

A Scalable Deep Learning Framework for Diabetic Retinopathy Classification with Web Integration and FPGA-Based Edge Deployment

Ram Rishik Nalukurthi¹, Ramadevi Kolisetty²

PG Student, Department of Electronics and Communication Engineering, University College of Engineering Kakinada

(A), JNTU Kakinada, Kakinada, Andhra Pradesh, India¹

Associate Professor, Department of Electronics and Communication Engineering, University College of Engineering

Kakinada (A), JNTU Kakinada, Kakinada, Andhra Pradesh, India²

Abstract: Diabetic retinopathy (DR) is a progressive microvascular complication of diabetes and a major cause of preventable vision loss. Early and accurate identification of DR severity is essential; however, conventional screening relies heavily on manual examination of retinal fundus images, making large-scale diagnosis time-consuming and dependent on specialist availability. This work presents an end-to-end framework for automated DR classification that combines deep learning, web-based accessibility, and FPGA-oriented hardware implementation. A customized residual convolutional neural network is developed to classify retinal fundus images into five severity levels: No DR, Mild, Moderate, Severe, and Proliferative DR. The model employs residual skip connections to improve feature propagation and support effective learning of pathological retinal characteristics. A preprocessing and augmentation pipeline is incorporated to standardize image representation, improve feature visibility, and address class imbalance. The proposed model achieves an overall classification accuracy of 97.2%, demonstrating strong performance in multi-class DR classification. For practical accessibility, the trained model is integrated into a Flask-based web application that enables users to upload retinal images and obtain automated predictions. To extend the framework toward edge-oriented deployment, the convolutional parameters are converted from floating-point to fixed-point integer representation and mapped to a custom FPGA accelerator on the Zynq-7000 XC7Z010-1CLG400 platform. The hardware architecture incorporates parallel 3×3 multiply-accumulate processing, bias addition, re-quantization, and ReLU activation, with parameter and data movement supported through the Zynq processing system, DDR memory, and AXI/DMA interfaces. The proposed framework therefore establishes a unified path from software-based DR classification to web-accessible screening and quantized FPGA implementation, providing a foundation for efficient edge-based medical image inference.

Keywords: Diabetic Retinopathy, Deep Learning, Residual Convolutional Neural Network, Fundus Image Classification, Multi-Class Classification, Quantization, FPGA, Zynq-7000, Edge AI, CNN Accelerator, Web-Based Screening.

I. INTRODUCTION

Diabetes mellitus is a chronic metabolic disorder that can cause long-term damage to several organ systems, including the cardiovascular system, kidneys, nerves, and eyes. Among its associated complications, diabetic retinopathy (DR) is a major cause of vision impairment and irreversible blindness. Persistent elevation of blood glucose levels can progressively damage the retinal microvasculature, resulting in pathological changes such as microaneurysms, hemorrhages, exudates, and abnormal neovascularization. Since the disease may remain asymptomatic during its initial stages, early identification is essential for reducing the risk of severe vision loss. The visual appearance of a healthy retina and a retina affected by diabetic retinopathy is illustrated in Fig. 1. The development of retinal lesions provides important clinical evidence for determining the onset and severity of the disease. The disease usually progresses through different stages, starting with No DR, where the retina does not show visible signs of damage. In Mild NPDR, small bulges in the retinal blood vessels, known as microaneurysms, may begin to appear. As the condition advances to Moderate NPDR, changes such as hemorrhages, exudates, and other vascular abnormalities become more noticeable. Severe NPDR is associated with extensive damage to the retinal blood supply and increased ischemic changes. In the most advanced stage, Proliferative Diabetic Retinopathy (PDR), new but fragile blood vessels grow abnormally and may cause bleeding and serious vision complications. Because the early stages often develop without obvious symptoms, regular screening and timely detection are important for preventing severe and irreversible vision loss.

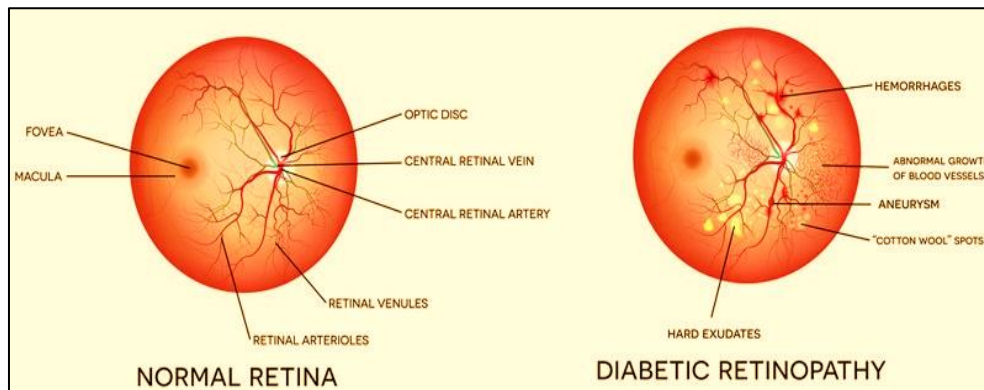


Fig. 1 Comparison of a normal retinal fundus image and a diabetic retinopathy-affected retinal fundus image.

The progression of DR across different severity levels is illustrated in Fig. 2, showing the gradual transition from a healthy retina to advanced proliferative disease.

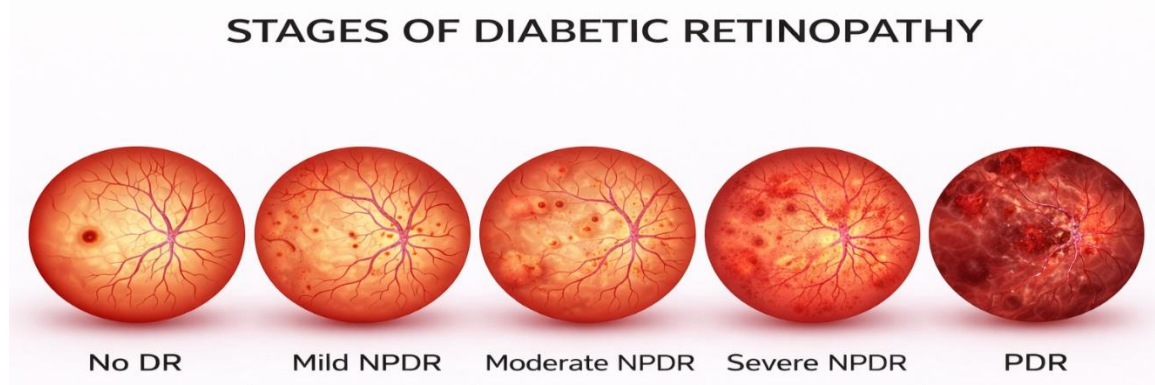


Fig. 2 Representative retinal fundus images corresponding to different stages of diabetic retinopathy.

Recent advances in artificial intelligence have improved automated DR analysis using CNNs, residual networks, hybrid models, and vision transformers [1], [2], [6], [9], [15], [26]. However, many high-performance models require substantial computational and memory resources, limiting efficient edge deployment. To address this, the proposed work combines residual CNN-based DR classification with quantized FPGA implementation using a custom 3×3 convolution engine, parallel MAC processing, re-quantization, and ReLU activation on the Zynq-7000 XC7Z010-1CLG400 platform.

The remainder of this paper is organized as follows. Section II presents the background, related work, motivation, problem statement, and research gap. Section III describes the proposed methodology, including dataset preparation, preprocessing, model development, quantization, and FPGA deployment. Section IV presents the experimental setup and software and hardware results. Finally, Sections V describes Application deployment and VI discuss the limitations, future scope, and conclusions.

II. RELATED WORK AND RESEARCH MOTIVATION

The growing need for early and reliable diabetic retinopathy (DR) screening has encouraged extensive research into automated diagnostic systems. Earlier approaches mainly depended on conventional image processing and machine learning techniques using handcrafted features such as texture, color, and morphological characteristics. Although these methods were useful for identifying retinal abnormalities, their performance was sensitive to image noise, illumination variations, and manual feature selection [5]. The emergence of deep learning significantly improved automated DR classification. CNN-based models can directly learn hierarchical features from retinal fundus images, while transfer learning and pre-processing techniques have further improved classification performance [8], [10]. However, many advanced approaches, including U-Net-based segmentation, hybrid CNN models, Vision Transformers, and ensemble networks, introduce additional computational complexity and resource requirements [1], [2], [15], [21], [26]. Recent studies have also explored IoT-based systems and hardware-oriented implementations for practical DR screening [13], [20]. While these approaches demonstrate the potential of edge and embedded deployment, efficiently translating a trained deep learning model into FPGA hardware remains challenging. In particular, floating-point computation and large

model parameters create significant memory and computational requirements. These limitations motivate the development of an integrated deep learning and FPGA-based framework that combines reliable DR classification with hardware-efficient implementation. The motivation of this work is to develop an accurate DR classification system that can move beyond conventional software-based implementation and support efficient edge deployment. The proposed approach combines deep learning with FPGA acceleration by translating the trained ResNet-based model into a hardware-compatible quantized architecture. This enables reduced memory requirements and efficient convolution processing while retaining the practical advantages of automated DR screening.

This work addresses the challenge of developing an integrated framework for five-class DR classification and FPGA-based deployment. The objective is to convert the trained ResNet-based model into a quantized hardware implementation and execute the core convolution operations using a custom accelerator on the Zynq-7000 FPGA. The proposed design employs integer-based weights and activations, MAC operations, re-quantization, and activation modules to establish an efficient bridge between software-trained neural networks and FPGA hardware, supporting low-overhead edge-oriented DR inference.

III. PROPOSED METHODOLOGY

The proposed framework integrates deep learning-based diabetic retinopathy (DR) classification with FPGA-oriented hardware implementation. The complete workflow begins with retinal fundus image preparation and five-class classification using a customized ResNet-based convolutional neural network. The trained model parameters are subsequently converted into hardware-compatible quantized representations for FPGA implementation. The proposed software–hardware framework is illustrated in Fig. 3.

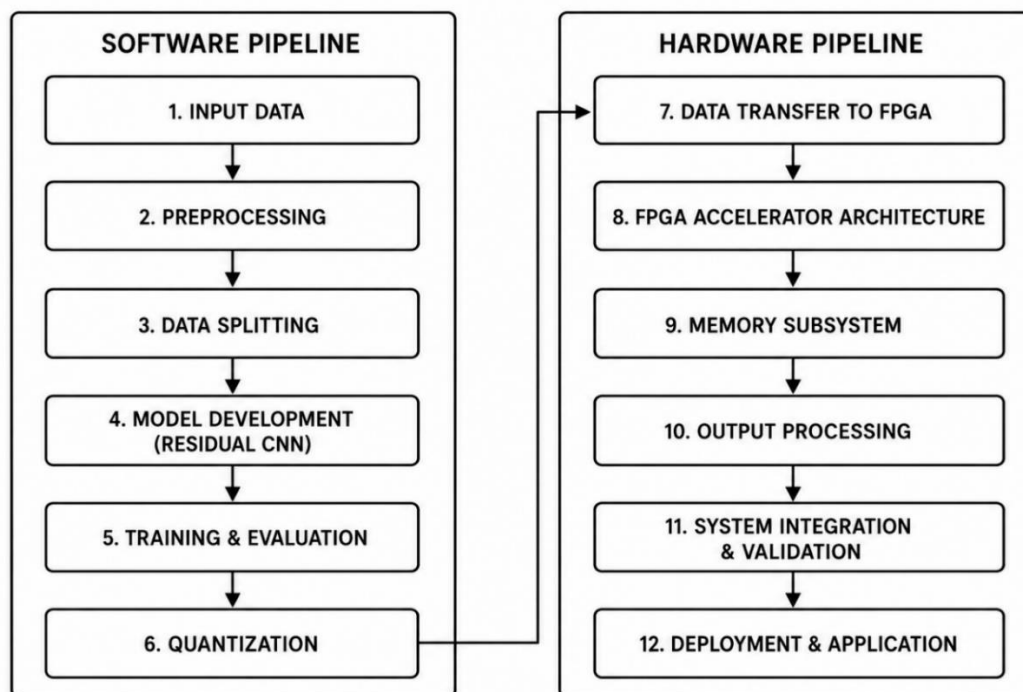


Fig. 3 Proposed end-to-end framework for DR classification and FPGA-based deployment.

As shown in Fig. 3, the retinal fundus images are acquired, preprocessed, augmented, and used to train the residual learning-based classification model. After software-level training and evaluation, the convolution parameters are converted from floating-point to fixed-point representations. The quantized data are then processed through the custom convolution accelerator implemented on the Zynq-7000 FPGA platform.

A. Dataset Preparation and Preprocessing

1) Dataset and Class Distribution:

The proposed model is trained using a publicly available Kaggle retinal fundus image dataset containing five DR severity classes: No DR, Mild, Moderate, Severe, and Proliferative DR. The original images exhibit variations in illumination,

resolution, and retinal appearance. In addition, the class distribution is imbalanced, with relatively fewer samples belonging to the advanced DR stages. To improve class representation, the dataset is processed and organized into five balanced classes. The final prepared dataset contains 5,000 images, with 1,000 images per class. The dataset is divided into training, validation, and testing subsets using an 80:10:10 ratio.

2) Channel-Based Analysis:

An exploratory analysis of the individual RGB channels was performed to examine their usefulness in highlighting retinal structures. The green channel provides improved visibility of blood vessels and pathological regions, whereas the blue channel contains comparatively lower contrast and greater noise. The channel analysis was performed only for visual understanding. The final classification model uses the complete RGB image to preserve the available color information.

3) Image Preprocessing and Labelling:

The retinal images undergo a preprocessing sequence before model training. The input images are converted to RGB format, and the dark background surrounding the retinal region is removed using threshold-based cropping. The resulting region of interest is resized to the required network input size. Gaussian smoothing and weighted enhancement are subsequently applied to improve the visibility of relevant retinal structures. The processed images are normalized to provide a consistent input range for the neural network. A representative preprocessing result is shown in Fig. 4.

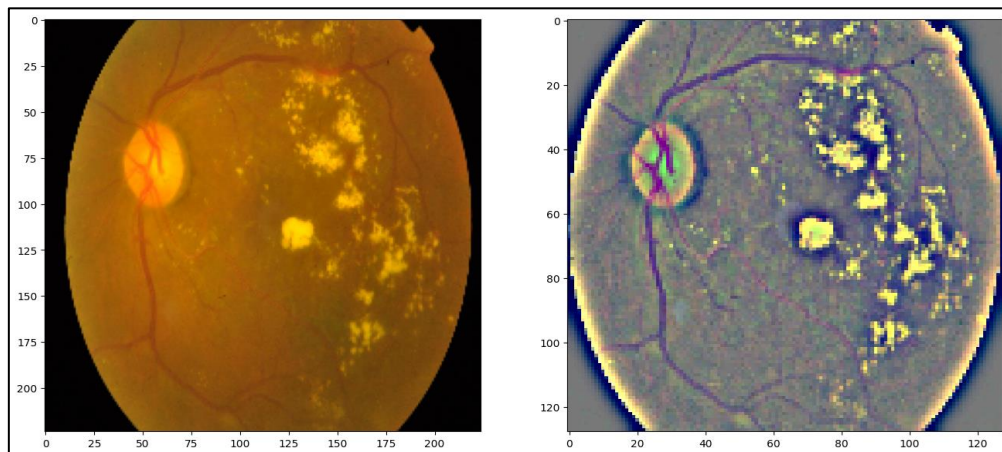


Fig. 4 Fundus image before and after preprocessing, illustrating background removal and image enhancement.

Each image is assigned a numerical label from 0 to 4, corresponding to No DR, Mild, Moderate, Severe, and Proliferative DR, respectively. The complete preprocessing operation applied to representative fundus images.

4) Data Augmentation:

Data augmentation is applied to increase sample diversity and reduce the effect of class imbalance. The augmentation process includes image rotation, horizontal and vertical flipping, scaling, and brightness variation. These transformations expose the model to variations that may occur during practical fundus image acquisition and improve generalization.

B. ResNet-Based DR Classification Model

The proposed classification network is a customized convolutional architecture inspired by residual learning. The model employs convolutional layers and residual blocks to extract progressively higher-level retinal features. A residual block is shown in Fig. 5.

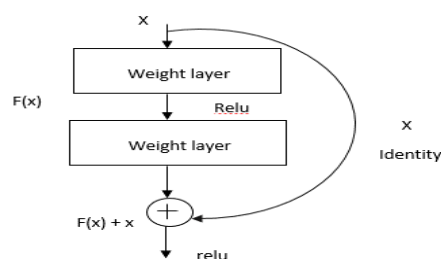


Fig. 5 Residual learning block with identity mapping.

For an input feature map $X^{(l-1)}$, the convolutional transformation at layer l can be expressed as

$$X^{(l)} = W^{(l)} * X^{(l-1)} + b^{(l)} \quad (1)$$

where $W^{(l)}$ and $b^{(l)}$ represent the convolution weights and bias, respectively, and $*$ denotes the convolution operation.

Batch normalization is applied to stabilize the feature distribution:

$$\widehat{X}^{(l)} = \gamma \frac{(X^{(l)} - \mu)}{\sqrt{\sigma^2 + \varepsilon}} + \beta \quad (2)$$

where μ and σ^2 are the batch mean and variance, while γ and β are learnable parameters.

The normalized features are passed through the ReLU activation function:

$$Y^{(l)} = \text{ReLU}(\widehat{X}^{(l)}) \quad (3)$$

The main residual learning operation combines the transformed feature representation with the input through a skip connection:

$$Y = F(X, W) + X \quad (4)$$

where $F(X, W)$ denotes the learned residual mapping. If the input and output dimensions differ, a projection is applied to the shortcut path. The architecture progressively extracts low-level and high-level retinal features through stacked residual blocks. The overall proposed network architecture is illustrated in Fig. 6.

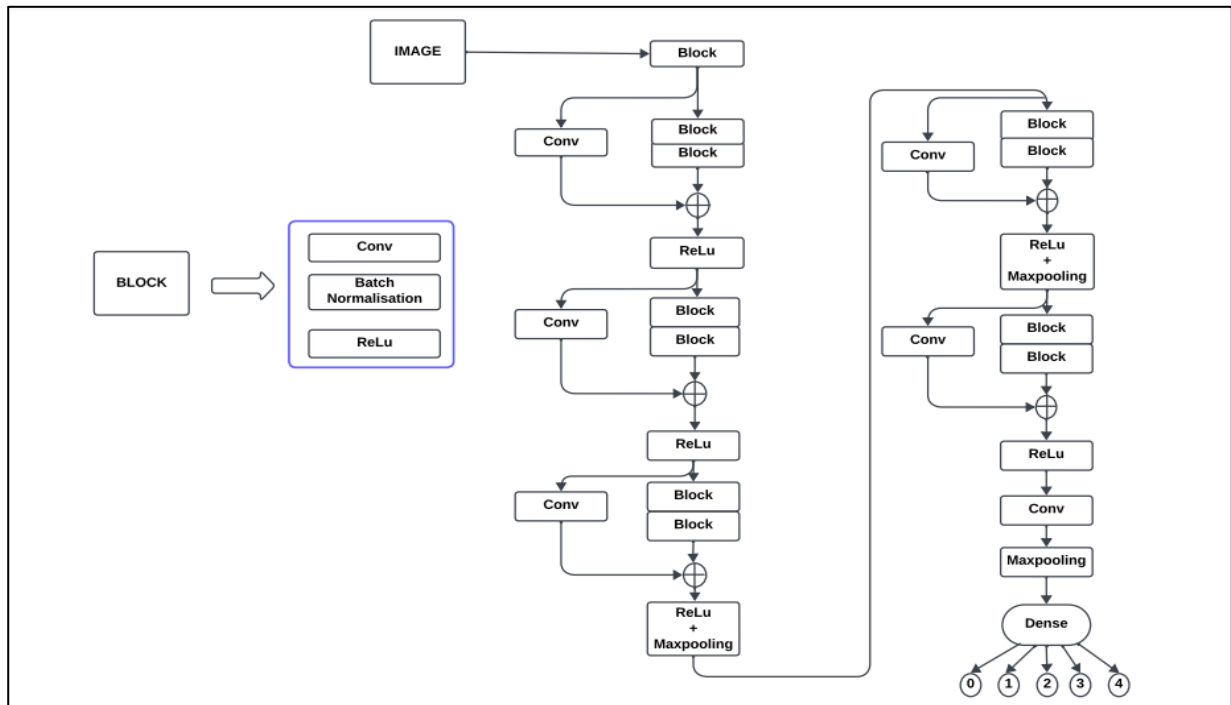


Fig. 6 Proposed ResNet-based architecture for five-class diabetic retinopathy classification.

The final feature representation is passed to the classification layers. Dropout and L2 regularization are incorporated to reduce overfitting. The output layer uses the softmax function to generate probabilities for the five DR severity classes:

$$P(y_i) = \frac{\exp(z_i)}{\sum_{j=1}^C \exp(z_j)} \quad (5)$$

where $C = 5$ is the number of output classes and z_i denotes the logit associated with class i .

C. Model Training and Software Evaluation

The model is trained in a supervised manner using the prepared retinal images and their corresponding DR severity labels. Categorical cross-entropy is used as the training loss:

$$L = -\sum_{i=1}^C y_i \log(\hat{y}_i) \quad (6)$$

where y_i is the ground-truth class representation and $\hat{y}_i$ is the predicted probability.

The Adam optimizer is employed for parameter optimization. Batch normalization and dropout are used to improve training stability and reduce overfitting. The training process is monitored using training and validation accuracy and loss curves.

The trained model is evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score. They are expressed as

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$F1 = 2 \left(\frac{Precision \times Recall}{Precision + Recall} \right) \quad (10)$$

where TP, FP, and FN denote true positives, false positives, and false negatives, respectively. These metrics, together with the confusion matrix and ROC analysis, are used to assess the software-level classification performance.

D. FPGA-Oriented Quantization and Hardware Deployment

Following software-level model development, the convolutional computation is translated into a hardware-oriented representation for deployment on the Zynq-7000 XC7Z010-1CLG400 FPGA. Since floating-point operations require greater hardware resources, the trained convolution parameters are represented using fixed-point or integer values suitable for FPGA processing.

The convolution weights, bias values, and scaling parameters are extracted from the trained model and converted into quantized values. Quantization reduces memory requirements and enables the use of integer arithmetic within the hardware accelerator. The FPGA-oriented implementation flow is shown in Fig. 7.

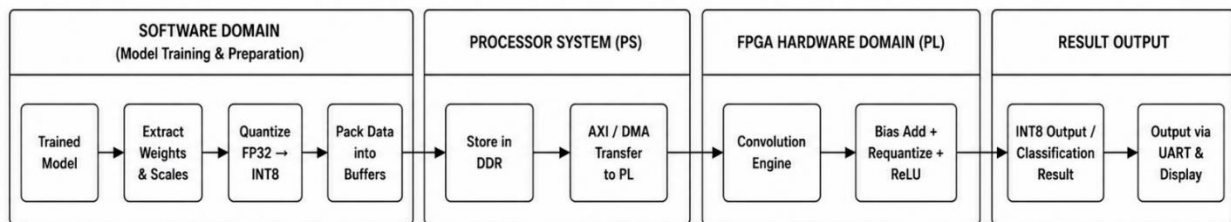


Fig. 7 FPGA deployment flow for the quantized ResNet-based convolution operation.

The hardware accelerator performs the convolution using parallel multiply–accumulate operations. Because the accumulation has a larger bit width than the input data, the result is requantized and passed through a ReLU activation. The FPGA accelerator consists of a custom 3×3 convolution engine, parallel MAC array, bias addition, requantization, and ReLU modules. The Zynq processing system provides system-level control and access to DDR memory. Quantized image and weight data are transferred through the AXI-based hardware interface to the programmable logic, where the custom convolution accelerator performs the integer computation.

IV. EXPERIMENTAL RESULTS AND HARDWARE IMPLEMENTATION ANALYSIS

This section presents the experimental configuration and performance evaluation of the proposed diabetic retinopathy (DR) classification framework. The analysis includes software-based training and classification results, random test-image prediction, comparative performance evaluation, and FPGA-oriented hardware implementation. The software

results establish the classification capability of the proposed residual CNN, whereas the hardware results demonstrate the feasibility of implementing the quantized convolutional operations on a Zynq-based FPGA platform. The complete experimental evaluation includes functional simulation, system-level hardware integration, post-implementation resource utilization, power analysis, physical implementation, and TFT-based board-level validation.

A. Experimental Setup Configuration

The proposed DR classification model was developed using the Keras deep learning framework with TensorFlow as the backend. Training and evaluation were performed on the Google Colab platform using an NVIDIA Tesla T4 GPU. The complete software pipeline included retinal image preprocessing, data augmentation, residual CNN construction, supervised training, and multi-class performance evaluation. The prepared Kaggle DR dataset contained five severity classes: No DR, Mild, Moderate, Severe, and Proliferative DR. The images were processed and divided into training, validation, and testing sets according to the methodology described in Section III. Categorical cross-entropy was used as the loss function, while the Adam optimizer was employed for parameter optimization.

B. Model Training and Performance Curves

The learning behavior of the proposed residual CNN was examined using the training and validation accuracy and loss curves. During the training process, the accuracy increased progressively, indicating that the network learned discriminative retinal features associated with different stages of DR. The validation accuracy followed a similar trend, demonstrating that the learned model retained its ability to classify previously unseen samples.

C. Classification Performance and ROC Analysis

The proposed model achieved an overall classification accuracy of 97.2% on the test data. The corresponding overall precision, recall, and F1-score were approximately 0.97, demonstrating balanced classification performance. The confusion matrix provides a detailed view of the predictions produced for the five DR classes. As shown in Fig. 8, the majority of samples were correctly assigned to their respective severity levels. The limited classification errors mainly occurred between visually similar or adjacent severity categories, where retinal abnormalities can exhibit gradual transitions.

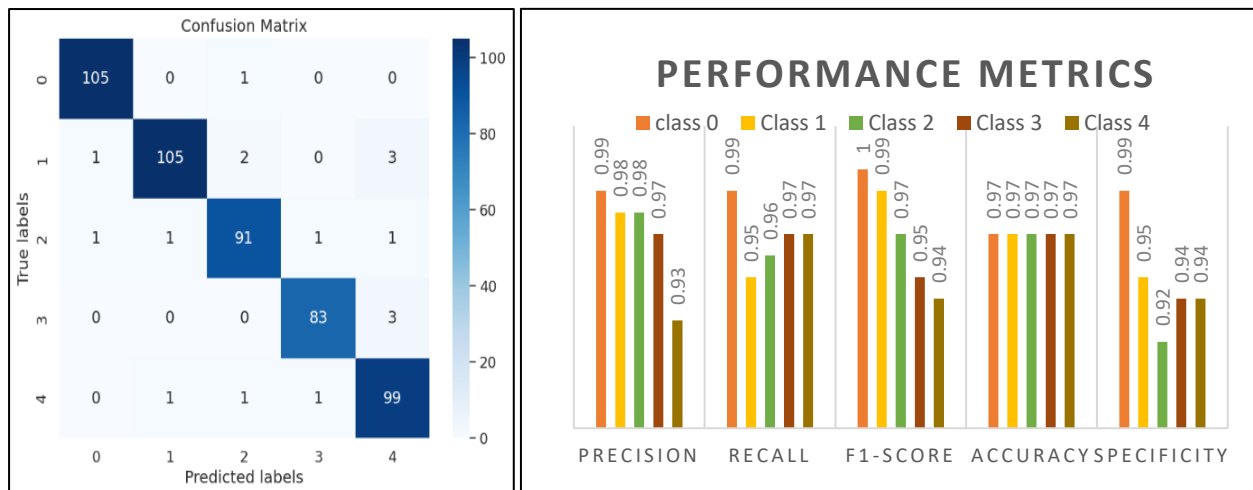


Fig. 8 Confusion matrix and Performance metrics of the proposed five-class diabetic retinopathy classification model.

The ROC curves further evaluate the separability of the five DR categories. As shown in Fig. 9, the curves indicate strong discrimination between the classes, confirming the effectiveness of the proposed residual CNN in separating different DR severity levels.

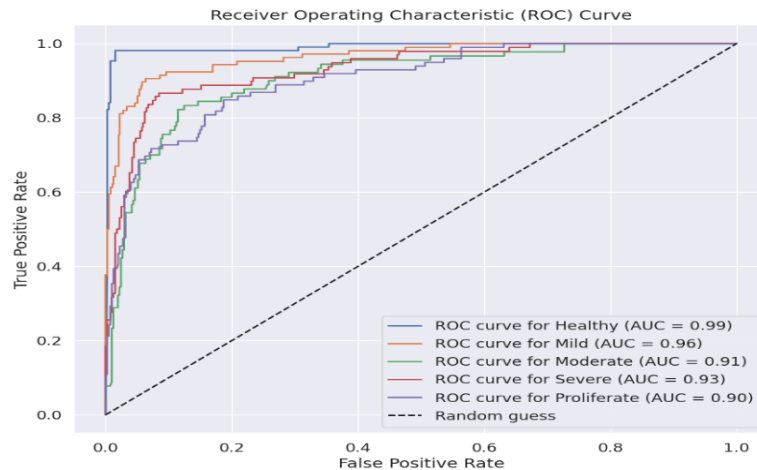


Fig. 9 Receiver operating characteristic (ROC) curves for the five diabetic retinopathy severity classes.

The combined results of the confusion matrix, classification metrics, and ROC analysis demonstrate that the proposed model maintains consistent and balanced multi-class classification performance.

D. Comparative Performance Analysis

To evaluate the proposed model with respect to the selected approaches discussed in the literature, its overall performance is compared in Table I. The proposed residual CNN achieves the highest reported accuracy among the compared methods.

TABLE I COMPARATIVE PERFORMANCE ANALYSIS

Author / Work	Dataset	Model	Target	Accuracy	Precision	Recall	F1-Score
EffNet-SVM [15]	APTOS	EfficientNet + SVM	Classification	94.2%	0.94	0.93	0.93
Ensemble Model [17]	Kaggle DR	Ensemble CNN	Classification	95.6%	0.95	0.95	0.95
Hybrid CNN [2]	EyePACS	Hybrid DL Model	Classification	96.1%	0.96	0.95	0.95
ViT Model [26]	EyePACS	Vision Transformer	Classification	96.5%	0.96	0.96	0.96
Proposed Model	Kaggle DR	Residual CNN	Classification	97.2%	0.97	0.97	0.97

The proposed model achieves an improvement over the selected comparative methods while maintaining a conventional convolution-based architecture suitable for subsequent hardware-oriented implementation. In contrast to architectures requiring complex ensemble processing or computationally intensive transformer operations, the residual CNN preserves a more practical path toward quantized FPGA deployment.

E. Random Test Image Classification

To further examine the generalization capability of the proposed model, retinal fundus images were randomly selected from the test set and passed through the trained classifier as shown in Fig. 10. The model predicted the corresponding DR severity level by assigning probabilities to the five output classes. The predictions were consistent with the visual characteristics and corresponding labels of the selected images. This evaluation demonstrates that the proposed model is capable of handling previously unseen fundus images and is not restricted to memorizing training samples.

Random Test Image Predictions

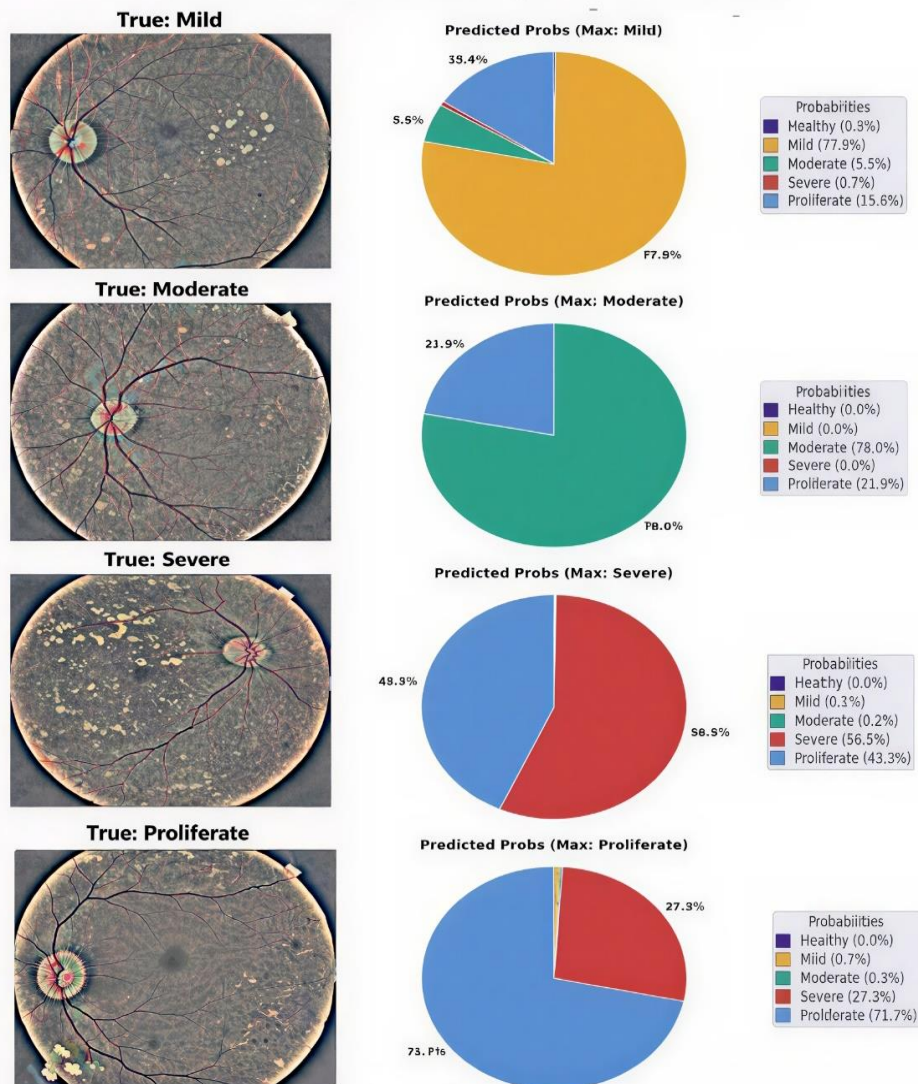


Fig. 10 Random test retinal fundus images and their predicted diabetic retinopathy severity levels.

The random-image evaluation provides an additional practical demonstration of the model's ability to identify different stages of DR from individual retinal images.

F. FPGA-Oriented Hardware Deployment

Although the complete residual CNN was trained and evaluated in the software environment, the computationally intensive convolution operation was selected as the primary target for FPGA implementation. The trained parameters were converted into an integer-compatible representation suitable for hardware execution. The custom accelerator performs the basic convolution operation using packed activations and weights, followed by accumulation, bias addition, scaling, and ReLU activation. The hardware implementation was developed using Verilog and synthesized using Xilinx Vivado. The implementation was integrated with the Zynq Processing System through AXI-based interfaces, enabling communication between the processing system and the programmable logic.

1) Functional Simulation of the Convolution Datapath:

The functionality of the custom convolution engine was initially verified through RTL simulation. The simulation waveform includes the input activations, packed weights, bias values, accumulated results, scaling parameters, and output activation. As shown in Fig. 11, the convolution datapath performs the required multiply-accumulate operation before generating the scaled and activated output. The valid_out signal indicates the availability of a valid result after the corresponding processing sequence.

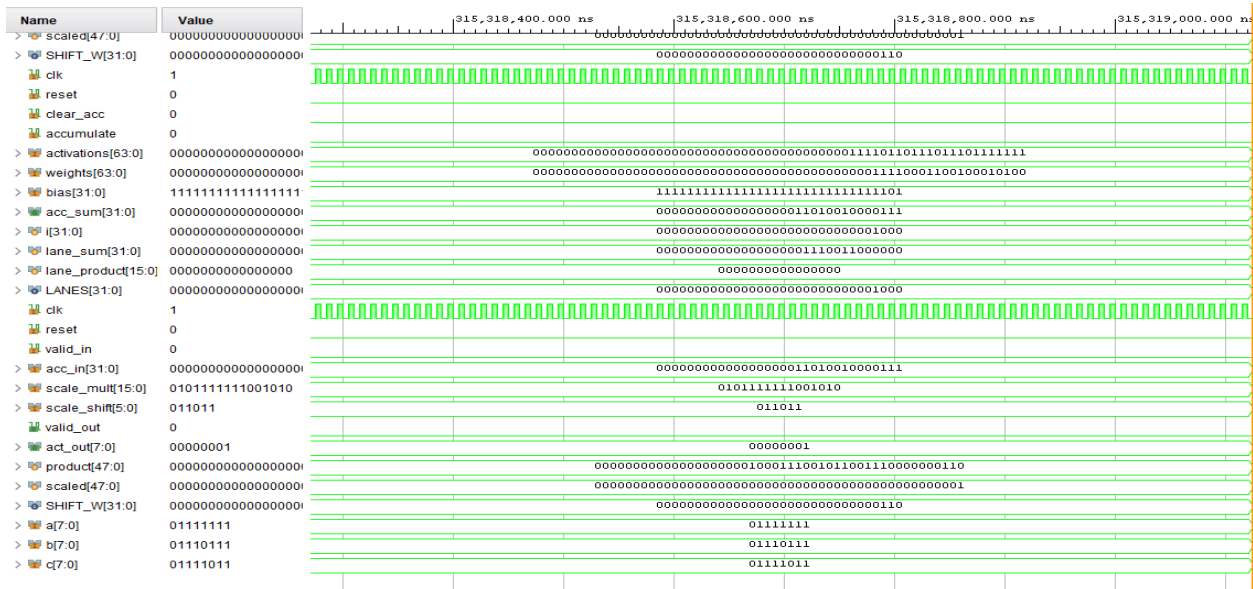


Fig. 11 RTL simulation waveform of the quantized packed 3×3 convolution datapath.

The successful simulation confirms the functional behavior of the arithmetic components before the design is transferred to the implementation stage.

2) Elaborated RTL Design:

The structural connectivity of the developed accelerator was verified using the elaborated design view. This stage provides a graphical representation of the instantiated modules and signal connections generated from the HDL description as shown in Fig. 12.

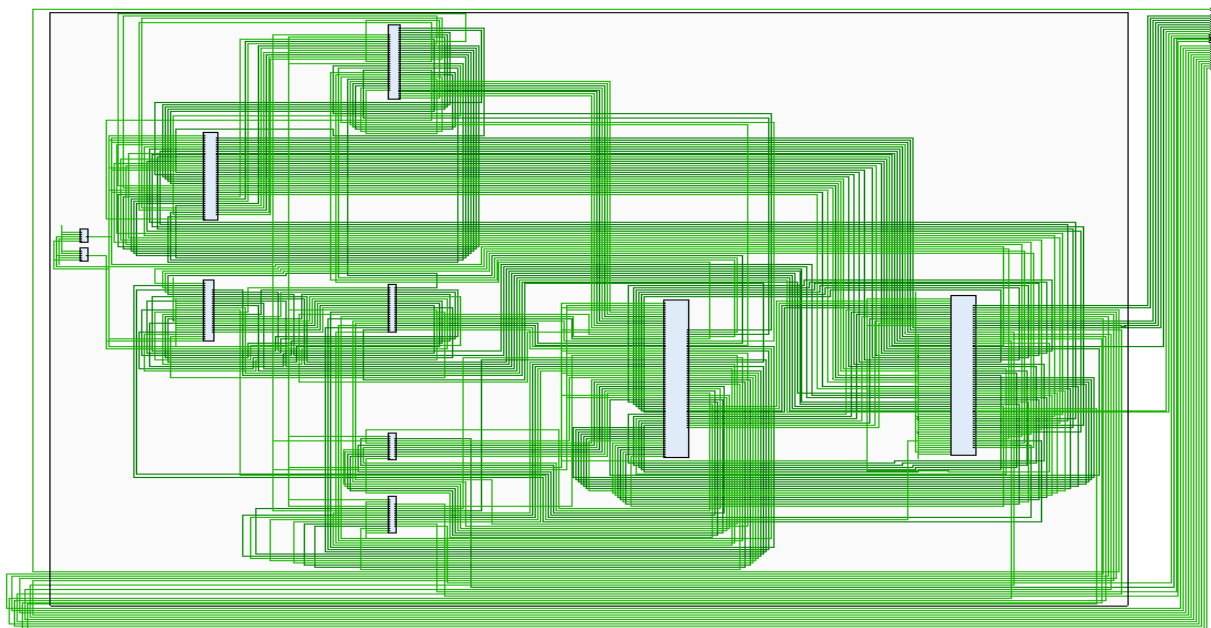


Fig. 12 Elaborated RTL design of the proposed FPGA convolution accelerator.

The elaborated design confirms that the custom modules are correctly connected before synthesis and implementation. It also provides a useful intermediate verification stage between RTL simulation and physical FPGA realization.

3) Zynq-Based Hardware System Integration:

The custom accelerator was integrated with the Zynq Processing System to establish a complete hardware platform. The system includes the processing system, AXI interconnects, AXI DMA, custom DR accelerator IP, memory interface, and

peripheral control interfaces. As shown in Fig. 13, the Processing System communicates with the programmable logic through AXI-based interconnections. The AXI DMA provides a high-speed path for transferring data between the DDR memory system and the custom accelerator.

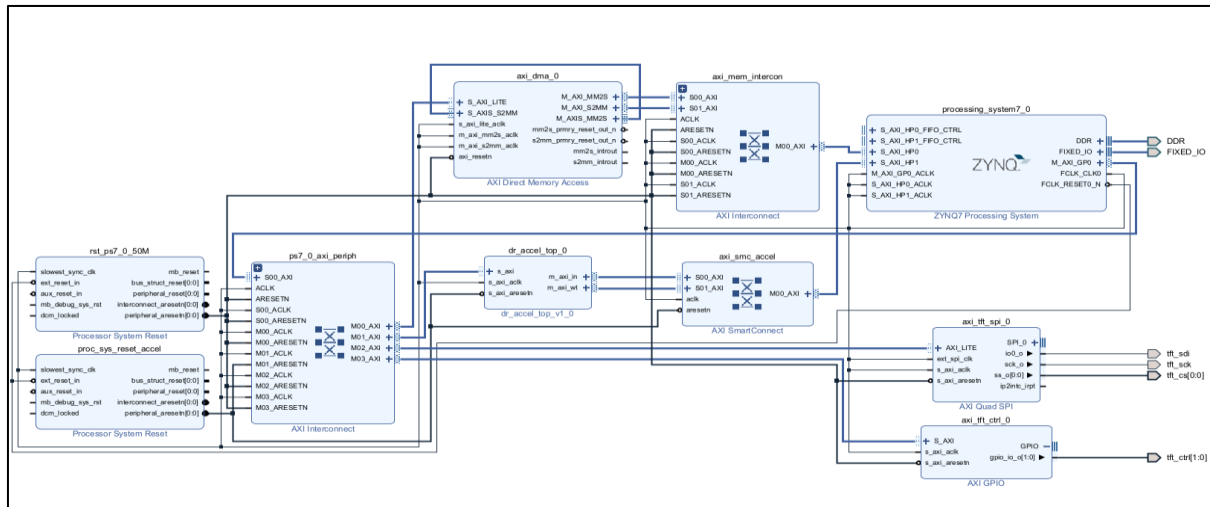


Fig. 13 Vivado block design of the Zynq-based system integrating the processing system, AXI interconnects, AXI DMA, and custom DR accelerator.

This architecture provides a scalable framework in which image data and quantized parameters can be transferred between the memory system and the programmable-logic accelerator.

4) FPGA Resource Utilization and Power Analysis:

The implementation occupies approximately 28.83% of the available LUT resources and uses only 3.33% of the BRAM resources and 6.25% of the available DSP blocks. These results indicate that substantial programmable-logic resources remain available for future extensions of the accelerator. The post-implementation power analysis was performed using the Vivado power estimation tool. The complete implemented Zynq-based design shows a total on-chip power consumption of approximately 1.667 W. The dynamic power consumption is approximately 1.549 W as shown in the Fig. 14, while the device static power is approximately 0.118 W.

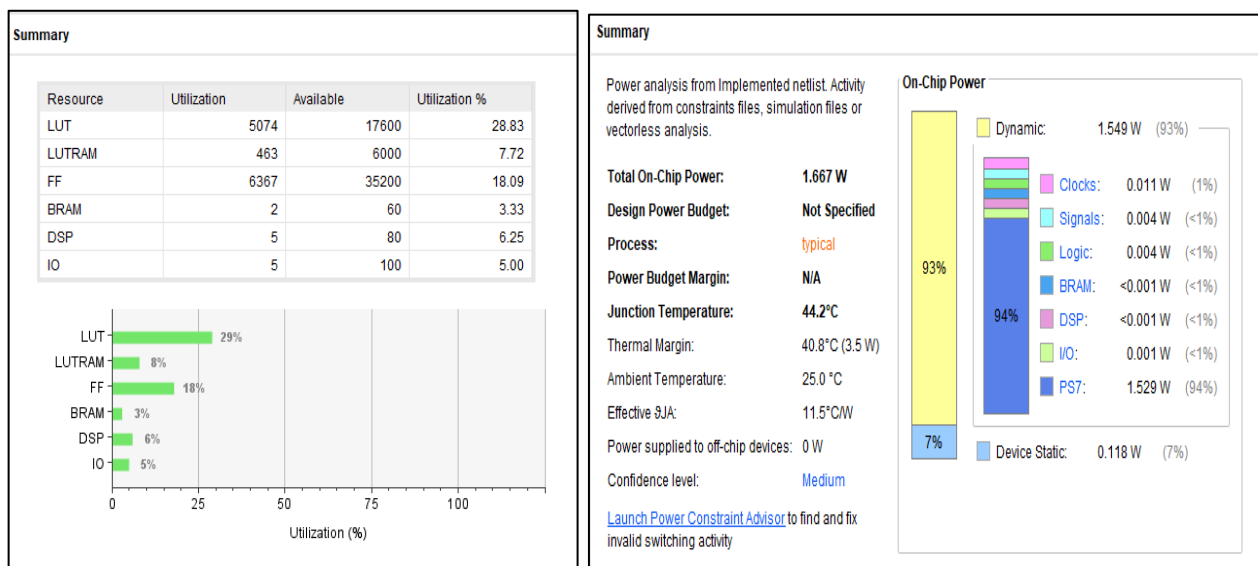


Fig. 14 Post-implementation FPGA resource utilization and Power Analysis of the proposed hardware design.

The moderate utilization of the FPGA fabric demonstrates that the present implementation can serve as a foundation for additional processing elements, parallel convolution units, or a larger portion of the complete neural network. The processing system contributes the largest portion of the overall power consumption, whereas the programmable logic

associated with the custom accelerator contributes only a comparatively small fraction. The power result should therefore be interpreted as the consumption of the complete implemented Zynq system, including the processing system and associated infrastructure, rather than the convolution accelerator alone. Nevertheless, the low resource requirement of the programmable logic supports the suitability of quantized arithmetic for energy-conscious edge implementation.

5) Physical FPGA Implementation and I/O Planning:

The physical implementation view confirms successful placement and routing of the synthesized design on the target FPGA device. The implementation result shows the distribution of logic cells and routing resources within the programmable fabric as shown in Fig. 15. Successful placement and routing demonstrate that the custom hardware design has progressed beyond behavioral simulation and has been physically implemented on the target FPGA.

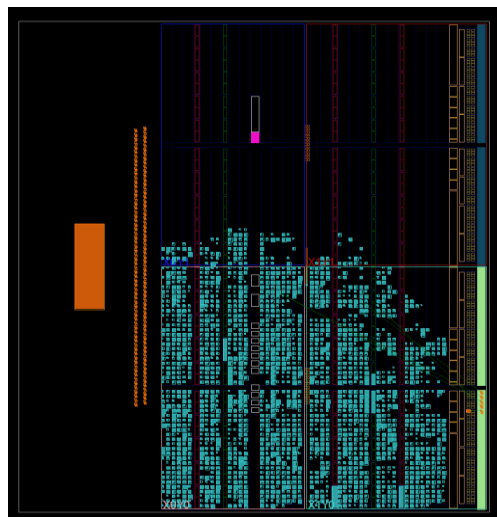


Fig. 15 Physical implementation and I/O planning of the proposed FPGA accelerator on the target Zynq device.

6) TFT-Based Board-Level Validation:

The final stage of the hardware experiment was performed on the Zynq-based FPGA development board. The implemented system was connected to a TFT display as shown in Fig. 16, providing a physical output interface for the developed hardware. The board-level result confirms successful operation of the FPGA platform and external display interface. Together with RTL simulation, system integration, implementation, resource analysis, and power evaluation, the TFT output demonstrates the complete progression from HDL development to physical hardware realization.

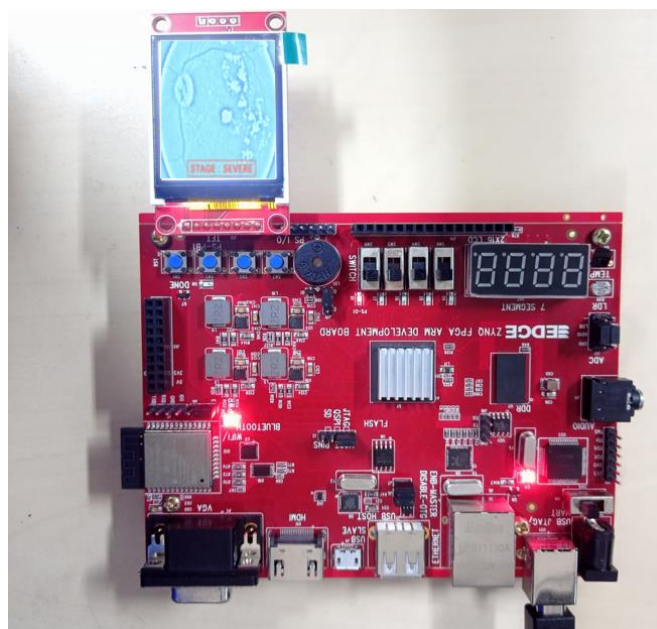


Fig. 16 TFT display output of the implemented FPGA-based diabetic retinopathy classification system.

V. WEB DEPLOYMENT APPLICATION

To demonstrate the practical usability of the proposed system, the trained diabetic retinopathy classification model was integrated into a local web application using the Flask framework. The application was developed using Python, while HTML and CSS were used to design the web interface and present the classification results. The system enables a user to upload a retinal fundus image through the browser, after which the image is processed and passed to the trained Residual CNN model for automated classification. The Python-based Flask backend handles image upload, pre-processing, model execution, and prediction processing. Once the inference is completed, the predicted DR severity level is automatically displayed through the web interface, together with the generated classification report. The application is executed as a local server, providing a simple end-to-end workflow from fundus image submission to automated disease-stage prediction. This integration demonstrates the practical applicability of the proposed model beyond the training environment and provides a user-friendly platform for automated retinal image analysis.

VI. CONCLUSION AND FUTURE SCOPE

The proposed diabetic retinopathy classification system successfully combines deep learning and practical deployment for automated five-class DR grading. The residual CNN achieved an overall classification accuracy of 97.2%, with precision, recall, and F1-score values of 0.97. The results from the confusion matrix, ROC analysis, and random-image testing show that the model can effectively distinguish between different stages of diabetic retinopathy. The model was also deployed using a Flask-based web application for practical image-based prediction. Furthermore, the FPGA implementation validated the hardware feasibility of the proposed approach through RTL simulation, Zynq system integration, resource utilization, power analysis, physical implementation, and TFT-based board-level output. In the future, the work can be extended toward complete end-to-end FPGA implementation of the entire residual CNN to achieve real-time inference with lower latency and power consumption. Further optimization using quantization, pipelining, parallel processing, and efficient memory architectures can improve hardware performance. The system can also be tested using larger and more diverse retinal datasets to evaluate its robustness across different imaging conditions. Finally, the web-based system can be integrated with cloud or edge platforms to support portable and real-time diabetic retinopathy screening applications.

REFERENCES

- [1] N. P., M. Kandasamy, R. Shanmugam, and S. K. A., "Automatic Detection and Classification of Diabetic Retinopathy Using Modified UNET," in *Proc. 2023 3rd Int. Conf. Artificial Intelligence and Smart Energy (ICAIS)*, Coimbatore, India, 2023, pp. 1468–1471, doi: 10.1109/ICAIS56108.2023.10073897.
- [2] A. Jabbar, H. B. Liaqat, A. Akram, M. U. Sana, I. Domínguez Azpíroz, and I. De La Torre Diez, "A Lesion-Based Diabetic Retinopathy Detection Through Hybrid Deep Learning Model," *IEEE Access*, vol. 12, pp. 40019–40036, 2024, doi: 10.1109/ACCESS.2024.3373467.
- [3] M. A. K. Raiaan, K. Fatema, I. U. Khan, S. Azam, M. R. U. Rashid, and M. S. H. Mukta, "A Lightweight Robust Deep Learning Model Gained High Accuracy in Classifying a Wide Range of Diabetic Retinopathy Images," *IEEE Access*, vol. 11, pp. 42361–42388, 2023, doi: 10.1109/ACCESS.2023.3272228.
- [4] Y. Yang, Z. Cai, S. Qiu, and P. Xu, "A Novel Transformer Model With Multiple Instance Learning for Diabetic Retinopathy Classification," *IEEE Access*, vol. 12, pp. 6768–6776, 2024, doi: 10.1109/ACCESS.2024.3351473.
- [5] H. Naz, T. Saba, F. S. Alamri, A. S. Almasoud, and A. Rehman, "An Improved Robust Fuzzy Local Information K-Means Clustering Algorithm for Diabetic Retinopathy Detection," *IEEE Access*, vol. 12, pp. 78611–78623, 2024, doi: 10.1109/ACCESS.2024.3392032.
- [6] S. K. Henge, N. R. Viraati, M. Alhussein, A. S. Kushwaha, K. Aurangzeb, and R. Singh, "Detection of Diabetic Retinopathy Using a Multi-Decision Inception-ResNet-Blended Hybrid Model," *IEEE Access*, vol. 13, pp. 8988–9005, 2025.
- [7] F. T. and A. M. K., "Diabetic Retinopathy Classification Based on Segmented Retinal Vasculature of Fundus Images Using Attention U-NET," in *Proc. 2022 IEEE 19th India Council Int. Conf. (INDICON)*, 2022.
- [8] A. J. Kadhim, H. Seyedarabi, R. Afrouzian, and F. S. Hasan, "Diabetic Retinopathy Classification Using Hybrid Color-Based CLAHE and Blood Vessel in Deep Convolution Neural Network," *IEEE Access*, vol. 12, pp. 194750–194761, 2024.
- [9] A. P. Budaragade, S. B. Ankali, V. U. Patil, D. Kadam, U. Bubane, and U. N. Salunkhe, "Diabetic Retinopathy Classification Using ResNet 34-U Net Architecture," in *Proc. 2025 Int. Conf. Smart & Sustainable Technology (INCSST)*, 2025, pp. 1–6.
- [10] S. Zhu, C. Xiong, Q. Zhong, and Y. Yao, "Diabetic Retinopathy Classification With Deep Learning via Fundus Images: A Short Survey," *IEEE Access*, vol. 12, pp. 20540–20558, 2024.

- [11] U. Tripathi, M. Singh, A. Mohan, and S. K. Tripathy, "Early Diabetic Retinopathy Severity Level Detection Using EffUNets," in *Proc. 2024 IEEE 21st India Council Int. Conf. (INDICON)*, Kharagpur, India, 2024, pp. 1–6, doi: 10.1109/INDICON63790.2024.10958373.
- [12] M. Tariq, V. Palade, and Y. Ma, "Effective Diabetic Retinopathy Classification With Siamese Neural Network: A Strategy for Small Dataset Challenges," *IEEE Access*, vol. 12, pp. 182814–182827, 2024.
- [13] K. P. Senthilkumar, A. P. J. Krishna, S. Tamilselvi, M. G., D. Sathish, and M. Devi, "Efficient VLSI Design of a Lightweight CNN for On-Chip Retinal Disease Detection Using Fundus Images," in *Proc. 2025 Int. Conf. Electronics, Computing, Communication, Networking and Automation Technologies (ICEC2NT)*, 2025, pp. 1–6.
- [14] <https://www.kaggle.com>.
- [15] K. V. Naveen, B. N. Anoop, K. S. Siju, M. K. Kar, and V. Venugopal, "EffNet-SVM: A Hybrid Model for Diabetic Retinopathy Classification Using Retinal Fundus Images," *IEEE Access*, vol. 13, pp. 79793–79804, 2025.
- [16] M. Vadduri and P. Kuppusamy, "EHO-Q-LGAN: An EHO-Based Q-Learning GAN for the Timely Diagnosis of Diabetic Retinopathy," *IEEE Access*, vol. 13, pp. 118959–118974, 2025.
- [17] K. Nazir, J. Kim, and Y.-C. Byun, "Enhancing Early-Stage Diabetic Retinopathy Detection Using a Weighted Ensemble of Deep Neural Networks," *IEEE Access*, vol. 12, pp. 113565–113579, 2024.
- [18] M. Elsharkawy et al., "Fused-AETNet: A Variational Transformer-Based Framework for Diabetic Retinopathy Classification Using OCT Biomarkers," *IEEE Access*, vol. 13, pp. 163120–163133, 2025.
- [19] A. Zedadra, O. Zedadra, M. Y. Salah-Salah, and A. Guerrieri, "Graph-Aware Multimodal Deep Learning for Classification of Diabetic Retinopathy Images," *IEEE Access*, vol. 13, pp. 74799–74810, 2025.
- [20] T. Palaniswamy and M. Vellingiri, "Internet of Things and Deep Learning Enabled Diabetic Retinopathy Diagnosis Using Retinal Fundus Images," *IEEE Access*, vol. 11, pp. 27590–27601, 2023.
- [21] X. Gou, Y. Wang, and W. Li, "LSD-HybridViT: A Hybrid Vision Transformer With Lightweight Mixed-Domain Attention and Frequency Multi-Scale Dilated Convolution Feature Fusion for Diabetic Retinopathy Grading," *IEEE Access*, vol. 13, 2025.
- [22] A. M. Mutawa, G. R. Hemalakshmi, N. B. Prakash, and M. Murugappan, "Randomization-Driven Hybrid Deep Learning for Diabetic Retinopathy Detection," *IEEE Access*, vol. 13, pp. 38901–38913, 2025, doi: 10.1109/ACCESS.2025.3546359.
- [23] K. Aurangzeb, R. S. Alharthi, S. I. Haider, and M. Alhussein, "Systematic Development of AI-Enabled Diagnostic Systems for Glaucoma and Diabetic Retinopathy," *IEEE Access*, vol. 11, pp. 105069–105081, 2023.
- [24] M. Kovac, P. Macsik, V. Kurilova, O. Straka, Z. Fellner, and J. Pavlovicova, "Unconditional GAN-Based Generation of Stage-Specific Fundus Images for Augmenting Diabetic Retinopathy Screening," *IEEE Access*, vol. 14, pp. 47800–47815, 2026.
- [25] R. Sairaghavakrishnan, M. K. Srilekha, K. R. Sudhaharen, and S. Karun, "Vision Transformer and Flask-Based Web System for Diabetic Retinopathy Detection," in *Proc. 2025 4th Int. Conf. Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*, 2025, pp. 850–854.
- [26] W. Nazih, A. O. Aseeri, O. Y. Atallah, and S. El-Sappagh, "Vision Transformer Model for Predicting the Severity of Diabetic Retinopathy in Fundus Photography-Based Retina Images," *IEEE Access*, 2023.
- [27] G. P. Reddy, P. D. Javali, R. Ramyashree, S. Raghavendra, C. B. Chandrakala, and P. S. Venugopala, "A Hybrid Deep Learning Approach With Explainable AI for Diabetic Retinopathy Classification," *IEEE Access*, vol. 14, pp. 28819–28839, 2026.
- [28] B. Farooq, M. Asim, A. Humayun, and S. A. Qazi, "FPGA Based Diabetic Retinopathy Classification Using Deep Neural Network Accelerator," *ResearchGate*, May 2022.