2022 International Conference on Futuristic Technologies (INCOFT)
Karnataka, India. Nov 25-27, 2022

Realization of Smart and Highly Efficient IoT-based Surveillance System using Facial Recognition on FPGA

Abhay Pratap Singh Shekhawat
Dept. of ESE, National Institute of Electronics and Information Technology.
Aurangabad, Maharashtra, India.
abhayshekhawet@gmail.com

Abhishek Chaurasiya
Dept. of ESE, National Institute of Electronics and Information Technology.
Aurangabad, Maharashtra, India.
Abhichaurasiya19@gmail.com

Pawan Chaurasiya
Dept. of ESE, National Institute of Electronics and Information Technology.
Aurangabad, Maharashtra, India.
chaurasiyap815@gmail.com

Pawan Kumar Patel
Sr. Technical Assistant,
Dept. of ESE, National Institute of Electronics and Information Technology
Aurangabad, Maharashtra, India.
pawankumar@nielit.gov.in

Prashant Pal
Scientist-'B'
Dept. of ESE, National Institute of Electronics and Information Technology
Aurangabad, Maharashtra, India
prashantpal@nielit.gov.in

Shashank Kumar Singh
Scientist-'B'
Dept. of ESE, National Institute of Electronics and Information Technology
Aurangabad, Maharashtra, India.
shashank@nielit.gov.in

2022 International Conference on Futuristic Technologies (INCOFT) | 978-1-6654-5046-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/INCOFT55651.2022.10094500

***Abstract*— In this paper, we have developed a security system using FPGA with the integration of IoT using face recognition technology with the help of the OpenCL library. We will be using HLS for the conversion of our C++ code so that it can be deployed on FPGA after converting it into VHDL. The designed system can recognize and allow the authenticated person to enter the building or office even if the person somehow bypass the authentication then also our system can catch the person using real-time face detection using all the cameras integrated with our IoT system and comparing the faces stored in the database and will alert the security personals, even out system will be able to find one particular person in building conditioned if the person is in camera's view which will make our system even more effective for surveillance and finding someone.**

Keywords—VLSI, openCL, FPGA, IoT

I. INTRODUCTION

Facial recognition is a means of authenticating or validating a person's identity through their face. Software for facial recognition can recognize people in real-time as well as in still photographs and movies. Face recognition is a subtype of identity verification. It works by detecting and measuring facial features in photos. Techniques for facial recognition may identify human faces in photographs or videos, determine whether a face appears in two distinct shots taken by the same individual, or search for a face in a large database of previously captured pictures. Monitoring and security applications are among the most popular application of real-time monitoring and face recognition technologies because boost productivity and dependability while reducing human error.

We can improve the accuracy and reduce the latency of our surveillance system by using FPGA [1]. As we can see that there will be a lot of cameras in our system so they all need to work together which requires creating a number of different instances for each camera on our processor for further face recognition and then generating output which will be the alert system or as required by the condition so to overcome that problem we will be taking help of parallel processing power of FPGA. This means that we can create as many instances as we want for our system so that they can work simultaneously to give a better result.

The IoT on Ai-powered system connecting computer vision to the IoT ends up creating strong security; it helps by providing real-world environments with speedier, better support with much less efficient time, as well as all systems are linked to the internet with necessary technological systems that support more effective and precise operations. Scalable frameworks are needed for IoT systems. IoT systems can readily communicate with the external world with low power, reduced latency, and excellent determinism by the usage of a Field Programmable Gate Array (FPGA).

II. LITERATURE REVIEW

Development and implementation of such an IoT-based real-time security monitoring system

An affordable IoT-based monitoring and security system is presented in this paper. The suggested system has applications for home security and other forms of management. Rpi, Wifi camera deployment aids in the detection, reporting, and user surveillance of incursion incidences by users Furthermore, the system alerts users and notifies the region, reducing the damage caused by burglaries. The system's use of a cloud architecture allows for the storage of acquired photographs and videos. [2]

With computer vision to detect people in real-time surveillance using IoT

The suggested system in this paper employs a prediction that, if matched, is closer to the predicted value than the real-time prediction value. With the appropriate geolocation, the system alerts the police. The tentative outcome of real-time face identification involves detecting faces in live human face streams and labeling them with your name if you have been verified; otherwise, if you are not identified inside the database, real-time face matching labels the individual as "UNKNOWN." When security officials are unable to watch 24 hours of Surveillance tapes, real-time

face recognition systems are used in video surveillance to track a single individual in real-time. Computer vision is capable of performing tasks similar to AI. IoT AI thinks more quickly and intelligently. In comparison to other systems, machine learning is simpler to use and cannot make things difficult.[3]

Remote DeepFace Models Face Recognizer Creation and Implementation Using the sbRIO FPGA Technology and the NB-IoT Module

In LabVIEW, they created a remote DeepFace model system for face recognition using the camera module., they integrate sbRIO FPGA with NB-IOT module in this work. The test findings are quite promising. Based on the NB-IOT module, we create a distant network infrastructure and local application server, and we test their suitability for data transfer. They completed the DeepFace model's FPGA implementation, which has application potential. However, the 3D alignment is not complete, and the advantages of the original approach model aren't realized. completely used, because of the constraints of calculating functions on RT. There is still an opportunity for improvement.[4]

OpenCL based FPGA Accelerator Development and implementation for YOLOv2

The development and implementation of a YOLOv2 FPGA accelerator built on OpenCL are presented in this work. To improve the model's detection speed, we processed the convolutional layer using the Winograd technique. The convolution procedure is sped up by highly parallel processing. Efficacy has significantly improved when compared to prior relevant investigations.[5]

III. Methodology

Our System will have two features one feature will be alerted if an unauthorized person enters the premises and the other feature will be finding a person in the building or we can say tracking a particular person in the building or an area wherever our system is implemented firstly all the camera will be taking giving us live video which will be then fed to our FPGA using Wi-Fi modules. FPGA is specifically programmed for the task of face recognition which will then transfer its result as an output for our IoT module which will be Arduino or Rpi in our case then RPi will generate the alert message required for the given condition for example if any unauthorized person was spotted in the building then it will send a message containing warning and Facial ID with the location of that person so that security persons can reach there and identify the intruder quickly and if we have selected to find a person and when the person if found then it will send a message containing the location of the spotted person to the concerned authority. The entire procedure is implemented as IOT using a linked FPGA, Raspberry Pi 3, and camera.

On Vivado HLS, we natively synthesized a C++ model of our code to produce HDL files for FPGA implementation [6]. Our system was created utilizing the LBPH OpenCV recognizer [7]. The human mind finds it extremely easy to recognize faces, but in the computer vision pipeline, we must first collect the data, then analyze it, and then the last train and educate the model to recognize different facial features. Since Eigenfaces & Fisherfaces, these two recognizers, are also influenced by light, we chose LBPH since we cannot always guarantee ideal lighting in real life. The LBPH face recognizer is a development that addresses this flaw. The idea is to focus on an image's local elements rather than its overall composition. The LBPH method compares each pixel with its surrounding pixels to determine the local structure of a picture. After constructing a list of LBP, we convert each binary image to a decimal value before generating a histogram that includes all of the values. The training data collection will ultimately contain one histogram for each face image. They are then kept for further identification. For the reason that the algorithm maintains track of which histogram corresponds to which individual. When we input a new picture to a recognizer for identification, the recognition system will produce a histogram with that new image, compare it to the histograms that it already has, select the best match histogram, and provide the individual label associated with that best fit histogram.[8]

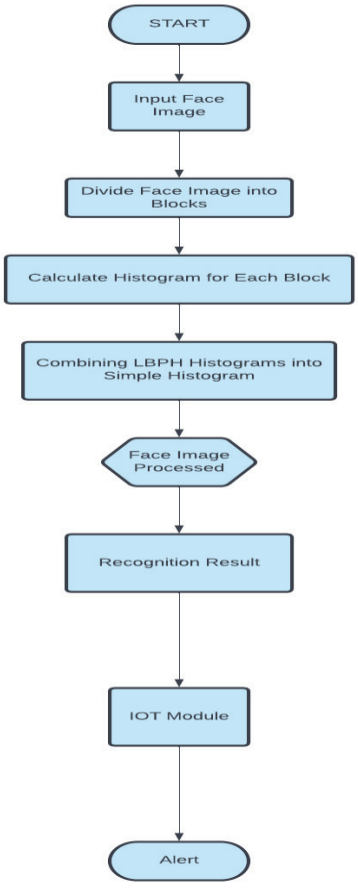

Fig. 1.

We may use a number of approaches to evaluate histograms, including such as absolute value, euclidean distance, and so on, to compute the distance between the two histograms. In this case, we may utilize the well-known Euclidean distance using the following formula:

$$D=\sqrt{(\sum(hist1-hist2)^2)}$$

Algorithms used to extract features, train datasets, and to recognize faces are as follows:

- Haar Cascade Algorithm: In order to identify different objects in a picture and video based on their features, face detection uses the Haar cascade. A cascade function is learned in this approach by utilizing a large number of both positive and negative pictures. The next step is to utilize it to find faces, which are the focus of other photographs. A Haar feature is formed by calculating the difference here between the sum of the adjacent pixels in successive rectangular parts at a defined location in a detection window. This differentiation is then used to categorize image subdivisions. A Haar feature with a pair of adjacent rectangles over the eye and cheek area, for example, may recognize the difference in the color of both the cheeks and eye in a facial image. These rectangles are placed in relation to the location of a video frame, which acts as the targeted face's bounding box. An integral image is a picture generated by accumulating the brightness of subsequent pixels in both the vertical and horizontal axes. By using the values of the point distinguishing the white and black areas and the corner points of the rectangle, integral images allow for the quick computation of Haar-like characteristics. Calculating the area of two neighboring rectangles only requires six location references.
- AdaBoost Algorithm: AdaBoost is a Machine Learning approach that is utilized as an Ensemble Method. The most frequent AdaBoost method is decision trees including one level, which is decision trees with just one break. These trees are often referred to as Decision Stumps. This approach constructs a model and assigns a weighting factor to all data points. It then applies larger weights to incorrectly categorized points. In the following model, all points with greater weights are given additional weight. It will keep on training the model until a low error is obtained. For facial recognition software, a focused frame is dragged over the input image, and Haar features are calculated. The difference is then compared to a cutoff that separates faces from non-faces. Because a single Haar feature is a poor predictor for face recognition, many Haar features are expected to characterize a face with appropriate accuracy and are thus constructed as cascade classifiers.
- Cascading Classifiers: Cascading is indeed a number of different settings ensemble strategy based on the conjunction of multiple weak classifiers that utilizes the data from the result of one weak classifier as additional info for the next classification in the cascade. When each classification is generated using a boosted strategy to enhance accuracy, a composite score of the choices provided by the weak learners is taken into consideration. Every classifier step labels the region described by the sliding screen's current location with a positive or negative label. Positive and negative results show the presence of a face, respectively. If the label is negative, the region has been properly classified, and the detector moves the window to the next spot. The classifier advances the region to the following stage if the label is positive. When the last stage qualifies the region as affirmative, the face is recognized at the current window location.

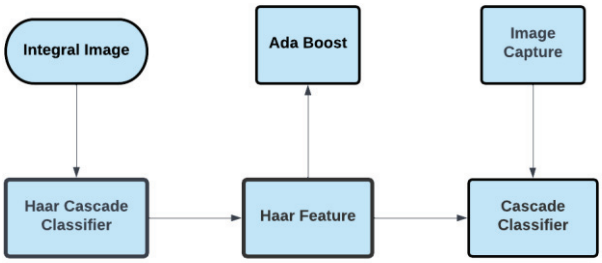

Fig. 2.

IV. System Architecture

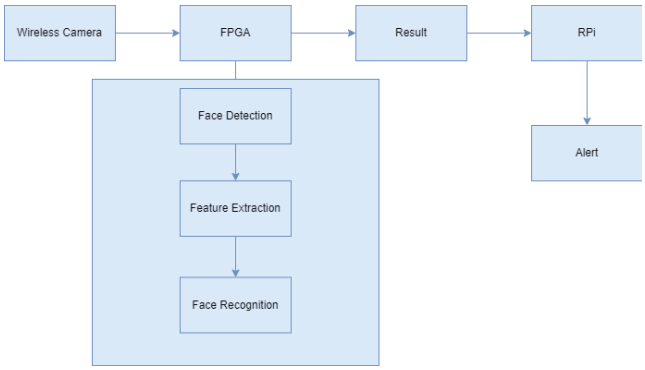

Fig. 3.

- FPGA Altera DE2-115
- WIFI Module ESP 8266
- Raspberry Pi 4 Model B-4GB

Fig. 4.

V. Result

Real-time surveillance video is sent into an FPGA for face extraction and recognition, and the result is fed into an IOT module, which hosts an application that sends alarm messages to security and other concerned parties through their mobile phones and desktop systems. As illustrated below, our facial recognition technology was able to discriminate between approved and unauthorized users. When an unauthorized individual was detected, our system displayed a red box around the intruder's face, as well as an intruder alert message on the desktop screen.

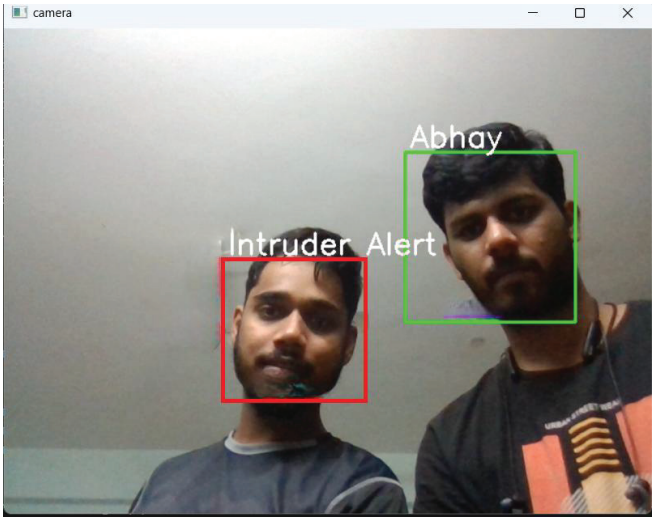

Fig. 5.

After detecting an illegal person, the system transmitted an alarm signal via a web application, as illustrated, with the intruder's location derived from the position of the camera on which the person was detected.

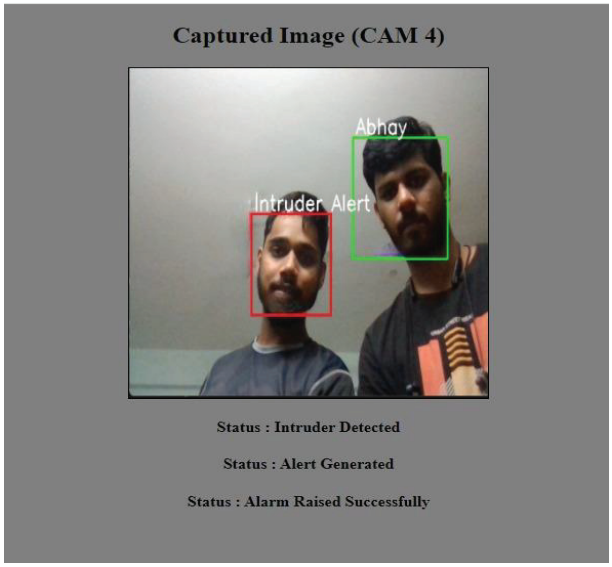

Fig. 6.

The real-time face recognizer is used in security surveillance to monitor a specific individual in real-time. This technology is beneficial when security staff is unable to watch 24 hours of Surveillance tapes. AI-like capabilities are possible with computer vision. AI on FPGA thinks smarter and quicker than previous systems where AI processing is done just on a Raspberry Pi or a standard CPU.

TABLE I.

S.NO.	Platform	Execution Time	Speed
1	FPGA[9]	53 µs	37x
2	Raspberry Pi 4B	4.6 ms	4x
3	CPU i7	2.3 ms	x

From the above table for we can conclude that the integration of raspberry pi and FPGA makes our system better than ever before where we either used raspberry pi or a CPU with a high processor for computation and processing of face recognition algorithm and for communicating the result of algorithm throughout the network.

VI. Conclusion

In this research, we provide a real-time face detection and recognition surveillance system that executes face extraction and identification on FPGA utilizing HLS synthesis. This system was created to detect an intruder in the building or any other place where the presence of an unauthorized person poses a threat. The dataset on our system contains all of the information about authorized people, including images that are used to recognize their faces. If an unauthorized person is spotted on any camera, our system generates an alert message and sends it to all of the people who are concerned about the issue, thanks to the IoT infrastructure that we have implemented on our system with Raspberry Pi. Our technology, which is used to host an application through which alarm messages are transmitted, may also be used to discover a certain individual and when the system does notice that person on any camera. Preliminary findings of the built system show that it is capable of doing all of the tasks described in a highly fluent way. Nonetheless, we have created a noble system with increased accuracy and reduced latency for greater usefulness and reliability.

References

[1] X. Zhang, A. Ramachandran, C. Zhuge, D. He, W. Zuo, Z. Cheng, K. Rupnow, and D. Chen, “Machine learning on FPGAs to face the IOT Revolution,” 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Nov. 2017.

[2] S. N. Jyothi and K. V. Vardhan, "Design and implementation of real time security surveillance system using IoT," 2016 International Conference on Communication and Electronics Systems (ICCES), 2016, pp. 1-5, doi: 10.1109/CESYS.2016.7890003.

[3] R. Saranya, C. Karthikeyan, V. S. N. Kumar and R. H. Kumar, "Computer Vision on Identifying Persons under Real Time Surveillance using IOT," 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), 2020, pp. 1-5, doi: 10.1109/ICSCAN49426.2020.9262407.

[4] L. Peng, Z. Xin and G. Ping, "Design and Implementation of Remote DeepFace Model Face Recognition System Based on sbRIO FPGA Platform and NB-IOT Module," 2019 2nd International Conference on Safety Produce Informatization (IICSPI), 2019, pp. 505-509, doi: 10.1109/IICSPI48186.2019.9095951.

[5] C. Cui, F. Ge, Z. Li, X. Yue, F. Zhou and N. Wu, "Design and Implementation of OpenCL-Based FPGA Accelerator for YOLOv2," 2021 IEEE 21st International Conference on Communication Technology (ICCT), 2021, pp. 1004-1007, doi: 10.1109/ICCT52962.2021.9657856.

[6] H. Li and W. Ye, "Efficient implementation of FPGA based on Vivado High Level Synthesis," 2016 2nd IEEE International

Conference on Computer and Communications (ICCC), 2016, pp. 2810-2813, doi: 10.1109/CompComm.2016.7925210.

[7] N. Stekas and D. 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), 2016, pp. 300-304, doi: 10.1109/IPDPSW.2016.67.

[8] Y. Zhang, W. Cao and L. Wang, "Implementation of high performance hardware architecture of face recognition algorithm based on local binary pattern on FPGA," 2015 IEEE 11th International Conference on ASIC (ASICON), 2015, pp. 1-4, doi: 10.1109/ASICON.2015.7516877.

[9] Mo, Zhuofeng & Luo, Dehan & Wen, Tengteng & Cheng, Yu & Li, Xin. (2021). FPGA Implementation for Odor Identification with Depthwise Separable Convolutional Neural Network. Sensors. 21. 832. 10.3390/s21030832.