Journal of Science and Technology
ISSN: 2362-1443 (Print) 3059-9490 (Online)
2024, Vol. 4, No. 2 (Special Issue), 24-30

Automated Attendance System using FPGA based on Face Recognition

Pujan Thapa
Dept. of Electronics and Communication, P.U.
thapakazipuzan@gmail.com

Mohan Lal Shrestha
Khwopa Engineering College, P.U.
shresthamohanlal31@gmail.com

Prabhat Tiwari
Dept. of Electronics and Communication, P.U.
prabhattiwari867@gmail.com

Ranjan Thapa
Khwopa Engineering College, P.U.
thaparanjan32@gmail.com

Article History:
Received: 30 July 2024
Revised: 20 August 2024
Accepted: 14 November 2024

Keywords— *FaceNet,FPGA, MTCNN, Proxy, PS, PL.*

***Abstract*—The manual method of taking attendance involves the director or the school instructor doing it by hand, which takes a lot of time and effort and involves mistakes and proxy attendance. Automated systems involving the use of biometrics like fingerprint and iris recognition have been well developed in recent times, but they are economically highly expensive for deployment on a large scale. Biometric features such as face recognition can be utilized to address these problems,which involve several stages, including image gathering, face identification, feature extraction, facial recognition, and finally recording the attendance. In this design, we have attempted to develop a successful and effective " Automated Attendance System Using FPGA Based on Facial Recognition". The goal of this design is to detect and recognize the face in a captured image or frame using MTCNN and FaceNet for attendance purposes using FPGA, where we have explored two approaches: PS-PS(Processing System) and PS-PL (Programmable Logic)approach. Our system achieved an accuracy of 88.7 percent using the PS-PS approach, and 72.5 percent with the PS-PL approach. While the PS-PL approach demonstrated reduced accuracy, it offered improved performance and decreased latency, making it a promising alternative for real-time applications in automated attendance.**

I. INTRODUCTION

Checking the performance of students and maintaining attendance is a tedious process for colleges, universities or institutes[1]. Each institution has adopted its own method of taking attendance, such as calling out names or passing around attendance sheets. These traditional methods suffer from several shortcomings. For instance, manually entering a lot of students attendance is labor-intensive, prone to error, and time-consuming. Manual systems are susceptible to errors like miscounting, false entries, and proxy attendance, leading to unreliable data.

Two of the most extensively used technologies are RFID and fingerprint technology which are used for automatic attendance systems available today. However, these systems often require queuing, which can be time-consuming and intrusive. Damage to an RFID card can result in inaccurate attendance records. Furthermore, it may not be economical to implement thesesystems widely[11].

Facial recognition technology offers a viable way to overcome these challenges and develop a system that is both time and cost-efficient while requiring little human involvement. A key technique for identifying people is facial recognition, and as image processing advances and techniques like pattern recognition and signature recognition are developed, facial recognition systems become more efficient.

This system aims to provide an automated attendance sys tem that uses facial recognition technology to record students attendance in lectures or classes and maintain a database of attendance records. Once the database of students/candidates is created, the system requires minimal effort from the user. This non-intrusive nature makes the system effective and efficient.

II. RELATED WORK

Facial recognition-based attendance system uses Convolutional Neural Network models like MTCNN and FaceNet, which handle face detection and recognition. The results indicate that the suggested architecture offers a viable way to manage student attendance in classrooms[1]. The framework maintains real-time performance while outperforming state-of-the-art methods on the challenging face detection dataset and benchmarks like WIDER FACE for face detection and annotated facial landmarks in the wild benchmark for face alignment. It accomplishes this by utilizing a cascaded architecture with three stages of carefully constructed deep convolutional networks to predict face and landmark locations in a coarse-to-fine manner[7]. An innovative and economical method is to implement and deploy a smart automated attendance system model on FPGA, which recognizes faces in real-time and compares them with an existing database to identify each one and register the attendance[6]. This article suggests using an Altera Cyclone-IV FPGA—a potent and effective embedded device—to construct a face recognition-based surveillance system.

III. METHODOLOGY

Block diagram of our proposed work is given below:

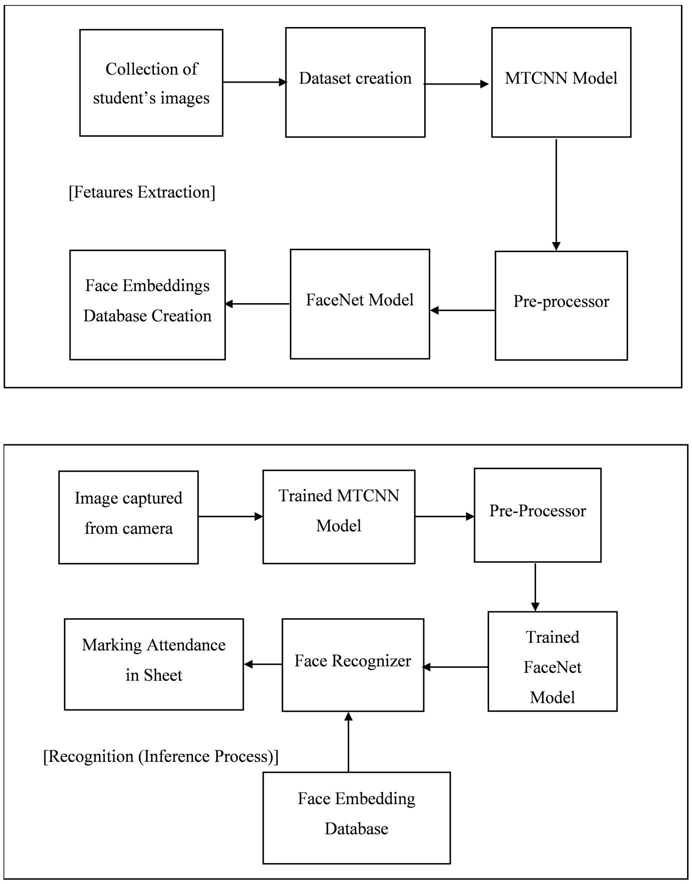

Fig. 1. Proposed Block Diagram

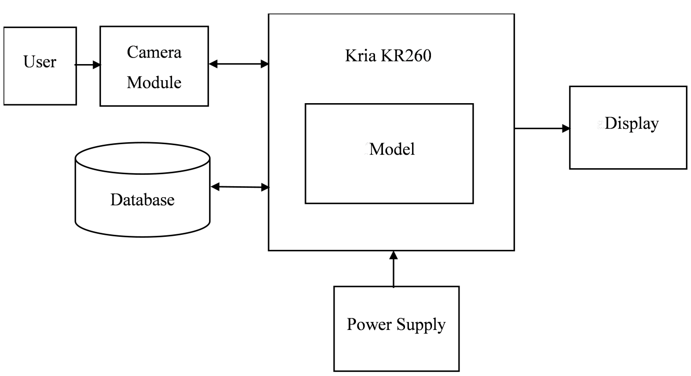

Fig. 2. Proposed Block Diagram

A. TMCAM 8305 720P High Resolution Web Camera

Fig. 3. TMCAM8305 CMOS Camera[18]

The TMCAM 8305 720P High-Resolution Web CMOS technology camera records video at a 30 frames per second (fps) frame rate in 720p resolution (1280 × 720 pixels). With a 6-layer glass-coated lens and a 2-megapixel CMOS sensor, it offers excellent resolution and accuracy.

B. The Kria KR260 Robotics Starter Kit

Fig. 4. The Kria KR260 Robotics Starter Kit[17]

A potent Xilinx Zynq UltraScale+ MPSoC, comprising an Arm Mali-400 MP2 GPU, a dual-core Arm Cortex-R5 real-time CPU, and a quad-core Arm Cortex-A53 processor, powers the Xilinx Kria KR260 Robotics Starter Kit. It is the perfect platform for creating and testing robotics and embedded vision applications because it has 4GB of LPDDR4 memory.

IV. FACE DETECTION ALGORITHM

In our proposed work we have used MTCNN to know the location of an image (face detection). A deep convolutional neural network-based method for face alignment and identification is called Multi-Task Cascaded Convolutional Neural Networks (MTCNN). Multi-tasking CNN, or MTCNN, is capable of doing two tasks at once: face alignment and face detection. MTCNN offers superior detection performance compared to previous approaches, accurately pinpointing the position of the face and meeting real-time detection requirements. The MTCNN model first resizes an image to different scales to create an image pyramid, which serves as the input for the PNet which is fully convolutional neural network. The original image input is used to construct the multi-scale image pyramid, which is then utilized to generate candidate regions through the complete convolution form of PNet. In RNet, non-maximum suppression is applied to eliminate candidate boxes with the highest degree of overlap, and in ONet, face regions are identified with more supervision. The network outputs five facial landmark positions, as shown in the structure diagram of the MTCNN model in Figure below.

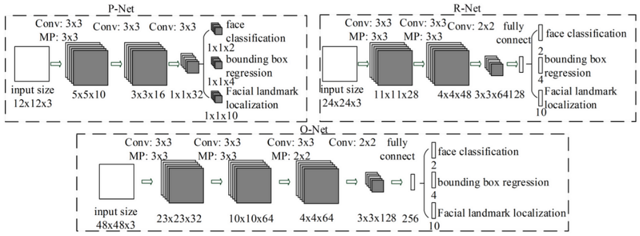

Fig. 5. Three networks of MTCNN[7]

V. FACE RECOGNITION ALGORITHM

Face recognition refers to a technique that uses a person's face to identify or confirm their identification. For our work, we have used the face recognition model FaceNet and feature matching algorithn Cosine Similarity.

A. Facenet

In order to extract information from a facial image, we used a deep neural network known as FaceNet. FaceNet creates a 128-dimensional embedding vector comprising the essential aspects of a face from a picture of the person face. This network includes Inception ResNet V1 architecture which is trained on two dataset which are CASIA-WebFace and VGGFace2 that have accuracy of 99.05 percent and 99.65 percent respectively. Features matching is performed using these 128-dimensional embeddings of each person, which are stored in a database along with their name. The input image size of facenet is 160x160.

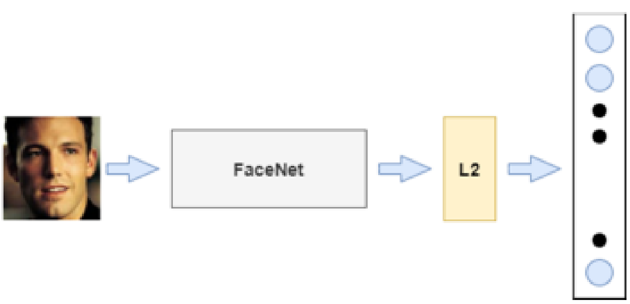

Fig. 6. FaceNet model structure[8]

B. Cosine Similarity

We have used the Cosine Similarity method for face recognition, which measures the similarity between embedding vectors in the database and the embedding vector of faces from images (frames) taken by the camera in real-time. By using the cosine similarity coefficient, one can calculate the cosine of the angle created by two vectors. It is a measure of similarity that ranges from -1 to 1. If we take any two vectors A and B, then its cosine similarity is given by:

$$cosinesimilarity = A.B/|A||B| \tag{1}$$

where 'A' and 'B' are two vectors. In our case 'A' is embeddings of faces taken by camera at real time and 'B' is embeddings stored in database.

VI. PROPOSED SYSTEM

This study focuses on creating a smart attendance system that applies a combination of face detection and face recognition models. The face detection model used is MTCNN by Zhang [7], and the face recognition model for feature extraction is Facenet by F. Schroff [8], along with the Cosine Similarity algorithm for feature matching. The proposed methodology is illustrated in the figure below.

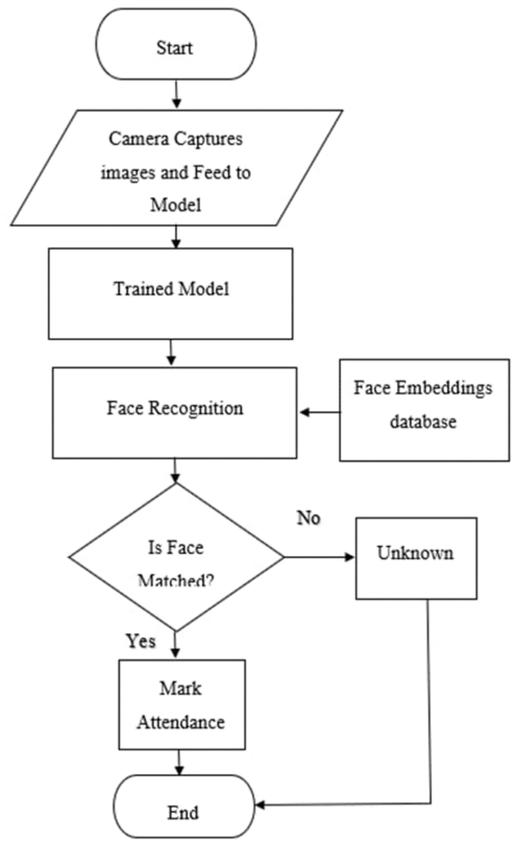

Fig. 7. Flowchart for automated attendance system

A. Data Collection

In order to generate our facial recognition model, we need to create a dataset from individual face images. It is necessary to compile a dataset with a number of images (we have used 200 images per individual with different face poses) for each individual to enhance the performance of our learning model. This stage involves capturing the image of each individual and extracting the region of interest, which in our case is the face.

B. Face Detection

It's the initial stage of a facial recognition system. Selecting the right face detection algorithm is essential to the facial recognition system's smooth operation. Face detection techniques vary and include machine learning-based approaches and feature-search based approaches based on skin tone. Among all these methods, we choose to work with a deep learning network (MTCNN) which can perform two tasks simultaneously: face detection and face alignment. The MTCNN model first resizes an image to different scales to create an image pyramid, which serves as the input for the PNet which is fully convolutional neural network. The original image input is used to construct the multi-scale image pyramid, which is then utilized to generate candidate regions through the complete convolution form of PNet. While RNet use non-maximum suppression to exclude candidate boxes with the highest degree of overlap, ONet uses greater supervision to identify face areas. The network outputs five face landmark points.

C. Image Preprocessing

The detected faces are extracted and must be pre-treated to meet the expectations of our CNN. This preprocessing phase involves resizing the face image to a square input size.

D. Feature Extraction

There are various methods for feature extraction such as PCA (Principal Component Analysis), HOG (Histogram of Oriented Gradients), FaceNet, etc. Among these methods, we used FaceNet, which is a deep neural network used to extract features from an image of a person's face. FaceNet takes an image of a person's face as input and generates a 128-dimensional embedding vector that represents the most important features of a face. This network includes Inception ResNet V1 architecture which is trained on two dataset which are CASIA-WebFace and VGGFace2 that have accuracy of 99.05 percent and 99.65 percent respectively. Then, these 128-D embeddings of each individual along with their name are stored in a database and used in feature matching.

E. Features Matching

The Cosine Similarity method is used for face recognition to measure the similarity between embedding vectors in a database and the embedding vector of faces from images (frames) taken by a camera in real-time. Cosine similarity measures the cosine of the angle between two vectors and is a measure of similarity that ranges from -1 to 1.

VII. Implementation of Model into FPGA

During our proposed work, we explored two approaches: the PS+PS approach and the PS+PL approach.

A. PS+PS Approach

In the first approach, the PS+PS approach, we ran both the detection model and recognition model in the processing system only of the FPGA. We implemented our project in the PS of the Kria KR 260 Robotics Starter Kit, which has much lower performance compared to using the PL of the Kria KR 260 Robotics Starter Kit. The frames per second taken and marking attendance were not happening in real-time, which means there was a delay in marking attendance. Since we are only using the PS of the Kria board, there was a delay in the whole process, including detection of faces, face recognition, and marking attendance in Google Sheets. Therefore, to improve the performance and overcome this limitation, we are going to quantize and compile our model using Vitis AI, which is the PL approach.The workflow of the PS+PS approach is shown in the flowchart in Figure 8.

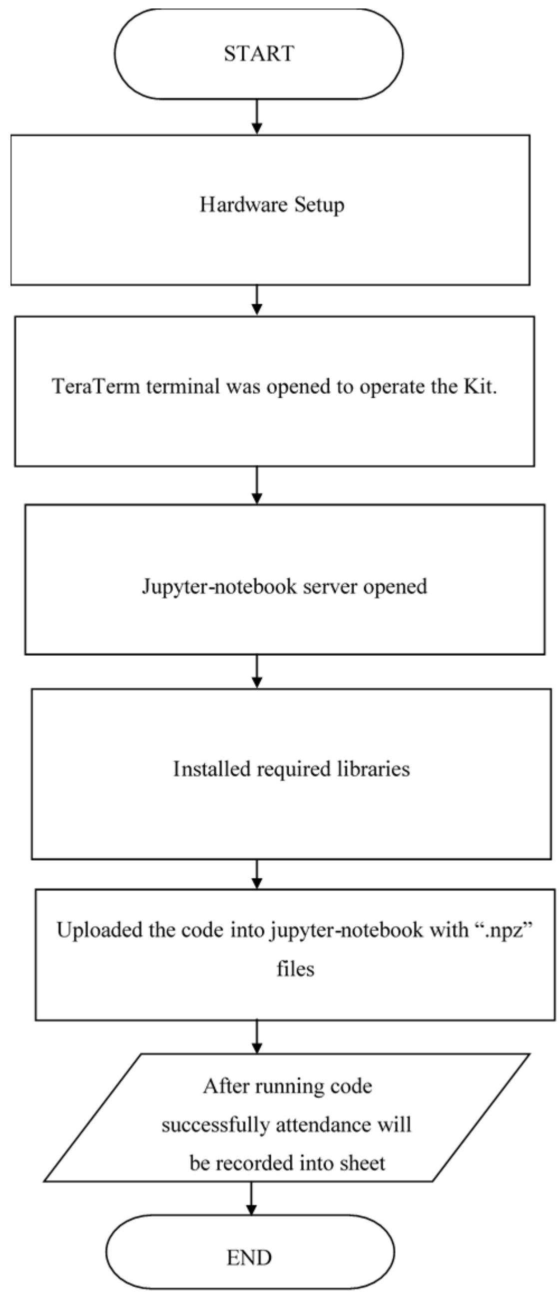

Fig. 8. Flowchart for Implementation of Model into FPGA(PS+PS Approach)

B. PS+PL Approach

In the second approach, the PS+PL approach is used to enhance the performance of our system. For the FPGA Board implementation, we are using the PYNQ flow. Since we are

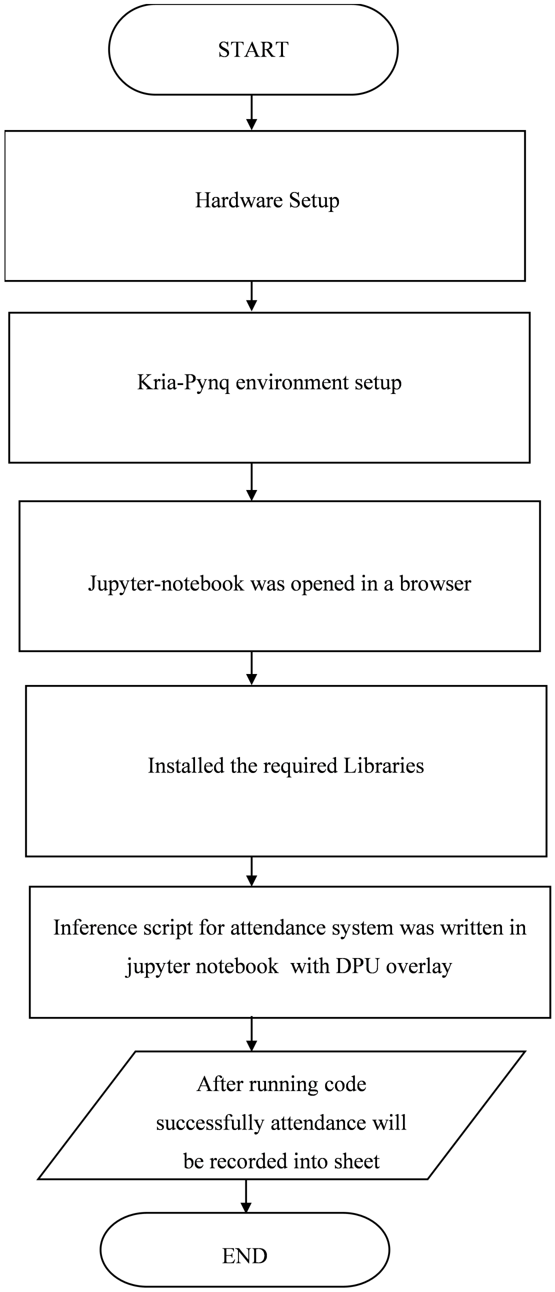

Fig. 9. Flowchart for Implementation of Model into FPGA(PS+PL Approach)

going through the PS+PL approach, we run the MTCNN model in the PS of the Kria KR260 Robotics Starter Kit and run the Facenet model in the PL of the Kria KR260 Robotics Starter Kit. Therefore, we have quantized and compiled our Facenet model, making it suitable for running on the FPGA board successfully and effectively. After quantization and compilation, we need to write our final inference script using Vitis-AI APIs. Finally, we have to go through the bring-up process, which involves running our AI application, i.e., executing our program on the FPGA board and checking the performance of the proposed system. This has resulted in an improvement in performance and a reduction in delay in the entire process, including face detection, face recognition, and marking attendance in Google Sheets, overcoming the PS+PS approach's shortcomings. The workflow of the PS+PL approach is shown in the flowchart in Figure 9.

C. Quantization

It is the process of converting the 32-bit floating-point ('float32') weights and activations into fixed-point formats like 'int8'. Due to this, a fixed-point network model will demand less memory bandwidth, leading to faster processing speed and improved power efficiency compared to the floating-point model.Additionally, quantization can significantly reduce latency, enhancing the responsiveness of real-time applications.

D. Compilation

It involves mapping AI/DL model to the extremely effective instruction set and dataflow model. It also performs sophisticated optimizations such as layer fusion, instruction scheduling, and reusing on-chip memory as much as possible. The flowchart to convert float model into fixed point model is shown below:

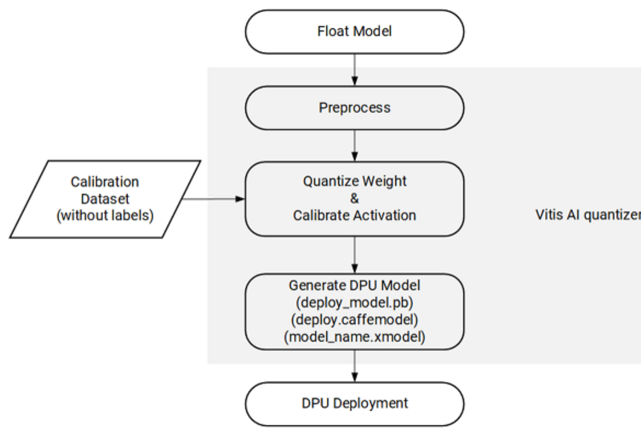

Fig. 10. Vitis-AI Quantization and Compilation[16]

Algorithm 1 Algorithm For Marking Attendance
Input: Frames
Output: Marking Of Attendance

1: Start.
2: Camera opened and captures the frames.
3: Send the frame to the MTCNN Detector that detects the faces on that frame.
4: Output from MTCNN are preprocessed.
5: Sends faces to the FaceNet model that generates embeddings for each unique faces.
6: The similarity between embeddings generates at real time and embeddings on database are calculated.
7: If the similarity value is greater than the given threshold value, mark the attendance of registered students; otherwise, mark them as unknown in the Google Sheet in real-time with the timestamp.
8: End.

VIII. RESULTS

We implemented the MTCNN and Facenet model in the the Kria KR260 Robotics Starter Kit using PYNQ-flow. The outputs,accuracy curve and time taken for PS+PS and PS+PL approach to mark the attendance in google sheet is shown in Fig.13, Fig.14,Fig.11,Fig.12 and in the TABLE 1 shown below:

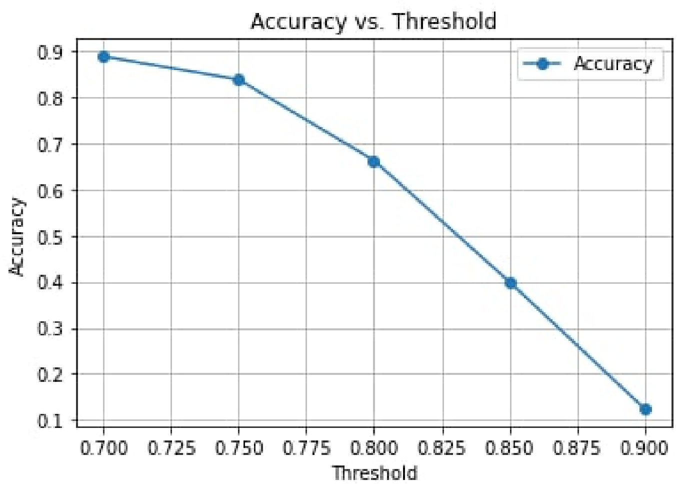

Fig. 11. Accuracy curve(PS+PS)

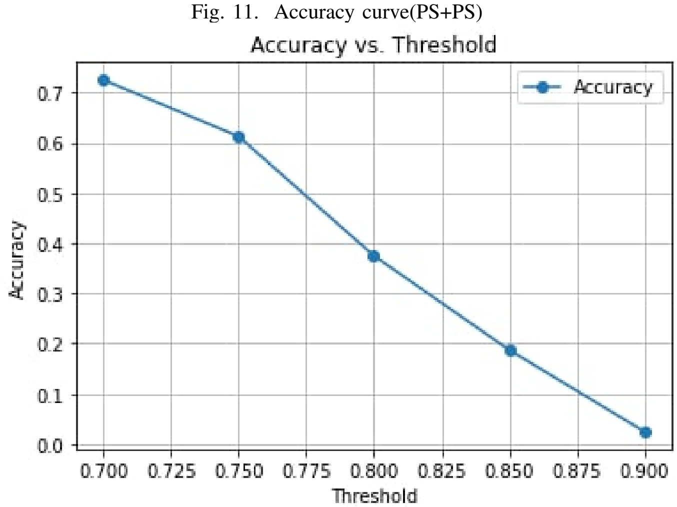

Fig. 12. Accuracy curve(PS+Pl)

TABLE I
DELAY TABLE

S.N.	Approach	Time Taken (sec)
1.	PS+PS	17.6733
2.	PS+PL	8.9639

Here the time taken means duration between frame taken from camera and recognition phase.

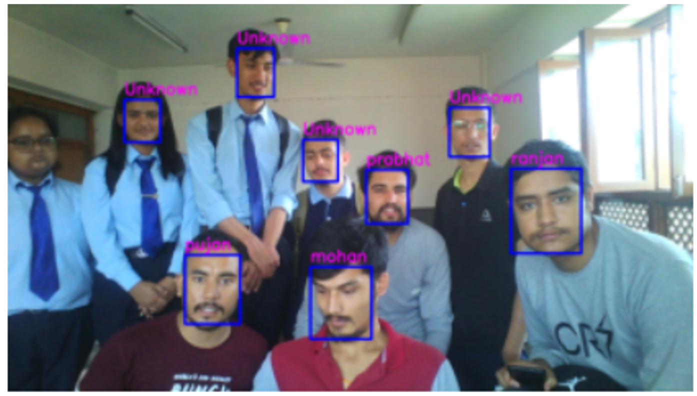

Fig. 13. Output of PS+PS Approach

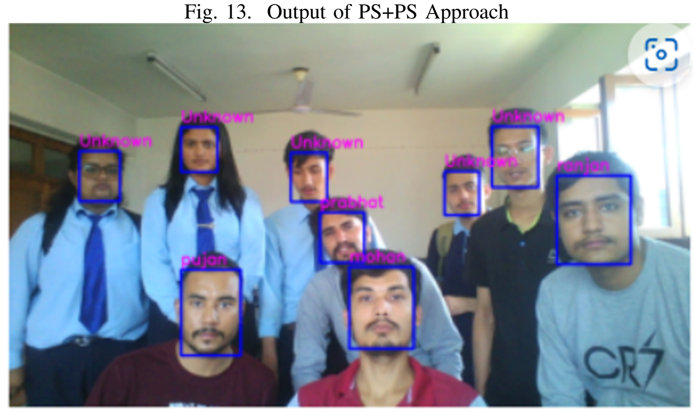

Fig. 14. Output of PS+PL Approach

IX. CONCLUSION AND FUTURE WORK

Automated Attendance System Using FPGA Based on Face Recognition was implemented successfully. This system detects faces from images taken by the camera in real-time, compares their embeddings with the database embeddings, and marks the attendance of registered students only in the Google Sheet. The accuracy for the PS+PS approach was found to be 88.7 percent, and the accuracy for the PS+PL approach was found to be 72.5 percent. Although the accuracy of the PS+PL approach is lower than the PS+PS approach, there is an improvement in performance and a reduction in delay in the entire process, including face detection, face recognition, and marking attendance in Google Sheets. The decrease in accuracy is attributed to the quantization of the facenet model. In the future, we can quantize and compile the mtcnn model as well and proceed with the PL+PL (PYNQ flow) approach or the Petalinux flow, which will significantly enhance and and decrease latency while enhancing system performance.

REFERENCES

[1] T. Hachad, A. Sadiq and F. Ghanimi, ”Student's attendance management using deep facial recognition,” in IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), 2020.

[2] S. Kumar, D. Kathpalia, D. Singh and M. Vats, ”Convolutional Neural Network based Automated System by using Facial Recognition,” in Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2020), 2020.

[3] H. Acharya, N. Khadka and R. Jamkatel, ”Face Recognition Based E-Attendance Using Raspberry Pi,” Khwopa Engineering College, 2019.

[4] S. Kadambari, D. Mistry, G. Prabhu and P. Khanore, ”Automation of Attendance System Using Facial Recognition,” 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), 2019.

[5] P. Patil, ”Video Analysis for Face Recognition in Real Time Attendance Monitoring System,” International Journal of Current Engineering and Technology, 2021.

[6] A. Shanker, D. Roy and M. P. M, ”Implementation of Smart Attendance on FPGA,” M.Tech VLSI Design, Dept. of Micro and Nano electronics, SENSE, Vellore Institute of Technology, Vellore, Tamil Nadu,, 2019.

[7] K. Zhang, Z. Zhang, Z. Li and Y. Qiao, ”Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks,” IEEE, 2016.

[8] F. Schroff, D. Kalenichenko and J. Philbin, ”FaceNet: A Unified Embedding for Face Recognition and Clustering,” IEEE, 2015.

[9] A. Alahmadi and S. M. Qaisar, ”Robust Real-time Embedded Face Detection Using Field Programmable Gate Arrays (FPGA),” International Journal of Current Engineering and Technology, 2019.

[10] A. J. A. AlBdairi, Z. Xiao, A. Alkhayyat and A. J. Humaidi, ”Face Recognition Based on Deep Learning and FPGA for Ethnicity Identification,” MDPI, 2021.

[11] Hemantkumar Rathod, Yudhisthir Ware, Snehal Sane, Suresh Raulo, Vishal Pakhare and Imdad A. Rizvi, ”Automated Attendance System using Machine Learning Approach,” International Conference on Nascent Technologies in the Engineering Field (ICNTE-2017), 2017.

[12] Sivashankar S, Renita Rexy, ”A Real Time FPGA Based Face Recognition System for Attendance Management and Power Control in Institutions Corporates,” Global Journal For Research Analysis (GJRA), 2014.

[13] GALUH PUTRA WARMAN, GEDE PUTRA KUSUMA, ”FACE RECOGNITION FOR SMART ATTENDANCE SYSTEM USING DEEP LEARNING,” Communications in Mathematical Biology and Neuroscience (CMBN), 2023.

[14] Qasim Al-Shebani, Prashan Premaratne and Peter Vial, ”Embedded Door Access Control Systems Based on Face Recognition: A Survey,” IEEE, 2013.

[15] Edwin Jose, Greeshma Manikandan, Mithun Haridas T P, Supriya M H, ”Face Recognition based Surveillance System Using FaceNet and MTCNN on Jetson TX2,” 5th International Conference on Advanced Computing and Communication Systems (ICACCS), 2019.

[16] ”AMD XILINX,” [Online]. Available: https://xilinx.github.io/xup-vitis-ai-tutorial/.

[17] AMD. [Online]. Available: https://www.amd.com/en/products/system-on-modules/kria/k26/kr260-robotics-starter-kit.html.

[18] ”FCCMOVIL,” [Online]. Available: https://www.fcc-movil.com/producto/camara-web-tmcam-8305-720p-alta-resolucion/.