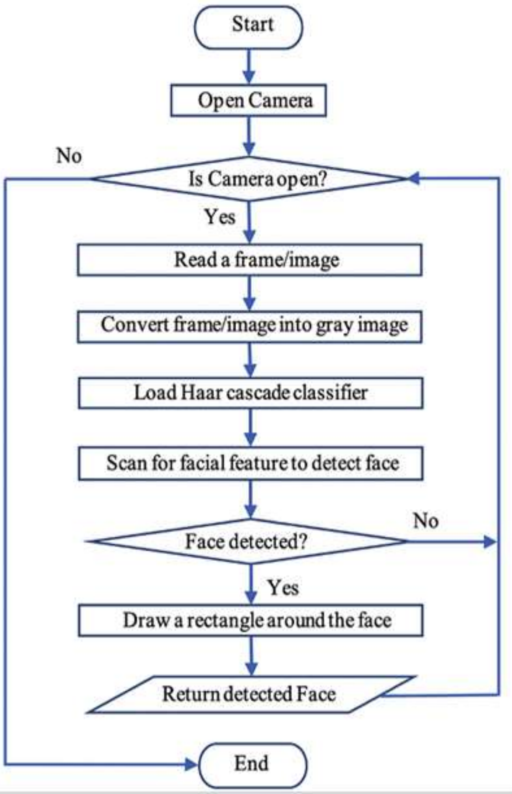


Figure 3. A cascade classifier of 25 stages & decision tree (T = Ture, F = False) [11].



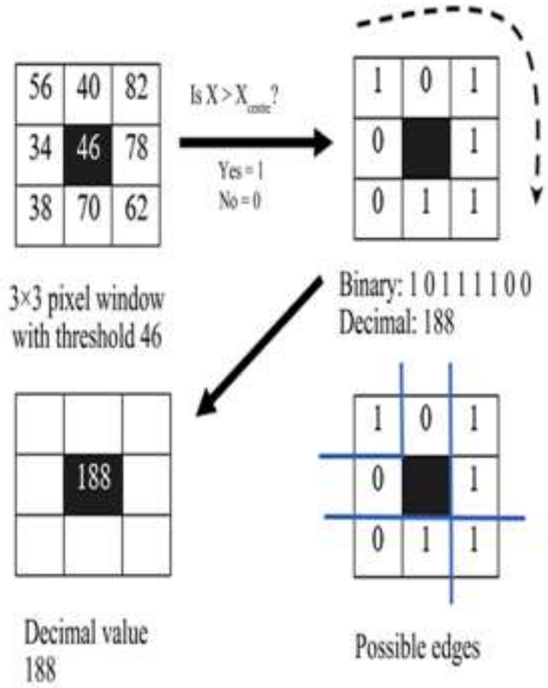


Figure 5. A 3 × 3 LBP operator

$$\mathrm{LBP}(x_c, y_c) = \sum_{p=0}^{P-1} S(g_c - g_p) 2^p \quad (1)$$

The threshold function $s(z)$ can be given by Equation (2)

$$s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \quad (2)$$

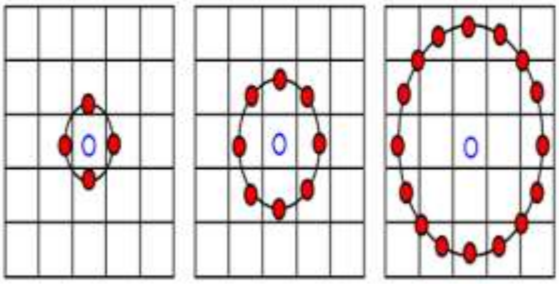
In the LBPH algorithm, the histogram which is used as a texture descriptor is basically a collection of the LBP codes of all the pixels for an input image, *i.e.*,

$$\mathrm{BPH}(i) = \sum_{x_c, y_c} \delta\{i, \mathrm{LBP}(x_c, y_c)\}, i = 0, 1, \cdots, 2^{P-1} \quad (3)$$

for which the Kronecker product function (.) is defined. We can make LBP operators with arbitrary radii and neighbourhood topologies using the LBPH method. Figure 6 shows the many configurations of the circular LBP operator, where P is the number of neighbouring pixels and R is the radius. The LBPH technique divides the whole grey face image into smaller parts, from which the LBP feature vectors are recovered. For each region, a histogram is created using the LBP feature vectors. By summing together the histograms of the smaller sections, we get the bigger histogram of the whole image, which represents the image's distinguishing characteristic.

Learning Face Recognition and Using It in the Classroom

To conduct this study, we have constructed our own database of 5 individuals to study. Our data sets are now divided in half. In Figure 7, we see an example of a database named LRD200, which has 1000 images altogether.



F figure 6. Circular neighbourhoods of the centre pixel with different neighbour pixels.

Figure 7. Sample face image of LRD200 database.

In a database going by the name LRD100, there are 100 pictures of faces for every individual, for a grand total of 500 pictures of people's faces. All of the facial photos from LRD100 are included in LRD200. To achieve better and more general identification, the photographs were taken under two lighting scenarios. We have developed an Android app called My New Cam to serve as a database for this project. Figure 8 and Figure 9 provide some pictures together with the progression information. Android Studio and the Java programming language were used to create the app. The SDK has been built to version 28 with camera API version 23. This software asks for the person's name when you take their photo in order to properly identify them. Later, during the facial recognition phase, this name will be utilised for verification. Subject photographs captured with the android app are sent to a designated folder on the system computer, where a directory watcher application is launched to identify any new face images. The directory watcher and face detector application will look for new photographs in the folder, identify any faces inside them, and then align the faces vertically such that the eyes are always facing front. The face photos are then pre-processed prior to cropping for improved facial recognition accuracy. To clean up the picture, we used a median filter. When the faces have been extracted, they are stored with the subjects' names in a "training-image" folder among all the other photographs in the database. As fresh face photos are added to the training-image folder, the training process will also begin automatically. The LBPH recognizer database.xml file stores the training data and is later used for recognition when the training procedure is complete. To train the picture database and repeat the face recognizer software once a fresh face image has been entered using the android app, the machine just has to be rebooted. We have implemented face recognition and detection using the OpenCV libraries in Python. Figure 10 illustrates the steps involved in developing a database and undergoing training.

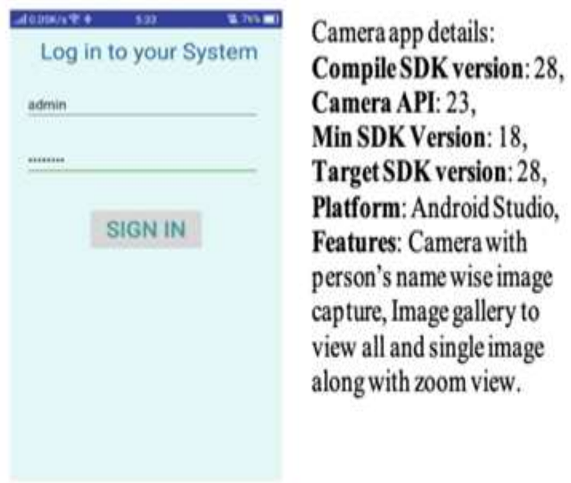
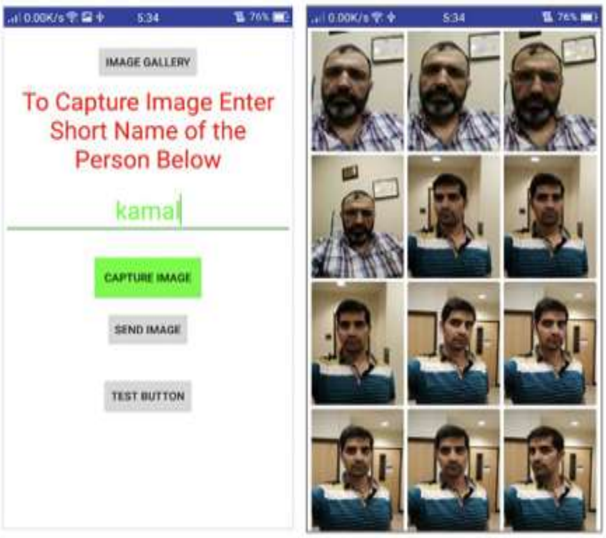


Figure 8. Login screen of the camera app (left) and camera features (right).



Face Recognition

We developed a webcam-based, real-time face recognition system. The input picture was obtained in real-time from the live video stream. The recognition process skipped certain frames. Recognizable frames are those captured after one hundredMs. The LBPH face database file and the NameList.txt file are both loaded before the camera is activated. CLAHE (Contrast Limited Adaptive Histogram Equalization) is used to improve the quality of each frame read, and then Gaussian filtering is used to remove the noise introduced by the camera during capture. A grayscale version of the original frame was created for facial recognition purposes. If the facial picture is skewed, it is corrected such that both eyes are level. When a face is spotted, it is scaled and cropped to remove unwanted parts. Using the LBPH technique, we can extract LBP feature vectors and generate a histogram of the face picture that captures the image's characteristics. The input picture is identified as the target image in the database by comparing its histogram to the database's.

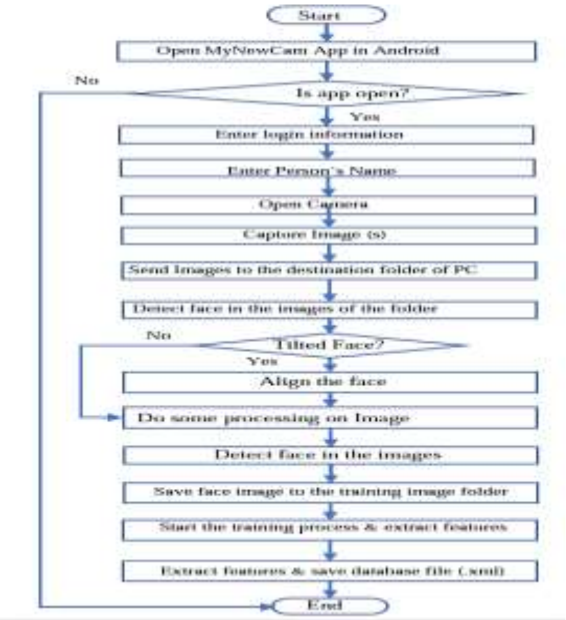


Figure 10. The process of creating a database and training.

having a same histogram distribution. Those who aren't in the picture database may be found by establishing a threshold value. We have used the Euclidean distance technique to evaluate histograms. In order to use the OpenCV libraries and Python for facial recognition, we need a list of names and IDs of the test participants. Each database picture in this study includes the subject's name, along with the image file name and an underscore (_) after the subject's name. Subject IDs were generated by reading all of the training photos, parsing out the names of the subjects shown in each image, listing only unique names, and sorting the results alphabetically. Following then, a sequential ID is assigned to each topic based on their position in the list. The list's names and topic IDs are then written to a text file with the generic name "NameList.txt." We use the ID of a subject name obtained during the identification procedure to look for that person in this text file. We've used pictures obtained with a laptop's camera as training data for a facial recognition system. Several low-resolution picture frames have been used in our tests of the recognition rate. There was a 100 Ms pause between each camera frame used in the identification procedure. Ten separate occasions employed 200 picture frames for identification. A recognition rate is then derived from those two thousand frames. The entire process of facial recognition is shown in this diagram.in Figure 11.

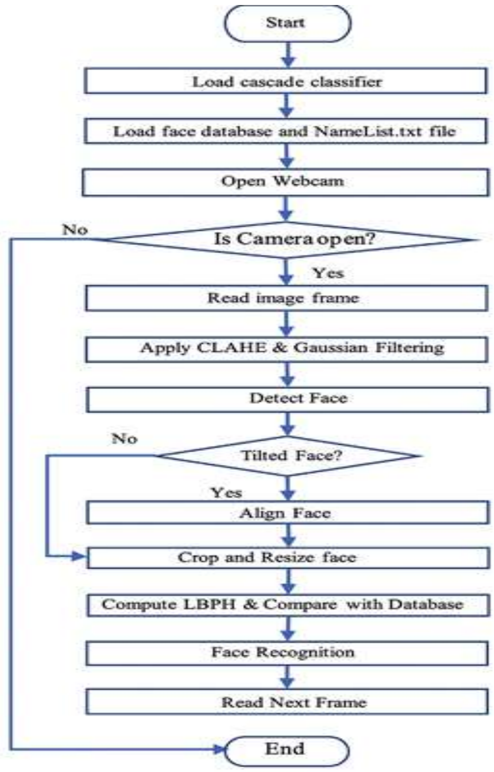


Figure 12 depicts the big picture of the proposed facial recognition system. The camera continually records footage and updates the recognition system with new photos, which are subsequently processed by the facial recognition module. On display, a name will appear next to a recognised face, while "unknown" will be used to denote a face that hasn't been identified.

Comment of the Experimental Findings

A core i5 Lenovo ThinkPad 14 with a 0.9 megapixel built-in camera has been utilised to successfully detect live human faces in our proposed face recognition system. The database was built using information gleaned from an Oppo R7kf Android phone running version 5.5.1. The camera on the rear is 13 megapixels. No flash was used in the photography. We tested real-time face recognition's efficacy using photos with resolutions of 15, 20, 30, 35, and 45 pixels in the experiment. Results are calculated using both the LDR200 and LDR100 datasets. For standard face recognition, where an individual's head is slowly rotated from the left 30 degrees to the right 30 degrees with an upward and downward face, a revolving head around the camera has been employed.

Face Detection at Various Low Resolutions

This section discusses the face recognition outcome for seVeral input photos with poor resolution. Table 1 shows the recognition rates using our own LRD200 database, whereas Table 2 shows the recognition rates using the LDR100 database when the subject's head is rotated around the camera. The recognition success rate grows as picture quality improves. There is a strong correlation between the amount of photographs in the database and the recognition accuracy. Table 3 demonstrates that using the LDR200 database, an identification accuracy of 78.84% is reached at only fifteen pixels, while this number rises to 98.05% at 45 pixels. The findings show that identification accuracy improves as the number of pixels in the input photos grows. Even while moving the head around, the identification accuracy remains good.

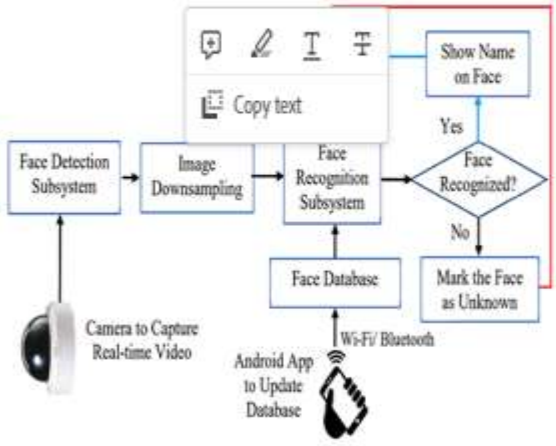


Figure 12. The proposed face recognition system.

Table 1. Recognition rate based on LRI200 database.

Recognition	Using Database LRD200		
	Correct Times	*Wrong Times*	*Recognition Rate*
At 15 pixel	1568	432	78.40%
At 20 pixel	1842	158	92.10%
At 30 pixel	1919	81	95.95%
At 35 pixel	1932	68	96.60%
At 45 pixel	1961	39	98.05%

Table 2. Recognition rate based on LRD100 database.

Recognition	Using Database LRD100		
	Correct Times	*Wrong Times*	*Recognition Rate*
At 15 pixel	1212	642	60.60%
At 20 pixel	1633	367	81.65%
At 30 pixel	1691	309	84.55%
At 35 pixel	1855	145	92.75%
At 45 pixel	1900	100	95.00%

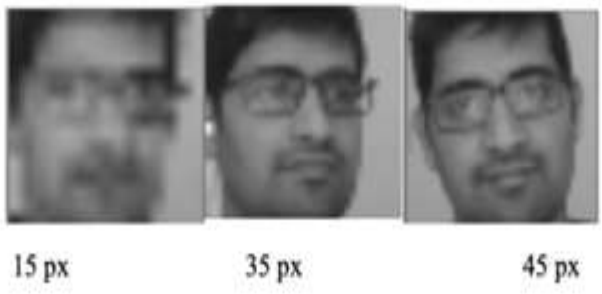
Table 3. Recognition rate based on LRD200 database.

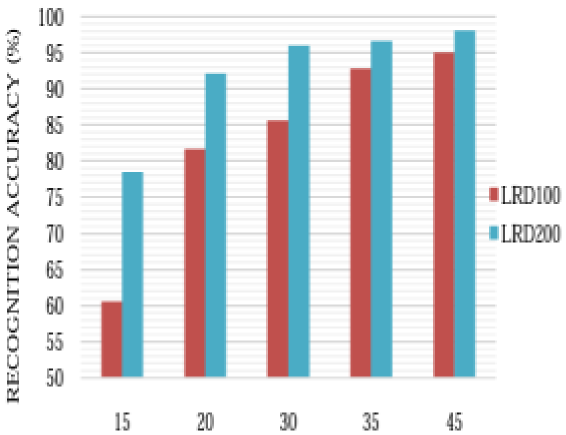
Recognition at 45 px	Using Database LRD200		
	Correct Times	*Wrong Times*	*Recognition Rate*
Front Facing	1993	7	99.65%
Facing 30° Right	1637	383	81.85%
Facing 30° Left	1545	455	72.25%

camera. This is because the identification accuracy is enhanced by aligning the cropped facial picture before recognition, even when the head is inclined with regard to the camera. In contrast, CLAHE pre-processes the low-resolution photos to boost the performance of face recognition. CLAHE made it so that there was less of a contrast between bright and dark areas of the photos. The input facial picture is shown at different resolutions in Figure 13 to illustrate the recognising process. The rate of face identification for several low-resolution photos is graphically shown in Figure 14. It also shows how recognition accuracy has changed over time as measured by the average number of photos associated with a given individual in the database.

Pose-Independent Facial Recognizance

Accuracy of recognition varies when head is tilted in relation to camera. The identification rate drops down precipitously as the deflection angle increases. A total of three different angles were used to record the recognition rates:





RESOLUTION IN PIXEL

LRD100 LRD200 Graphical representation of recognition accuracy with image resolution (pixel).

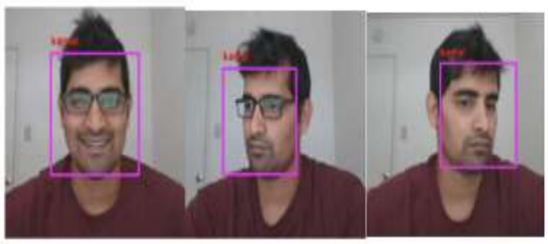


Figure 15. Face recognition with different angular deflection.

Positions of the head and face in the front, 30 degrees to the right and 30 degrees to the left, as indicated in Figure 15. The head was moved vertically while taking pictures of the subject's face from the front. The face deflections were around 30 degrees left and right, and the head was moved up and down in the same manner. When just the front of a person's face is seen, the recognition rate is greater. The identification accuracy from 2000 frames was calculated by using 200 frames at a time and repeating the procedure 10 times.

Comparison of Our Results with Other Methods

In this subsection, we show how our suggested algorithm stacks up against similar existing approaches. Table 4 provides a comprehensive comparison of our suggested facial recognition system. Although the referenced paper [16] demonstrated 90% accuracy for face recognition at 35 pox, our system is capable of real-time identification.

Particulars	Zhao and Wei [15]	Ahmed et al [16]	Ours
Method	MLBPH	LBPH	LBPH + CLAHE + alignment
Lowest image resolution	-	35px	15px
Accuracy at 15 px	-	-	78.4%
Accuracy at 35 px	-	90%	96.60%
Accuracy at 45 px	-	94%	98.05%
Accuracy with 30° angular deflection	48.4% - 55%	-	72.25% - 81.85% @ 45 px
Use of Android app for database	-	-	Yes
Auto-update of the database & auto-restart of process	No	No	Yes
Number of image per person in the database	7	500	200

Photos of people's faces with a sensitivity of 96.6 percent at a resolution of 35 pixels. In addition, our suggested technique has a recognition accuracy of 78% for face photos as small as fifteenpox. Comparing our database to reference [16], we find that we have much less photos of each individual. By using facial deflection, our suggested system outperforms reference [15] in terms of recognition accuracy.

Conclusions

The face may be recognised in a variety of lighting situations and at low resolution with the help of a strong algorithm called Local Binary Pattern Histogram (LBPH) architecture of face identification. Median filtering was utilised for photographs in the database, whereas Gaussian filtering was employed for facial recognition. While using the LRD200 database, we found that our innovative face recognition method achieved an accuracy of 78.40% at a low resolution of fifteenpox, and 98.05% at a high resolution of 45 pox. In addition, at 15 pox and 45 pox, respectively, we have reached 60.60% and 95% using LRD100 data-based. There was a considerable increase in identification precision after adjusting for attitude. To further streamline the process of retraining and resuming the identification process, an android app has been built for capturing the subject picture and delivering photographs from the android app to the pc. This study also discovered that the identification accuracy at low resolution is significantly influenced by the amount of photographs per individual in the training set database. The accuracy of face recognition may improve with a bigger dataset that includes a wider range of poses and lighting conditions [18]. The suggested approach will aid law enforcement in identifying potential criminals in busy public spaces like airports, train stations, and bus terminals. The photographs in the research database cannot be deleted using the android app created for this study; only new images may be added. Further work has to be done on the app so that it may be used to manage the whole face database from an Android smartphone.

References

[1] Kaur, G., Kaur, H. and Kaur, M. (2017) A Survey on Face Recognition Techniques. International Journal of Advanced Research in Computer Science,

[2] Ferial, R., Wibi Rama, S. and Setiawan, N.A. (2017) Gender Recognition Using PCA and LDA with Improving Pre-processing and Classification Technique. 2017 7th International Annual Engineering Seminar (Ines), Yogyakarta, 1-2 August 2017, 1-6. https://doi.org/10.1109/INAES.2017.8068547

[3] Parmar, D.N. and Mehta, B.B. (2014) Face Recognition Methods & Applications. arXiv:14030485.

[4] Shije, S.K. (2013) Biometric Prediction on Face Images Using Eigenface Approach. 2013 IEEE Conference on Information Communication Technologies, Tuckaway, 11-12 April 2013, 104-109. https://doi.org/10.1109/CICT.2013.6558071 8, 33-35.

[5] Morisot, N., Amiel, F., Hamed, I.D. and Ea., T. (2007) A Comparative Implementation of PCA Face Recognition Algorithm. 2007 14th IEEE International Conference on Electronics, Circuits and Systems, Marrakech, 11-14 December 2007, 865-868. https://doi.org/10.1109/ICECS.2007.4511128

[6] Lu, J., Platanista's, K.N. and Venetsanopoulos, A.N. (2003) Face Recognition Using LDA-Based Algorithms. IEEE Transactions on Neural Networks and Learning Systems, 14, 195-200. https://doi.org/10.1109/TNN.2002.806647

[7] Viola, P. and Jones, M. (2001) Rapid Object Detection Using a Boosted Cascade of Simple Features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, 8-14 December 2001, I-511-I-518. https://doi.org/10.1109/CVPR.2001.990517

[8] Chakras Ali, S.V. and Kitale, S. (2016) Optimized Face Detection on FPGA. 2016 International Conference on Circuits, Controls, Communications and Computing (I4C), Bangalore, 4-6 October 2016, 1-6. https://doi.org/10.1109/CIMCA.2016.8053269

[9] Mohan, A.R., Sudha, N. and Meher, P.K. (2008) An Embedded Face Recognition System on A VLSI Array Architecture and Its FPGA Implementation. 2008 34th Annual Conference of IEEE Industrial Electronics, Orlando, 10-13 November 2008, 2432-2437. https://doi.org/10.1109/IECON.2008.4758338

[10] Jiamusi, A.Y., Ghaibi, S.F. and Masmoudi, D.S. (2009) Implementation of Face Recognition System in Vertex II Pro Platform. 2009 3rd International Conference on Signals, Circuits, and Systems (SCS), Mede nine, 6-8 November 2009, 1-6. https://doi.org/10.1109/ICSCS.2009.5412313

[11] Endure, R., Katha it, M., and Ray, K.C. (2013) Face Recognition Using PCA on FPGA Based Embedded Platform. 2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE), Jabalpur, 16-18 December 2013, 1-4. https://doi.org/10.1109/CARE.2013.6733778

[12] Schaffer, L., Kincses, Z. and Pletzl, S. (2017) FPGA-Based Low-Cost Real-Time Face Recognition. 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, 14-16 September 2017, 000035-000038. https://doi.org/10.1109/SISY.2017.8080568

[13] Borkar, N. and Kudelka, S. (2017) Face Recognition System Using PCA, LDA & Jacobi Method. European Journal of Advances in Engineering and Technology, DOI: 10.4236/opj.2021.114005 4, 77 Optics and Photonics Journal. C. Paul, S. Aslan 236-331.

[14] Irgens, P., Bader, C., El, T., Saxena, D. and Ababei, C. (2017) An Efficient and Cost-Effective FPGA Based Implementation of the Viola-Jones Face Detection Algorithm. Hardware, 1, 68-75. https://doi.org/10.1016/j.ohx.2017.03.002

[15] Zhao, X. and Wei, C. (2017) A Real-Time Face Recognition System Based on the Improved LBPH Algorithm. 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP), Singapore, 4-6 August 2017, 72-76.