International Journal of Emerging Trends in Computer Science and Information Technology
ISSN: 3050-9246 | https://doi.org/10.63282/3050-9246.IJETCSIT-V6I4P116
Eureka Vision Publication | Volume 6, Issue 4, 116-120, 2025

Original Article

Implementation of Face Recognition on FPGA Using Haar Features and LBPH Algorithm

Darko Pajkovski[1], Kostandina Veljanovska[2], Zoran Kotevski[3], Nikola Rendevski[4]
[1,2,3,4]Faculty of Information and Communication Technologies, "St. Kliment Ohridski University" – Bitola, Republic of North Macedonia.

Received On: 15/10/2025 Revised On: 07/11/2025 Accepted On: 15/11/2025 Published On: 22/11/2025

Abstract - Face recognition technology has wide usage in applications such as surveillance, biometrics, and security. In this paper, we introduce a complete real-time face recognition system that is implemented on an FPGA and consists of face detection and recognition modules. The face recognition system is divided in two modules: training module and evaluating module. Then, the Local Binary Patterns Histogram (LBPH) extracts facial features from the detected faces in the live stream. Recognition is then performed by applying a Euclidean distance classifier to recognize and match the faces.

Keywords – FPGA, face recognition, LBPH .

1. Introduction

Facial recognition is a method used to verify or authenticate the identity of a person based on the analysis of his or her facial features. Recognition software has the ability to identify individuals in real-time situations or using still photos or video footage. As a subset of identity verification, it works by identifying and analyzing unique facial features in different images. These processes can be used to detect faces in photos or videos, determine if two different images are of the same person, or to find a face in a large database of stored images. A common use of facial recognition is in the areas of surveillance and security since it increases accuracy and speed while at the same time decreasing the chances of human error. Ever since the incidents of 9/11, more focus has been given to creating sophisticated security systems for maintaining public safety, especially at important sites like airports, corporate offices, and border control points where correct identification and verification are absolutely necessary. Facial recognition systems can help minimize security threats and serve as a very important element in the prevention of future attacks [1].

CCTV cameras that are equipped with facial recognition systems play a vital role in tracking and recognizing specific individuals. Moreover, such systems are useful tools in missing people searches, but their success relies on strong recognition algorithms and an extensive facial database. Last but not least, facial recognition has become ubiquitous on social media websites, such as Facebook or Instagram where the system provides recommendations for tagging friends identified in photos. Indeed, facial recognition systems display a multifaceted set of applications in multiple fields. In this paper, face recognition is carried out by using the Haar Cascade algorithm for detecting faces, the LBPH algorithm for extracting features, and the Euclidean distance classifier for identification. The overall process includes a number of imperative steps: the generation of a dataset, face capturing, feature extraction, and classification thereafter. All the experimental tasks and implementations were done in Python using the OpenCV library.

2. Related Works

The authors in [2] presented a face recognition system that is designed to work on exterior based approaches. They used the Viola-Jones algorithm to detect and extract face images to be stored in the database and the Square Euclidean Distance measure to compare the similarity between two images by measuring their distance.

The authors of [3] present hybrid ARM and FPGA-based face detection system powered by the OpenCV computer vison library and Cyclone V system-on-chip. The developed system was compared to CISC-based setup with Intel Core i72670QM CPU. Results presented showed that SoC has 43% less time that Intel Cored i7 setup in detecting a face from standard input file.

The authors [4] described the design and implementation of a real-time facial recognition system on FPGA technology, which can operate at a speed of 45 frames per second. The system has three main modules: face detection, downsampling, and facial recognition. Each module was designed and implemented on a Virtex-5 FPGA. Furthermore, the paper demonstrates the architectural synthesis of the detection and recognition modules into an overall system on actual hardware.

This paper [6] focuses on face recognition, aiming to improve recognition accuracy for practical use in crowded areas such as airports, railway stations, universities, and malls for security purposes. Face detection is applied to locate and size human faces in images, focusing only on facial features using Haar Cascades and Local Binary Patterns.

This paper [7] presents an efficient reconfigurable architecture for face recognition using Local Binary Pattern (LBP) features. A Gaussian filter is applied for preprocessing, followed by an optimized LBP and histogram-based feature extraction using a low-complexity moving window design. Recognition is achieved by comparing histogram features of database and test samples. Experiments on the ORL dataset are done in Matlab and Spartan-6 Fpga board.

This work [8] presents an automated facial recognition-based attendance system employing the Local Binary Pattern Histogram (LBPH) algorithm for robust identity verification in multi-face scenarios. The proposed solution achieves 98% recognition accuracy, supports simultaneous detection of multiple individuals, identifies unauthorized persons, and automatically logs attendance details (name, date, and time) into an Excel file, offering a secure and efficient alternative to traditional manual attendance methods.

The study [9] examines various face detection and recognition approaches, focusing on the use of HaarCascade for face detection and Local Binary Pattern Histogram (LBPH) for feature extraction to enable reliable identification from images and live video feeds. The work emphasizes the importance of machine-learning-based facial analysis for modern AI systems, particularly in security and real-time recognition applications.

3. Methodology

Face recognition system contains tree main phases for face detection, feature extraction and face recognition.

1. Face Detection: Face detection and localization from camera feed with preprocessing of human faces separating from objects present in an image.
2. Feature Extraction: From previous face detection we are extracting the features through Local Binary Pattern Histogram.
3. Face Recognition: The features thus extracted are then fed into a classifier, which classifies or labels the input based on a machine learning algorithm. The classifier matches the test image with the images stored in the database, and the task may be accomplished through a supervised machine learning process.

Whereas the human mind recognizes faces naturally, in computer vision it takes a number of steps: data collection, analysis, and then training a model to recognize various facial features. Older approaches such as Eigenfaces and Fisherfaces are susceptible to changes in lighting, which can compromise accuracy. Because of this, LBPH was selected, since it is more consistent in everyday environments where lighting is not always controllable. The (LBPH) face recognizer was developed to overcome this limitation by focusing on the local features of an image rather than its overall arrangement. In this method, every pixel is compared to its neighboring pixels in order to acquire the local texture information. These comparisons enable the creation of Local Binary Patterns (LBP), which are then converted from binary forms to decimal equivalents. A histogram is then developed from these decimal values to represent the image. When in training mode, a histogram is developed for each facial image in the dataset, and the histograms are stored with the identity of the respective person. When a new image is encountered, the system develops its histogram and compares it with the recorded histograms. The closest matching histogram is chosen, and the label of the matched person is given as the recognition result [5].

There are several methods that can be used to compare histograms, such as the use of absolute differences, Euclidean distance, among others, to measure the distance between pairs of histograms. In the current study, the widely used Euclidean distance approach is used, which is calculated based on the following formula:

$$D = \sqrt{(\Sigma(H1 - H2)^2)} \tag{1}$$

Following are described algorithms that are used in this research:

The Haar Cascade Algorithm is we are using in face detection to identify objects in images or video based on their visual features. This method relies on a cascade function that is trained with a large dataset containing both positive and negative images. Once trained, the function can detect faces in new images. A Haar feature is created by calculating the difference between the sums of pixel values in adjacent rectangular regions within a detection window. These differences are then used to classify sections of the image. Therefore, a Haar feature made up of two rectangles positioned over the eye and cheek area can capture the contrast between the darker eye region and the lighter cheek area. These rectangles are placed relative to the position of the face within the video frame, forming the bounding box around the detected face. To make the computation faster, the algorithm uses an integral image, which stores cumulative pixel intensities along both vertical and horizontal axes. With this approach, Haar-like features can be computed efficiently using only a few reference points from the rectangle's corners. As a result, the calculation of adjacent rectangular areas requires just six references, making the algorithm both effective and computationally efficient.

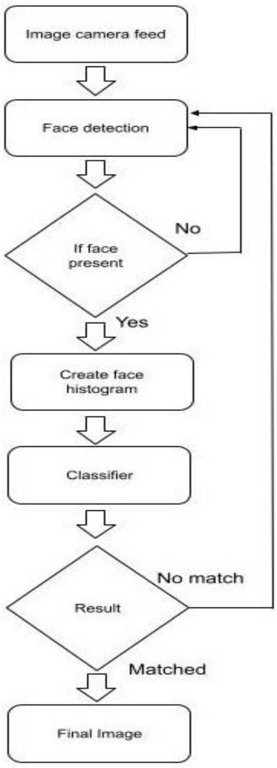

Figure 1. Flow chart for purposed system

The AdaBoost algorithm functions by creating a model that places a weight on all the data points and gives higher weight to the points that are classified incorrectly. The focus on these more complex cases by increasing their weight occurs in the next iteration and continues until all the error is minimized. For facial recognition, the algorithm analyzes the input picture by using a ‘sliding window’ and calculates Haar features for each region. The computed Haar features are then assessed against a predetermined threshold in order to separate facial and non-facial areas. Because a single Haar feature is insufficient for reliable face recognition, multiple Haar features are used. AdaBoost arranges these features into a cascade of classifiers for face detection which optimizes face recognition.

Cascading Classifiers: The Cascading technique is a unique facet of an ensemble strategy which is formed by multiple weak classifiers and uses the output of one weak classifier as supplementary information for the next classifier in the cascade. If the boosted strategy is used to increase classifier accuracy for each classification, the classifier combines the scores from the weak learners. Each classifier's step whilst the sliding window moves, will mark the area the classifier is currently positioning with a positive or a negative label. Positive results indicate the presence of a face while negative results indicate the absence of a face. If the label is negative, the region is correctly classified and the detector proceeds to the next window. The classifier moves on to the next stage of the region classifier if a positive label is assigned. When the last stage classifies the region as positive, the face is detected at the current window position.

4. Proposed System Architecture

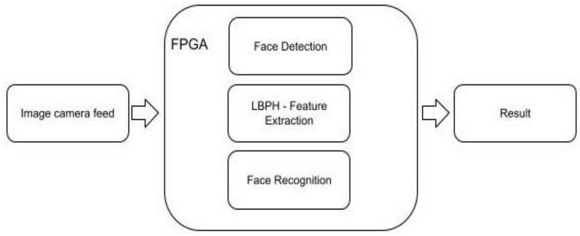

Figure 2. Proposed system architecture

As demonstrated in Fig.2, the algorithm put forth conducts an analysis of sub-windows within the incoming camera feed to identify facial features. The operation of the model is developed into two principal stages: enrollment and testing. Throughout both stages, the procedure encompasses face acquisition, preprocessing, and the extraction of features. Upon the completion of these stages, a classifier is employed to ascertain whether the test image is congruent with any images preserved within the dataset. In conventional image processing practices, the input image is often resized to various dimensions, following which a fixed-size detector is implemented across these scaled images to guarantee precise face detection.

The first step of face detection algorithm is Haar features selection. Therefore, all human faces share some similar properties. This kind of features are used to detect difference in the black and white parts of the image. Properties that are common to human faces are:

- The area around eyes usually appears darker compared to the upper cheek region.
- The bridge of the nose is generally brighter than the eye region.

The second step of the face detection algorithm is to convert an input camera feed image into an integral image. In an integral image, every pixel is the summation of the pixels above and to the left of it shown on Fig.3.

1	5	2
2	4	1
2	1	1

1	6	8
3	12	15
5	15	19

Figure 3. Input image (left) and integral image (right)

4. Results and Discussions

In dataset development, every participant is also assigned a unique identification number alongside their name. In the recognition process, if the test subject happens to be present within the dataset, the classifier recognizes the person by showing the name along with the recognition rate. If the person does not happen to be present within the dataset, the classifier assigns the person an unknown status, as shown by Figures 4 and 5 respectively.

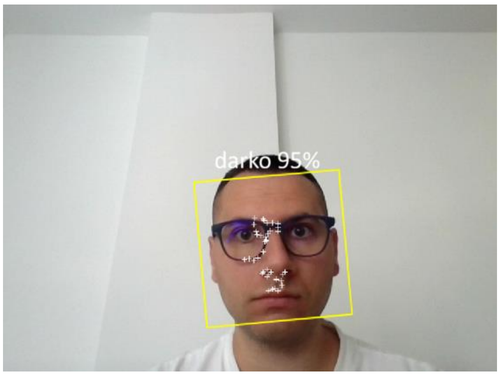

Figure 4. Recognizing face with accuracy percentage

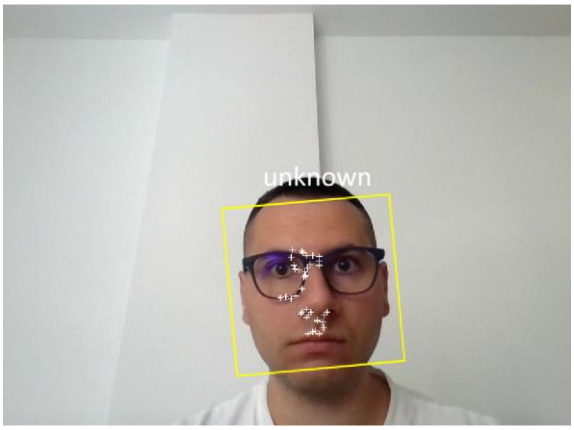

Figure 5. Not recognized person

The proposed system contains a Graphical User Interface (GUI) that includes an upload button for loading an image to test the accuracy of recognition and an open camera live button, which opens up the camera for live testing of the accuracy. When the user uploads an image that is exactly similar to one present in the dataset, the system provides 100% accuracy in recognition. During testing on live cameras, this may be variable due to factors such as the number of training images for that person, angle of the face, with/without glasses, and illumination. The system consistently identifies the correct person ID (for example, Person_43 from the database). However, the confidence percentage may fluctuate due to real-time variations in the live camera feed. All experiments were performed by manually adding images to the database one by one and retraining the model after each update. When around ten images per person are included in the dataset, the recognition process becomes significantly more stable, particularly in the live camera mode. The experimental results in this test scenario indicates that person_43 from databse was successfully recognized in all 10 evaluation cases with the associated confidence scores presented in Table 1.

Table 1. Real-time confidence score

Number of pictures	Real-time confindence score obtained from live webcam expressed as a percentage
1	20
2	22
3	24
4	21
5	22
6	21
7	25
8	24
9	23
10	26

The decrease in recognition accuracy is because the wearing or removal of glasses leads to significant changes in the visual appearance of the area around the eyes. Glasses add new shapes, reflections, and shadows, which impact the local texture patterns on which the algorithm relies. This means that the features extracted no longer match those in the images stored, thus making identification rather challenging for the algorithm.

The second test was conducted using the Oracle Research Laboratory (ORL) face database, which contains 400 images in total with 70x80 resolution of each image. From this dataset, 80% of the images were used for training, while the remaining 20% were reserved for testing. Complete face detection system has around 100 Adaboost iterations in total distributed across roughly 10-15 stages of the cascade classifier with accuracy of 80%.

The third test was conducted on Labeled Faces in the Wild database from University of Massachusetts containing 13233 images and 5749 people with 250x250 resolution of each image. Each iteration, Adaboost adds one weak classifier to influence on accuracy improving. Therefore, with around 200 iterations, accuracy is 70%. Table 2 shows number of iterations and accuracy.

Table 2. Database accuracy comparison

	Algorithm Iterations	Accuracy %	Comments
ORL DB	50	75	strong balance (speed and accuracy)
ORL DB	100	80	borderline speed, target accuracy
ORL DB	150	82	near saturation
LFW DB	50	62	Fast, real-time preview
LFW DB	100	66	moderate speed, average accuracy

LFW DB	200	70	borderline speed, target accuracy
LFW DB	500	76	Too slow, near saturation

From the table above, we can observe that the ORL database achieves higher accuracy with significantly fewer AdaBoost iterations compared to the LFW database. In other words, because the ORL database contains smaller and less complex images, the classifies can learn effective decision boundaries much faster, Also, LFW dataset contains more variation in lighting, facial orientation and requires more iterations to reach accuracy level.

Table 3. Comparison of platforms with time and efficiency

Platform	Execution time	Energy efficiency
AMD Ryzen 7	10-20 ms	70-100 W
De1-Soc with Fpga	60-100 ms	5-10 W

From the above table, it can be concluded that the AMD Ryzen performs significantly faster than the FPGA in this scenario, however, it also consumes considerably more energy. On the other hand, the FPGA platform can be incorporated into surveillance security systems to follow selected persons in real-time since it is energy-efficient as well as transportable. This practice is particularly effective where it is impracticable for security officers to monitor surveillance feeds constantly. With the computer vision technologies, such systems can offer abilities equivalent to those of AI, therefore increasing the both effectiveness and credibility of security operations.

4. Conclusion and Further Work

Within this project, face detection, and classification through the use of the Euclidean distance classifier while feature extraction through Local Binary Patterns Histograms (LBPH) is done sequentially. The face detection and classification system is developed and implemented using the OpenCV framework. Analysis stemming from the current study points out the main attributes that future studies may focus on. These include ensuring system consistency no matter the prevailing conditions. These would include having different illumination, wearing glasses, having facial hair, or dealing with complex situations like distinguishing between identical twins. Such situations would warrant using a minimum distance classifier.

There are additional attributes that future studies can focus on. These include modifying the proposed approach to study the development and evolution of genetically inherited facial expression traits. Such studies would increase the accuracy of recognition. There is also a need to refine the system to include additional security features. Such advanced measures would be necessary for sensitive areas like the protection of government confidential databases and in the identification of criminals.

References

[1] Abbas, Eyad I., Mohammed E. Safi, and Khalida S. Rijab. "Face recognition rate using different classifier methods based on PCA." 2017 International Conference on Current Research in Computer Science and Information Technology (ICCIT). IEEE, 2017.

[2] Barnouti, Nawaf Hazim, et al. "Face detection and recognition using Viola-Jones with PCA-LDA and square Euclidean distance." International Journal of Advanced Computer Science and Applications 7.5 (2016).

[3] Regalado, Gil Michael E., et al. "Integration of OpenCV and Cyclone V Hybrid ARM and FPGA SoC for Face Detection Application." 2023 22nd International Symposium on Communications and Information Technologies (ISCIT). IEEE, 2023.

[4] Matai, Janarbek, Ali Irturk, and Ryan Kastner. "Design and implementation of an fpga-based real-time face recognition system." 2011 IEEE 19th Annual International Symposium on Field-Programmable Custom Computing Machines. IEEE, 2011.

[5] Stekas, Nikolaos, and Dirk Van Den Heuvel. "Face recognition using local binary patterns histograms (LBPH) on an FPGA-based system on chip (SoC)." 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE, 2016.

[6] Chilap, Priyanka, et al. "Haar Cascade Algorithm And Local Binary Pattern Histogram LBPH Algorithm In Face Recognition." International Journal of Research Publication and Reviews 3.4 (2022): 2395-2398

[7] Bhavikatti, Sumangala, and Satish Bhairannawar. "E cient Reconfigurable Architecture to Extract Image Features using Local Binary Pattern for Face recognition." (2023).

[8] Rao, Manjula Gururaj, et al. "Innovative Solutions: LBPH Technology Driving the Future of Intelligent Security Systems." 2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE). IEEE, 2024.

[9] Gupta, Mridul, et al. "HaarCascade and LBPH algorithms in face recognition analysis." 2023 World Conference on Communication & Computing (WCONF). IEEE, 2023.