Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 6, No. 1, 2023

Implementation of Facial Recognition System for Metaverse using sbRIO FPGA and NB-IOT Module

Peng Lu[1, a], Meijiao Yu[2, b], Renyong Zhang[2, c]

[1]Chongqing University, China
[2]Chongqing Institute of Engineering, China
[a]552437915@qq.com, [b]1156353459@qq.com, [c]913213836@qq.com

Abstract: Facial recognition is crucial for identity verification in the metaverse, but it requires significant processing and operation overhead. Transmitting high-definition images to a server PC for processing is not feasible in low-capacity, low-bandwidth, or low-processor virtual environments. To overcome these challenges, we developed a narrow-bandwidth framework integrating embedded FPGA technology with a low-power NB-IOTcommunication module. Our approach uses a DNN-based DeepFace model with front face detection and 7-layer DNN convolution result extraction performed on the Zynq FPGA chip of sbRIO , reducing computational overhead and enabling efficient processing. By leveraging NB-IOT's remote transmission capabilities, classification data is transmitted back to the local server for comparison. Our proposed framework improves speed and accuracy while overcoming bandwidth and processing power challenges, making it a promising solution for facial recognition in immersive virtual environments.

Keywords: Metaverse, FPGA, DeepFace, NB-IOT.

1. Introduction

This research presents the design and implementation of a remote, low-bandwidth face recognition system for the metaverse, utilizing the sbRIO FPGA [1] hardware platform and NB-IOT module [2]. The system architecture can be divided into four functional modules: high-definition camera acquisition, FPGA-based DNNs [3] utilizing the DeepFace Model [4], NB-IOT remote communication, and local server reception, as shown in Figure 1. The proposed approach enables efficient and accurate face recognition within the constraints of the metaverse, providing a practical solution for immersive virtual environments. This system has significant potential for various metaverse applications, such as virtual social interactions, virtual commerce, and virtual entertainment. Our research contributes to the development of advanced face recognition techniques in the context of the metaverse, providing a valuable contribution to the field of virtual reality and immersive technology.

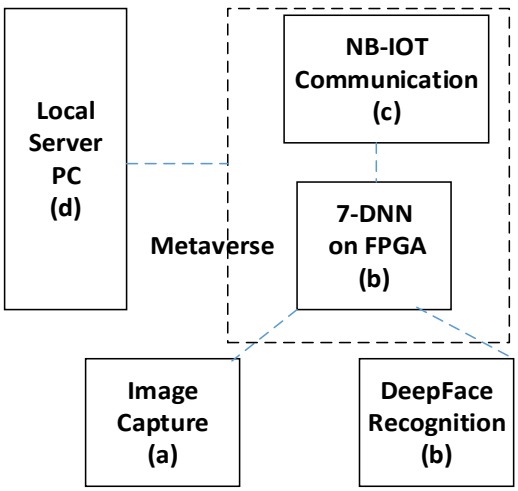


Figure 1. System function structure chart

The proposed system for efficient and accurate face recognition in the metaverse involves four functional modules as figure 1: (a) high-definition image acquisition, (b) DNN-FPGA operation,(c) NB-IOT remote communication, and (d) local server comparison. The 2-megapixel OV5640 camera module captures high-definition images [5]. On the sbRIO FPGA platform, the 7-layer DNN of the DeepFace model is completed, generating (32*32*8 bits)1024-bytes classification data. The NB-IOT module with the CoAP protocol sends the data back to the remote server [6], and the classification data is compared with the primary key in the local MySQL database to display real-time face recognition results. The system offers a practical solution for low-bandwidth face recognition in the metaverse, with potential applications in virtual social interactions, virtual commerce, and virtual entertainment.

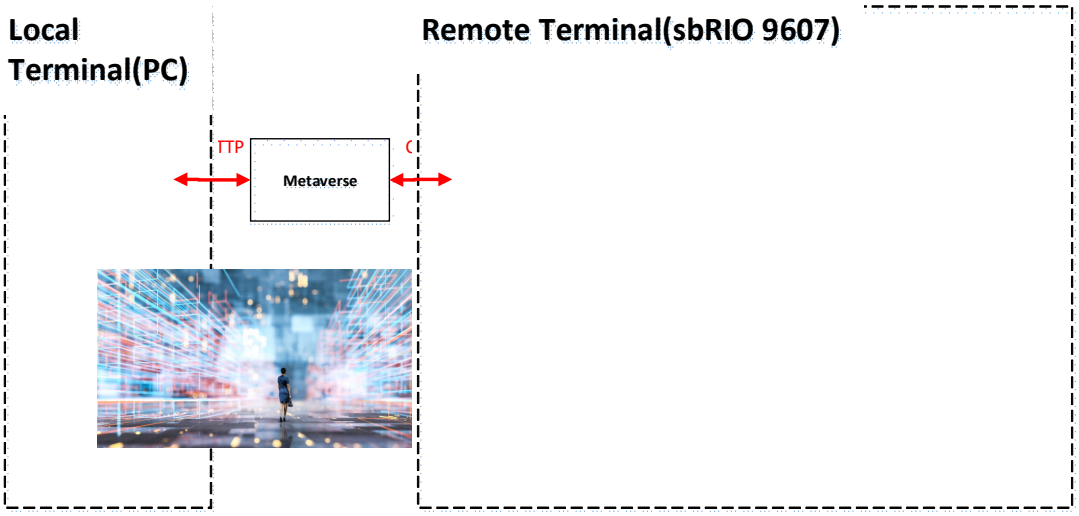


Figure 2. Hardware structure chart

The hardware structure for the proposed system is divided into two parts as figure 2: the local terminal and the remote terminal. The remote terminal consists of the FPGA hardware, camera module, and NB-IOT module. DNNs are implemented on the FPGA resource using LabVIEW to run the DeepFace model. The classification data is transmitted through the NB-IOT module to the local terminal, where it is compared with the primary key in the local MySQL database. Both local and remote terminal development processes are done using LabVIEW, offering a practical solution for low-bandwidth face recognition in the metaverse with potential applications in virtual social interactions, commerce, and entertainment.

The local terminal used for comparison operations has an Intel i7 CPU, 8GB DDR4 RAM, and Gigabit Ethernet, while the remote terminal includes an OV5640 camera module, a Zynq 7020 chip, and an NB-IOT module with a baud rate of 115200bps. The remote terminal has a maximum current of 268mA and a receive sensitivity of -129dBm, and the transmit power ranges from 23dBm to -40dBm.

2. Implementation of Image Acquisition on FPGA and DeepFace Model on Remote Terminal

2.1. Image Acquisition On FPGA

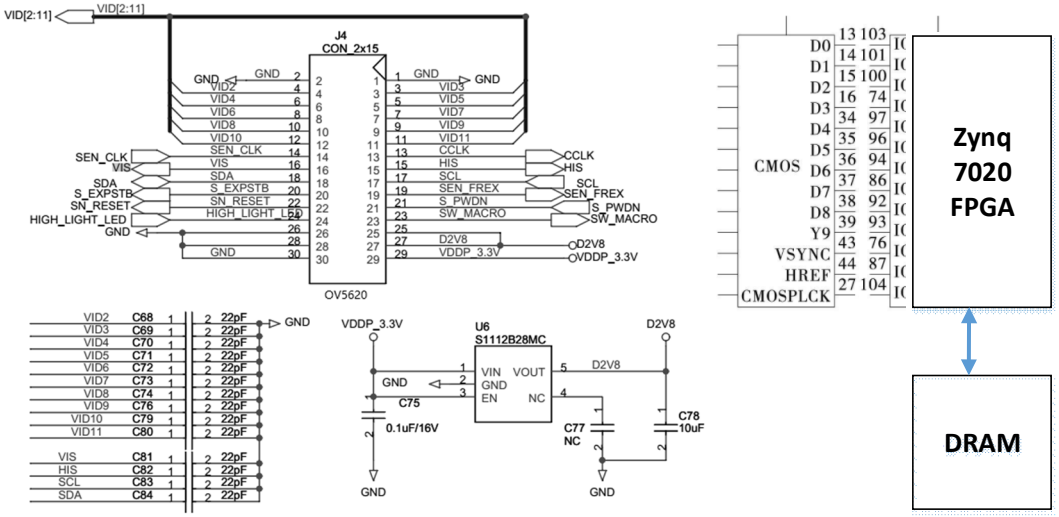


Figure 3. Schematic Circuit Diagram of CMOS Sensor with Zynq 7020 FPGA

As figure 3, the schematic circuit diagram describes a streamlined image acquisition framework developed using a stable VHD IP core of the OV5620 Sensor to improve the efficiency of image acquisition. The OV5620 image sensor has a resolution of 2592*1944 and a data bit width of 10 bits, and outputs RAW RGB in progressive scan mode [7]. The image sensor is configured through the SCCB protocol to enter the working state. The falling edge of the frame synchronization signals VSYNC and HREF is then evaluated within the FPGA to determine the start of a new frame. After a new frame starts, the OV5620 transfers the original image data to the FPGA under the synchronization of PCLK. Finally, each frame of the image data is continuously cached into DRAM. The described framework allows for streamlined and efficient image acquisition in face recognition systems.

2.2. 2.2 Implementation of Face Recognition in DeepFace model

This work utilized a Zynq 7020 FPGA to build a 7-layer DNN for face recognition using the DeepFace model. Recent research has shown that the DeepFace model has achieved human-like precision in face recognition benchmarks. The

section presents an efficient method to deploy the 7-layer DNN on the FPGA and implement the DeepFace model.

2.2.1. Related Work

While traditional machine learning methods have difficulty handling large amounts of data, deep neural networks (DNNs) have shown to scale well. Specifically, LabVIEW FPGA DNN Solver for fully connected Multi-Layer Perceptron (MLP) [8] with ReLU activation was utilized in this work to leverage the resources of the Zynq 7020 FPGA.

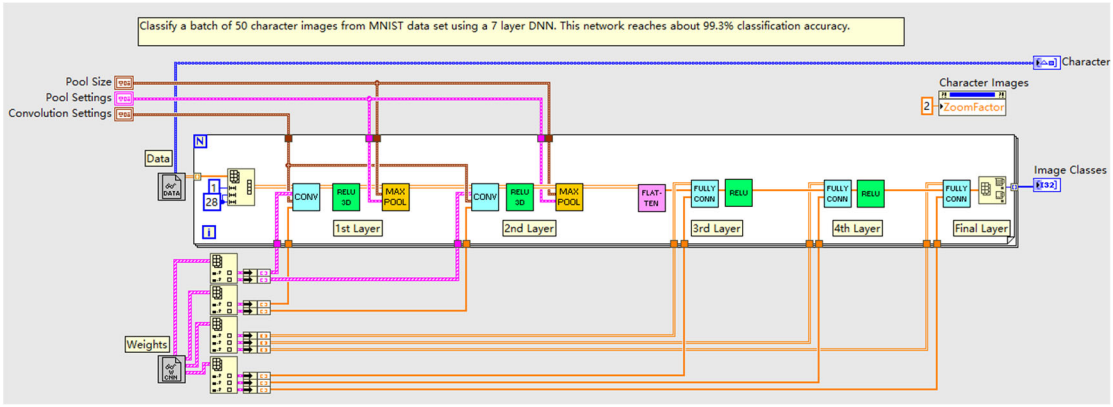


Figure 4. LabVIEW FPGA code function diagram of Deep Neural Network with ReLU 2-Layer

The FPGA code for a 7-layer DNN using the solver is shown in Figure 4. A layer-enabled handshake is implemented to minimize delay. The top layer triggers the layer below and reads its first output immediately after writing. On the Zynq-7020 sbRIO-9607, up to a 7-layer deep neural network was fitted to execute the DeepFace Model. The core algorithm (pseudo code) of each layer in DNNs is shown in Figure 5.

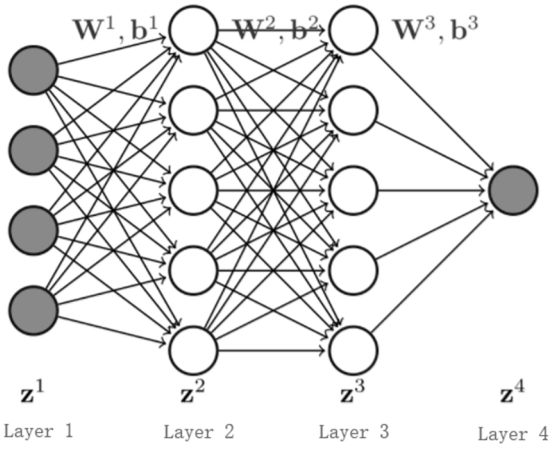


Figure 5. Diagram of DNN Structure[9]

Suppose the$L(L \geq 2)$ layer neural network, let $z^l, 1 \leq l \leq L$ denote the variable of the l layer, then The relationship between the two layers is as follows:

$$a^{l+1} = W^l z^l + b^l$$
$$z^{l+1} = f^{l+1}(a^{l+1})$$

The objective of this FPGA DNN solver is to attain maximum performance of the model resources with minimal usage of FPGA resources. The pipelined floating-point FPGA code ensures that each mathematical operation (Multiplication, Addition, ReLU) is performed on each FPGA clock cycle without any idle time. As each layer is executed in parallel, the loop rate is not impacted by adding more layers. An MNIST example is included in all machine learning tools, and Figure 6 depicts the MNIST example application running. The execution time per image on the Zynq 7020 FPGA is 150 microseconds in clock ticks.

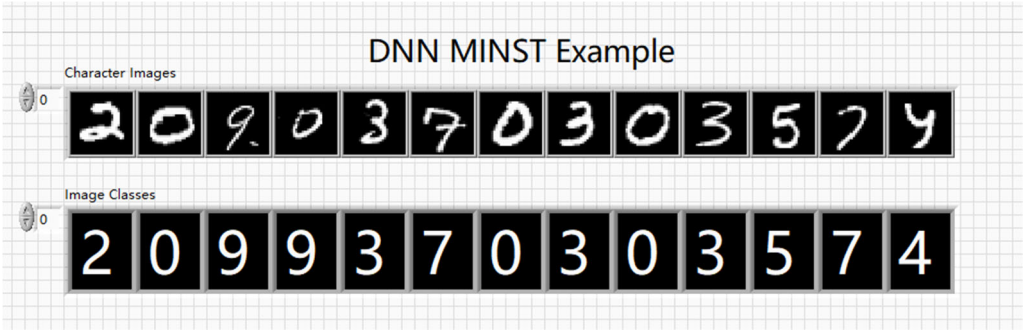


Figure 6. MINIST Example on sbRIO Zynq 7020 FPGA

The remote terminal achieved successful execution of a 7-layer DNN with 16 inputs, 100 neurons per layer, and 16 outputs at 25 kHz using their FPGA-based solver. The execution speed is not affected by the number of layers in the architecture, but the number of layers is limited by the DSP cores available on the target device. In the case of Zynq-7020, the number of layers that can be implemented is limited by the amount of block RAM available, which is approximately 4.8 Mbits, enough to store about 150,000 weights and biases or roughly 7 layers with 150 neurons per layer. The performance achieved is sufficient for high-bandwidth DeepFace Model applications.

2.2.2. DeepFace Model

The DeepFace Model involves four processes: detection, alignment, representation, and classification. For alignment, the model uses 3D alignment, which requires significant computational resources. To overcome this, LBP histogram was used for alignment due to the limited computing abilities of the Zynq 7020's ARM Core. The LBP histogram extracts texture features by counting the number of occurrences of pixels in each area. SVR processing is then applied to extract the face and its six basic points. The image is cached in DRAM before processing. Face detection results are shown in Figure 7 using LabVIEW RT Code.

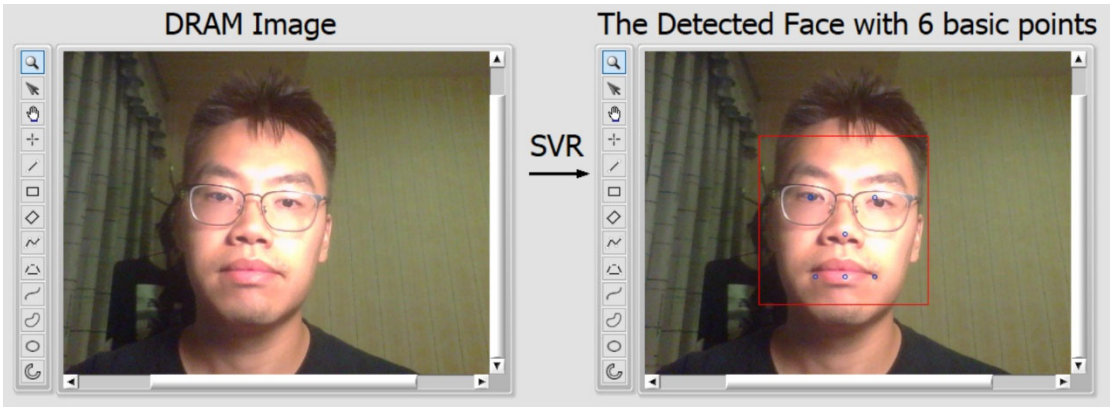


Figure 7. Face Detection in LabVIEW RT Code

DNN Architecture and Training We trained a DNN for a multi-class face recognition task, specifically for classifying the identity of a face image. The architecture of our model is illustrated in Figure 8.

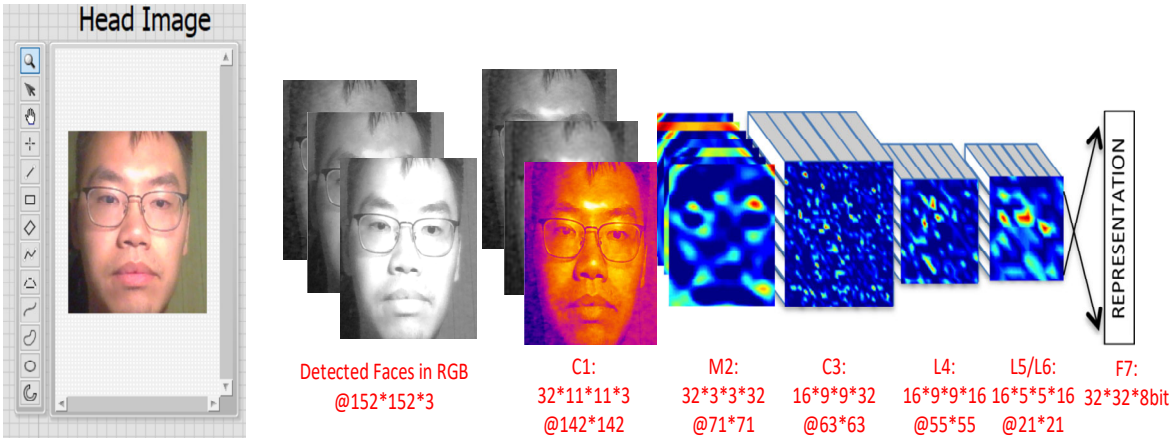


Figure 8. Outline of the DeepFace Model architecture

The FPGA-based 7-layer DNN architecture efficiently runs the DeepFace model using shared and non-shared convolutional layers and fully connected layers to extract features and classify face identity. The convolutional layers involve multiple layers, including C1, M2, C2, L4, L5, L6, and F7 as figure 8, each utilizing different convolution kernel sizes. The F7 layer is a fully connected layer with 1024 bytes corresponding to the extracted face features.

The FPGA's 7 layer DNN is efficient and can be utilized in metaverse applications such as advanced facial recognition systems in virtual reality environments. Non-shared convolutional layers are effective in face recognition tasks, avoiding information loss. The DeepFace model uses dropout and max pooling to prevent over-fitting and enhance robustness. The model achieves high accuracy in multi-class face recognition tasks and provides an efficient solution for real-time face recognition, which is in high demand for applications like security, surveillance, and biometrics.

The DeepFace model generates its final output by applying a softmax activation function to the output of the last layer. This results in a probability distribution over the possible classes in the face recognition task, with each element of the output vector representing the probability of the input image belonging to a specific class [10]. The softmax function

ensures that the sum of all probabilities is equal to 1, making the results easier to interpret.

The probability assigned to the k-th class can be calculated as the output of the softmax function on the k-th element of the network's output vector using the following formula.

$$p_k = exp(o_k) / \sum_h exp(o_h)$$

The FPGA computes the softmax function on the output vector of the last layer of the DNN to generate a probability distribution that assigns a likelihood to each of the 1024 bytes. This output is then sent back to the server computer for further processing. The classification data can be used to identify the specific person in the image by matching the probability distribution to the pre-trained face database, which involves comparing the features extracted from the input image with the features stored in the database and selecting the most likely match based on the probability distribution.

3. Implementation of Receiving Data from NB-IoT and Local Terminal into the Metaverse

3.1. NB-IOT Module

The Narrow Band Internet of Things (NB-IoT) is an important component of the Internet of Everything (IoE) in the metaverse, as it allows for reliable and cost-effective transmission of data from IoT devices. This technology uses a cellular network and requires only a small bandwidth of around 180 kHz, which reduces deployment costs and allows for smooth upgrades on existing networks. The Zynq 7020 FPGA is connected to an NB-IoT module via its RS232 port, enabling the transmission of classification data to the local server for further processing and identification of the specific person. This implementation demonstrates the effectiveness of NB-IoT in enhancing the functionality and efficiency of IoT systems in the metaverse as in figure 9.

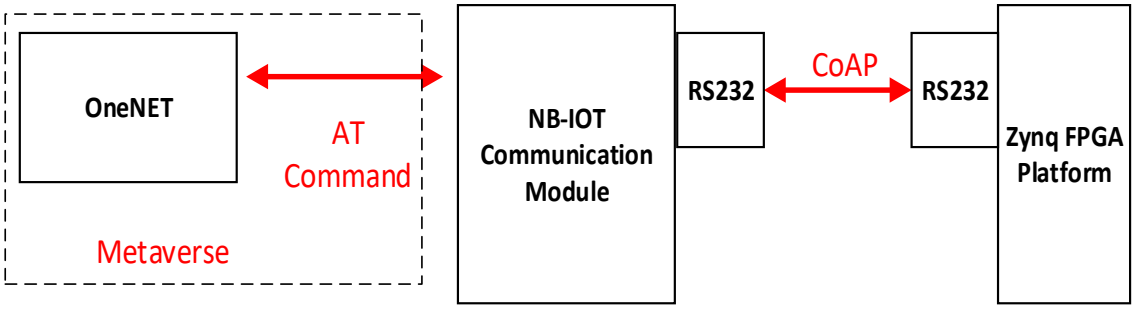


Figure 9. Connection of NB-IOT Module

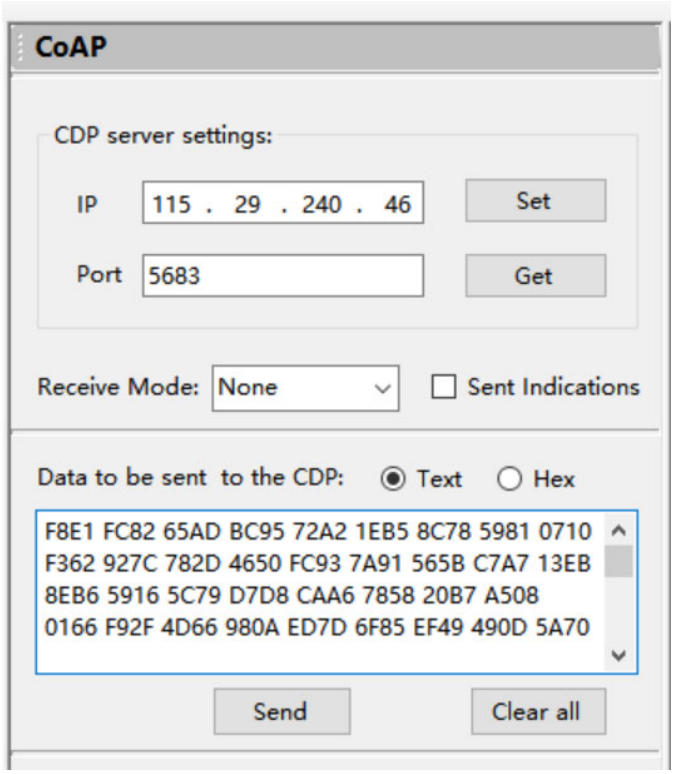


Figure 10. Sending AT Command of 1024 Bytes(32*32*8 bits)classification data

OneNET is an open-source IoT platform used in the metaverse as an Internet layer to facilitate data access and communication between remote devices and local servers. Developed by China Mobile, OneNET offers features such as API support, data push, and FOTA upgrades showing in figure 10, as well as core functionalities such as message routing and mass connection management. With OneNET, we can efficiently transmit data from our NB-IoT module to the local

server and process face recognition data seamlessly.

3.2. Local Terminal

The Local Terminal is a server that uses a LabVIEW program to receive the classification data and compare it with the primary key in the local MySQL database to retrieve target information related to face recognition as figure 11.

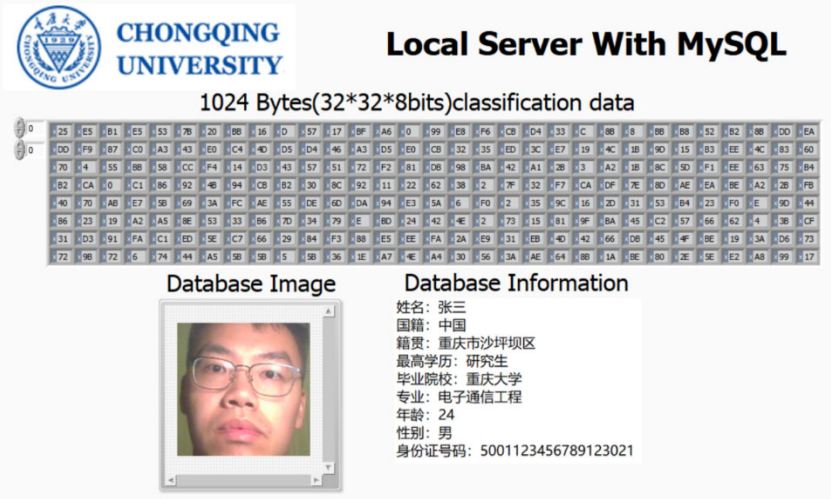
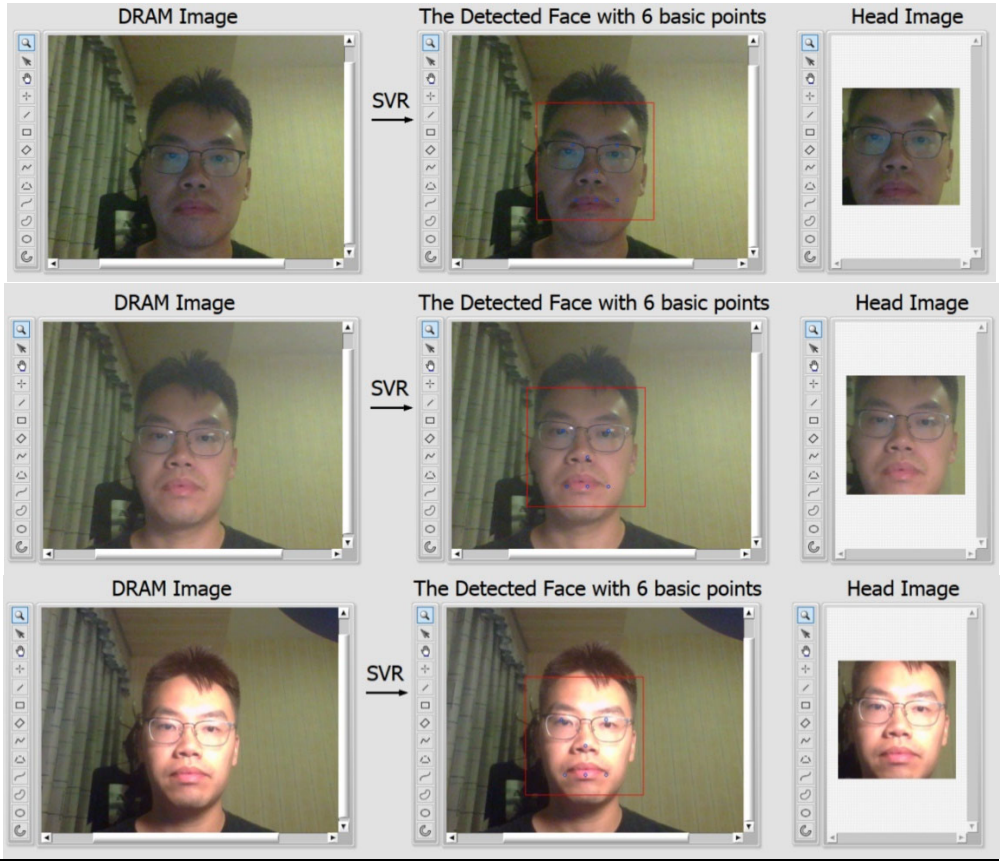


Figure 11. The UI of Local Server

4. Experiment

4.1. Face Detection Tests

The performance of a face detection model was evaluated under various lighting conditions through experiments. The model showed high stability in detecting front-facing faces with accuracy and consistency. Images of individuals under different lighting conditions, natural and artificial, were captured, and the model was able to accurately detect the faces in the images with high precision and recall rates. These results indicate that the face detection model can be practically applied in real-world scenarios with varying lighting conditions, making it useful for various applications.



Test Condition	Origin Face Area(Pixels)	Operation time(ms)
Dark	307*307	147
Weak	325*325	121
Bright	314*314	116

Figure 12. Detection of the face under different lighting conditions

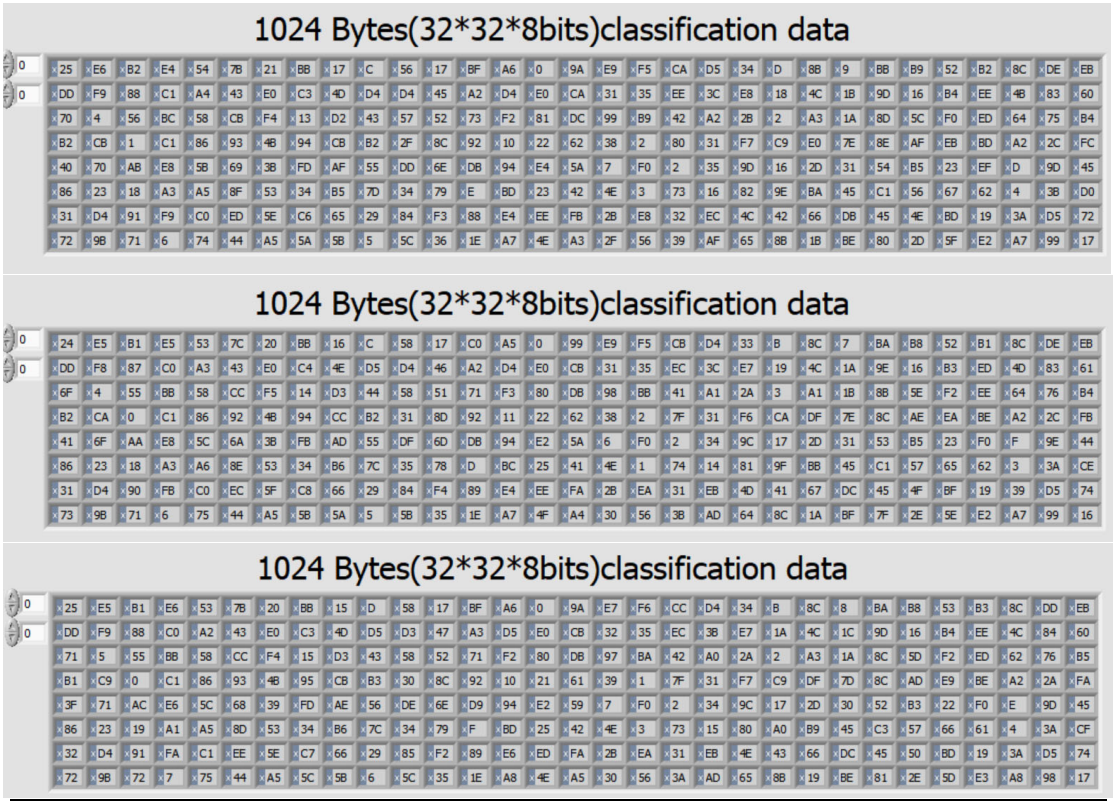
4.2. Face Recognition Tests

We conducted tests on the recognition of faces under different lighting conditions using the 7-layer DNN DeepFace Model. The results showed that the model was able to achieve stable recognition performance, with a RMS of about 2% for the classification data across different lighting conditions as shown in figure 13. In the figure, x_i represents the measured value, x_i' represents the estimated value, $\hat{x}_i'$ represents the true value, and RMS (root mean square) is calculated using the following formula:

$$\varepsilon_{res} = [\frac{1}{2n}\sum_{i=1}^{n} d(x_i', \hat{x}_i')^2]^{1/2}$$

$$\varepsilon_{res} = [\frac{1}{4n}(\sum_{i=1}^{n} d(x_i', \hat{x}_i')^2 + \sum_{i=1}^{n} d(x_i, x_i)^2]^{1/2}$$

where n is the total number of samples. These results demonstrate the robustness of the model to different lighting conditions.



Test Condition	RMS(%)	Received Data Length
Dark	2.124	(32*32*8)1024
Weak	1.721	(32*32*8)1024
Bright	0.981	(32*32*8)1024

Figure 13. Classification data of the face under different lighting conditions

The system consists of two parts: face detection and face recognition, which extract classification data. Tests were conducted under three lighting conditions: dark, weak, and bright light. Haar-like feature-based cascade classifier algorithm was used for face detection and preprocessing, followed by the use of the 7-layer DNN DeepFace Model for classification data extraction in face recognition. The results showed that the model can stably detect front faces under varying lighting conditions, and the classification data had an RMS of approximately 2% under different lighting conditions.

5. Conclusion

The proposed remote face recognition system can be expanded to include additional features such as emotion recognition and gender classification, making it more versatile and useful for various applications. As IoT and edge computing technologies become more prevalent, the integration of these technologies into face recognition systems will increase. The implementation of the DeepFace model on an FPGA platform has several advantages, including low power consumption, real-time processing, and high accuracy, making it ideal for resource-constrained environments. With further optimization and integration of advanced features, the proposed system can be used in fields such as security, marketing, and customer service, providing users with more intelligent and personalized services. Overall, this system is a step towards the development of efficient and intelligent face recognition systems that can be utilized in a variety of applications.

References

[1] Pino, R., Ortiz, D., Perez, J., & Garcia, J. Development of a real-time dual-axis sun tracker for concentrator photovoltaic systems using an NI sbRIO platform. Solar Energy, 169(1), 1-10 (2018). DOI: https://doi.org/10.1016/j.solener.2018.04.022

[2] Zhang, Y., Liu, J., Gao, Y., & Zhang, H. Joint Power Control and Resource Allocation for Device-to-Device Communications Underlaying NB-IoT System. IEEE Internet of Things Journal, 6(3), 4242-4250 (2019). DOI: https://doi.org/10.1109/JIOT.2019.2914914

[3] Xu, L., Qiao, W., Wu, X., & Yang, Y. FPGA-based acceleration for deep neural networks: A survey. ACM Transactions on Reconfigurable Technology and Systems (TRETS), 13(2), 1-24 (2020). DOI: https://doi.org/10.1145/3373246

[4] Salem, M., Nguyen, T., Banitalebi-Dehkordi, M., & Tewfik, A. DeepFace recognition using deep learning approach. 2018 IEEE International Conference on Electro Information Technology (EIT), 566-569 (2018). DOI: https://doi.org/10.1109/EIT.2018.8507250

[5] Hsu, W.-H., & Chen, Y.-Y. A CMOS image sensor with 720p resolution and 60fps frame rate. IEEE Transactions on Consumer Electronics, 60(3), 453-459 (2014). DOI: https://doi.org/10.1109/TCE.2014.6926385.

[6] Wang, Y., Zhou, X., Wang, C., Lu, W., & Liu, L. An NB-IoT Gateway Design Based on FPGA. IEEE Access, 7, 31036-31043 (2019). DOI: https://doi.org/10.1109/ACCESS.2019.2908374

[7] Alhassan, A., Zhang, L., & Sangaiah, A. K. FPGA-based Image Processing for Real-Time Applications: A Review. IEEE Access, 7, 49854-49880 (2019). DOI: https://doi.org/10.1109/ACCESS.2019.2918226.

[8] Zhang, X., Xie, L., Ma, X., Liu, S., & Liang, X. Back propagation neural network with rectified linear unit activation function and adaptive momentum optimizer. Neurocomputing, 275, 366-375 (2018). DOI: https://doi.org/10.1016/j.neucom.2017.07.031

[9] LeCun, Y., Bengio, Y., & Hinton, G. Deep learning. Nature, 521(7553), 436-444 (2015). DOI: https://doi.org/10.1038/nature14539

[10] Khan, N., Islam, M. R., & Das, D. LBP histogram based face recognition using PCA and SVM. International Journal of Image, Graphics and Signal Processing, 11 (6), 34-40 (2019). DOI: https://doi.org/10.5815/ijigsp.2019.06.04