International Journal of Web of Multidisciplinary Studies
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
website: https://ijwos.com
Volume 3 Issue 8 , August 2026
E-ISSN : 3049-2424
DOI: https://doi.org/10.71366/ijwos

AI Driven Diabetic Retinopathy Detection and Risk Assessment System Using Convolutional Neural Networks

Varun K S[1], Shwetha H C[2]
[1] Assistant Professor, Department of Master of Computer Applications , G M University, Davanagere
[2] Student, Department of Master of Computer Applications , G M University, Davanagere.

Article Info

Article History:

Published: 11 August 2026

Publication Issue:
Volume 3, Issue 8
August-2026

Page Number:
133-145

Corresponding Author:
Shwetha H C

Abstract:

Diabetic retinopathy is a major healthcare concern that causes blindness, which makes it essential to detect the eye condition. The report discusses Retina Sense, a system to diagnose diabetic retinopathy and assess risk based on artificial intelligence techniques to create an Efficient Net B0 convolutional neural network and a clinical risk assessment module. The software was created with the integration of React Tan Stack for developing the front end, Node Js Express for the back end, and PostgreSQL database to automate retinopathy detection and risk assessment as well as treatment planning. The classifier was trained using the APTOS 2019 Blindness Detection dataset employing transfer learning and evaluated using its performance indicators.

Keywords: Diabetic Retinopathy, Convolutional Neural Networks, Deep Learning, Transfer Learning, EfficientNet-B0, Medical Image Classification, Risk Assessment, Explainable Artificial Intelligence, Fundus Imaging, Clinical Decision Support

1. INTRODUCTION

Diabetes mellitus affects millions of people around the world and high blood sugar over a time can hurt the small blood vessels in the eye. This can lead to Diabetic Retinopathy a cause of vision loss that can be prevented. Diabetic Retinopathy starts with changes and can get worse if not treated. This makes checking the back of the eye very important to stop losing eyesight [2] [6].

Looking at the back of the eye by experts is the way to find Diabetic Retinopathy.. This process takes time and different doctors might see things differently. Also there are not experts, especially in poorer areas [2] [6] [9]. Because of these problems people have been trying to create computer systems that can check for Diabetic Retinopathy automatically.

Deep learning systems, like Convolutional Neural Networks have done well in finding Diabetic Retinopathy. These networks can learn to spot signs of disease on their own without needing people to teach them what to look for [1] [4] [5] [7].. Most of these systems just say if someone has the disease or not. They don't look at things that might affect a person's chance of getting worse or offer advice that doctors can use [2] [3] [6] [8] [9].

In order to address these problems, the current study proposes a system called Retina Sense. This is an assessment tool used to detect and evaluate Diabetic Retinopathy and assess its risk factor on an individual basis. The proposed model employs an architecture called EfficientNet-B0 for predicting the severity level of Diabetic Retinopathy and also includes the rule-based component responsible for assigning risk levels and recommendations. The overall system consists of a web application with three modules.

2. LITERATURE REVIEW

A. Custom CNN and Explainable Architectures

A variety of custom CNNs trained end-to-end, without using a frozen backbone pre-trained network, have exhibited good multi-class classification performance on the APTOS 2019 data set by using extensive class balancing augmentation and Grad-CAM visualizations [1]. The methods like Grad-CAM and other saliency techniques increase interpretability by marking the image regions which contribute towards the classification decision [1], [3]; however, these visualizations are still inside the training network and not an external one; moreover, good classification accuracies are obtained by using heavily augmented data sets in this research track [1].

B. Transfer Learning Approaches

Using transfer learning with backbones trained on ImageNet data such as DenseNet-169, ResNet-50, Inception-v3 and Inception-v4, the accuracy in retinopathy classification is greatly enhanced.

It will need labeled data for us to achieve better accuracy.Depending on our backbones, we may get different results.For example, DenseNet-169 with our head achieved about 90% accuracy in Kaggle and APTOS datasets.

Other approaches using feature extraction and capsule networks achieved over 96% accuracy on smaller data sets coming from one source only.

It is evident that fine tuning of the backbone plays an important role in achieving results, and it is not just about which backbone you select, but how well you fine tune it to the fundus images.

Fine tuning plays a significant role in the effectiveness of pretrained features on fundus images.

C. Compact and Task-Specific Architectures

Several papers have demonstrated that CNN models trained from scratch for DR using DR-specific datasets are capable of competing with and even outperforming their transfer learning counterparts, while being very computationally efficient. As an illustration, RSG-Net, consisting of three specifically-designed blocks, provides test accuracy of over 99% within 25 milliseconds on Messidor-1; however, its flawless performance on the training set is a clear indication of overfitting on a relatively small dataset. [7].

D. Clinical Validation and Deployment

Large-scale clinical validation studies present a more conservative but arguably more representative picture of achievable performance: the FDA-cleared IDx-DR system reports 87.2% sensitivity and 90.7% specificity in prospective use [2], Google's ARDA system reports sensitivity and specificity above 90% across multiethnic cohorts exceeding 100,000 images [6], [8], and EyeArt v2.1 sustains comparable performance across more than 800,000 patients with results reported as unaffected by ethnicity, camera type, or gender [6]. These deployment-focused studies consistently emphasize that reported accuracy narrows as validation cohorts grow larger and more heterogeneous, and they identify workflow integration, responsibility attribution for misdiagnosis, and the opacity of black-box predictions as unresolved barriers to broader clinical adoption [2], [6], [8].

E. Risk Stratification and System-Level Gaps

In the literature studied, the risk of DR is generally presented in a single category determined by the imaging classifier without the incorporation of non-imaging factors, such as glucose levels, duration of diabetes, and blood pressure, in a composite risk score [9]. Platforms that provide a comprehensive solution, integrating both classification and patient management and risk assessment and recommendations are also quite scarce, since the majority of the published methods stop at the stage of off-line model evaluation [2], [6], [9]. This is one of the reasons for adopting the risk fusion and system integration strategy used in the current study.

Comparative Analysis

Table I. Comparative Analysis of Reviewed Studies

Ref.	Method	Dataset	Performance
[1]	Custom CNN + Grad-CAM	APTOS (Aug.)	**95.3% Accuracy**
[4]	DenseNet-169	Kaggle-2015 + APTOS	**90.3% Accuracy**
[5]	RetNet-10 / Capsule Net	Mixed Dataset	**98.7% Accuracy**
[6]	Inception-v3 (ARDA) / Eye Art	Eye PACS	**91.3% Sensitivity**
[7]	RSG-Net	Messidor-1	**99.4% Accuracy**
[2]	ID x-DR	Clinical Dataset	**87.2% Sensitivity**
Proposed	EfficientNet-B0	APTOS 2019	**9.3% Test Accuracy**

Ref.	Method	Performance	Strength	Limitation
[1]	Custom CNN + Grad-CAM	95.3% Acc.	Explainable AI	High complexity
[5]	RetNet-10	98.7% Acc.	High accuracy	Limited validation
[6]	Inception-v3 / Eye Art	91.3% Sens.	Clinical validation	Workflow gap
[7]	RSG-Net	99.4% Acc.	Fast inference	Overfitting risk
[2]	ID x-DR	87.2% Sens.	FDA approved	Black-box model
Proposed	EfficientNet-B0	9.3% Test Acc.	CDSS + Risk Index + XAI	Needs optimization

Above Table shows, the highest reported accuracies in the literature are generally obtained on small, single-source datasets under heavy augmentation [5], [7], while accuracy on large, heterogeneous validation cohorts is consistently lower, though still clinically useful [2], [6]. The proposed system's classifier, evaluated here without adjustment, falls below every reviewed comparator, a gap that Section VII traces to a specific and diagnosable training-strategy cause rather than to an inherent property of the dataset.

3.RESEARCH GAP

The literature highlights three gaps, which are intertwined with each other. First, the highest accuracy models are usually trained and evaluated on only one dataset, where there is a lot of class balancing, which raises questions regarding the applicability of these models to other datasets [1], [5], [7]. Second, model interpretability usually consists of visualizing gradients within a model, and not a separate risk score, taking into account both imaging and non-imaging clinical data [1], [3], [9]. Finally, and perhaps most critically, in terms of trying to implement these findings, negative or weak results are almost never published in full, hiding common pitfalls that will be encountered [4], [9].

In order to deal with these issues directly, the current study combines the problem of classifying DR with the development of a risk assessment component that does not depend on the classifier's performance. It does not visualize the gradients but offers an easily interpretable result. The results of the experiment are presented in accordance with a certain clinical aim, even if this aim is not met. Most importantly, it presents a controlled experiment in which the cause of the classifier's failure can be revealed.

4. PROPOSED METHODOLOGY

The proposed platform integrates dataset preparation, preprocessing and augmentation, an EfficientNet-B0 classification model, a rule-based risk-assessment module, and a full-stack web platform with secure access and persistent storage.

A. Overall Workflow

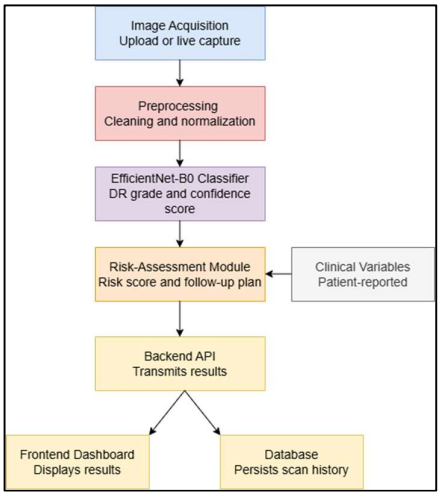

Fig1:System Workflow

The fundus images can be taken either by uploading or recording in live, processed, and normalized and fed into the EfficientNet-B0 model, which gives a prediction on the grade of DR along with a level of confidence for the prediction. The prediction grade along with the clinical information given by the patient is fed into the risk assessment model, which generates a risk score between 0 and 100 and a recommendation follow-up structure.

B. Dataset Preparation

APTOS 2019 Blindness Detection data set is used for training the model and contains 3662 images of fundus of the eye. Images are of size 224x224 pixels. Images are classified into five categories depending upon the severity of the disease. There is one limitation associated with the data set. There are numerous images available in No DR class, and 49.3% of the total images belong to No DR category. The data set of APTOS 2019 Blindness Detection is segregated into training, validation and test subsets with the ratio of 80%, 10% and 10%, respectively. The distribution of the

images among various classes is maintained while segregating the data set into subsets for blindness detection based on APTOS 2019 data set.

C. Preprocessing and Augmentation

All images will be decoded as the first step before formatting them. Normalization is defined as the step through which we modify the values of our images to be within the range of -1.0 to 1.0. In the training stage, we use augmentation techniques that introduce complexities into our training images through the following: cropping images, rotation, flipping images, modifying their brightness and colors and, lastly, blurring. All images used in the APTOS 2019 Blindness Detection dataset for validation and testing are just normalized.

D. EfficientNet-B0 Classification Model

For training the classifier, the backbone EfficientNet-B0 [10] was used, which is pre-trained on ImageNet weights using global average pooling instead of a classification head. The classifier head includes Dense(512) layers that use Swish activation and L2 regularization. There is also a Dense(256) layer with Swish activation and dropout (0.30, 0.24, 0.15), followed by batch normalization. Training is performed in two stages: in the first one, the backbone is frozen, and the training is performed only for the head. Unfreezing and training the top 30% of the layers of the backbone with a reduced learning rate happens at the second stage only. The training procedure is conducted using the AdamW optimizer and categorical focal cross-entropy loss ($\alpha = 0.25$, $\gamma = 2.0$).

E. Mathematical Formulation

Pixel normalization maps each intensity $x \in [0,255]$ to the network's input range:

$$x' = \frac{x}{127.5} - 1.0$$

The convolution and softmax operations underlying the backbone and output layer are given by

$$S(i,j) = \sum_m \sum_n I\,(i+m, j+n)\,K(m,n)$$

$$\sigma(z)_k = \frac{e^{z_k}}{\sum_{c=1}^{K} e^{z_c}}$$

and the training loss, categorical focal cross-entropy, is

$$\mathcal{L}_{FL} = -\sum_{c=1}^{K} \alpha_c\,(1-\hat{y}_c)^{\gamma}\,y_c \log(\hat{y}_c).$$

Evaluation uses standard classification metrics — accuracy, precision, recall, F1-score, and quadratic-weighted Cohen's kappa —

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}, \quad F1 = \frac{2\,PR}{P+R}$$

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j}\,O_{i,j}}{\sum_{i,j} w_{i,j}\,E_{i,j}}, \quad w_{i,j} = \frac{(i-j)^2}{(K-1)^2}.$$

F. Risk-Assessment and Recommendation Module

Besides CNN, other deterministic risk scoring method using regulatory factors involves DR grading and variables such as HbA1c/Fast blood glucose, duration of diabetes, blood pressure, BMI, smoking status, family history and

previous eye treatments taken by the patient. A score between 0 and 100 is assigned to each patient using the above mentioned data. The above scores relate to one of four risk categories (low, moderate, high, very high) and advice on the measures to be undertaken by the doctor.

G. Web Platform

The frontend of the application has been developed in React 19, TanStack router, and TypeScript for image upload, real-time capture, and visualization of results. On the other hand, the backend of the application includes Node.js/Express REST API for authentication using JWT and bcrypt, image upload using multer and Python inference service. The database to be used is PostgreSQL and Supabase to store user accounts, scan history, and risk scores from model predictions.

5. SYSTEM ARCHITECTURE

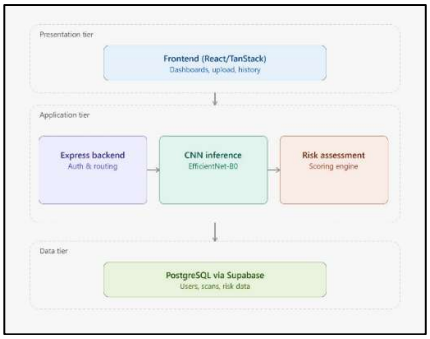

Fig2: System Architecture

The system follows a three-tier architecture — a presentation tier, an application tier incorporating the CNN inference and risk-assessment services, and a data tier — deployed within a web-accessible environment.

A. Frontend

The React/TanStack frontend renders patient, doctor, and administrator dashboards; an image-upload and live-capture interface; a risk-assessment page; and a scan-history and reporting view, communicating with the backend through authenticated HTTP requests.

B. Backend

The Express.js backend authenticates requests, routes uploaded images to the inference service, assembles the combined classification-and-risk response, and manages persistence through its controller layer for patients, doctors, scans, reports, and appointments.

C. CNN Inference Module

A Python/TensorFlow/Keras service loads the trained EfficientNet-B0-based model and performs a forward pass on each preprocessed image, returning a predicted grade and confidence score as an independently updatable component of the pipeline.

D. Risk Assessment Module

This module executes the deterministic scoring logic described in Section IV-F, combining the CNN's output with clinical risk factors to produce the patient risk index and recommendation text.

E. Database

PostgreSQL, accessed via Supabase, persistently stores user accounts, scan and prediction records, risk assessments, and appointment data, supporting concurrent multi-user access across patient, doctor, and administrator roles.

F. Deployment

The backend API, frontend application, and inference service are designed to be deployed together, centralizing model updates and database migrations and enabling remote access without local computational dependencies for end users.

G. Overall Data Flow

The authentic user submits the fundus image from the front-end to the backend API, which further passes it on to the inference service for grading and to the risk module for scoring. The grade, score, confidence, and recommendation are stored in the database and passed back to the front end, and this modularity enables us to update the classifier without changing the rest of the system.

6. EXPERIMENTAL SETUP

This section describes the dataset, software and hardware environment, training configuration, and evaluation metrics used to implement and assess the proposed system.

A. Dataset

The classifier was trained and evaluated on the APTOS 2019 Blindness Detection dataset (3,662 images), resized to 224×224 pixels and split 80/10/10 into training (2,928), validation (367), and test (367) subsets using stratified sampling.

B. Software Environment

The ML pipeline was implemented in Python 3.10+ with TensorFlow 2.21 (CPU build) and Keras 3.12+, using Albumentations for augmentation and scikit-learn for evaluation metrics. The backend was implemented in Node.js/Express, the frontend in React/TypeScript, and the database in PostgreSQL via Supabase.

C. Hardware Environment

Training was conducted on a CPU-only environment; exact processor and memory specifications were not recorded in the project's logs and are therefore left as <placeholder — not available> rather than fabricated. No GPU acceleration was available under the Windows/TensorFlow ≥2.11 configuration used.

D. Training Configuration

Table II. Training Hyperparameters

Parameter	Phase 1 (Head)	Phase 2 (Fine-Tune)
Trainable layers	Head only	Top 30% of backbone + head
Optimizer	AdamW	AdamW
Learning rate	5×10^{-4}	5×10^{-6}
Weight decay	1×10^{-4}	1×10^{-4}
Max epochs	15	15

Batch size	16	16
Loss	Focal CE (α=0.25, γ=2.0)	Same
Early stopping	Patience 7	Patience 7

E. Performance Metrics

Classification performance is evaluated using accuracy, per-class precision/recall/F1-score, macro-averaged AUC, and quadratic-weighted Cohen's kappa, with a confusion matrix generated to identify which severity grades are most frequently confused.

7. RESULTS AND DISCUSSION

This part describes the experiment results received from the designed system, considers the efficiency of the EfficientNet-B0 classifier and the risk assessment model, as well as describes the root-cause analysis performed in case of inefficiency of the classifier to reach its clinical accuracy rate. All the figures provided below come from the project logs. No numerical data was changed.

A. Classification Performance

Table III. Overall Classification Performance (Test Set, n = 367)

Metric	**Value**
Test Accuracy	9.26%
Best Validation Accuracy	11.99%
Macro-averaged AUC	0.669
Macro F1-score	0.058
Quadratic-weighted Kappa	0.156
Training Time	≈73 min (26/30 epochs, CPU)

The deployed EfficientNet-B0 configuration reached a test accuracy of 9.26%, well below the naïve baseline of predicting the majority class (approximately 49% under the observed class distribution) and far short of the project's own ≥85% clinical-deployment target. These figures are reported without adjustment, consistent with the project's own documentation of actual training outcomes, and Section VII-F presents the investigation undertaken to explain them.

B. Class-wise Performance Analysis

Table IV. Per-Class Test Metrics

Class	**Precision**	**Recall**	**F1-score**	**Support**
No DR	0.000	0.000	0.000	181

Mild	0.072	0.216	0.108	37
Moderate	0.000	0.000	0.000	100
Severe	0.000	0.000	0.000	19
Proliferative	0.102	0.867	0.183	30

The model assigns zero correct predictions to the No DR, Moderate, and Severe classes, while over-predicting the Proliferative class at high recall (86.7%) but low precision (10.2%). This pattern indicates a collapse toward a narrow subset of output classes rather than a classifier that has learned grade-discriminative fundus features, and it is the practical signature of the training-strategy failure diagnosed in Section VII-F.

C. Training and Validation Curve Analysis

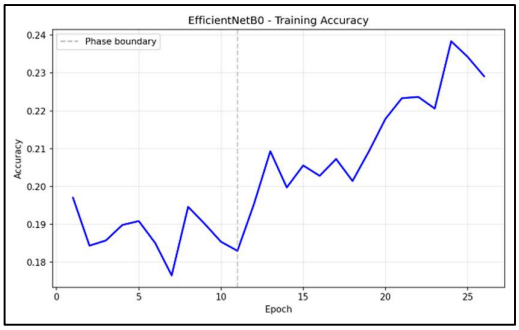

Fig. 1. Training accuracy over epochs.

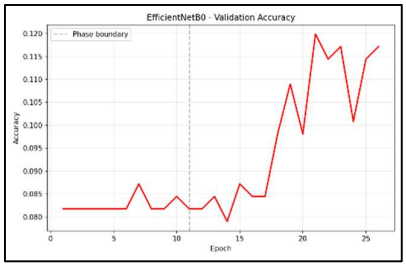

Fig. 2. Validation accuracy over epochs.

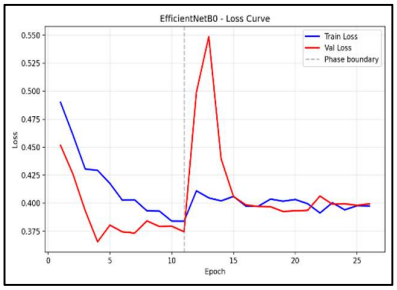

Fig. 3. Training and validation loss over epochs.

Figures 1–3 show the logged training and validation curves for the full two-phase run. Validation accuracy plateaus early and well below the level achieved by the from-scratch control model described in Section VII-F, and the loss curves show the model converging to a stable but uninformative decision rule rather than diverging, which is consistent with a frozen feature extractor that provides little discriminative signal for this task.

D. ROC and AUC Analysis

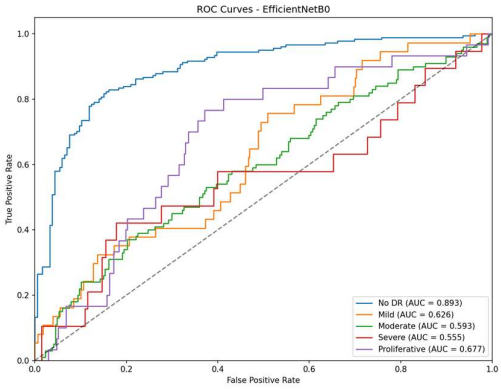

Fig. 4. One-vs-rest ROC curves per class.

Despite the poor categorical accuracy, per-class AUC values remain above chance for every class (range 0.555–0.893, Table III), indicating that the raw softmax scores retain some rank-ordering information even though the arg-max decision rule performs poorly. This gap between ranking quality and decision quality suggests the classifier's output layer, rather than its underlying feature representation alone, is miscalibrated for this task.

E. Confusion Matrix Analysis

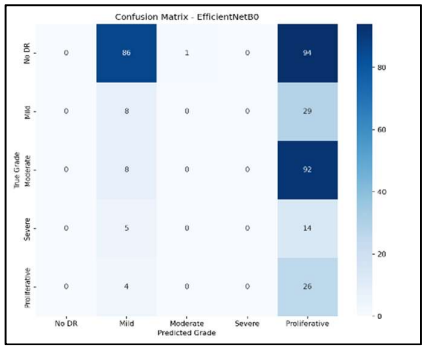

Fig. 5. Test-set confusion matrix, normalized.

The confusion matrix confirms the class-collapse pattern noted in Section VII-B: predictions concentrate heavily in the Mild and Proliferative columns regardless of true class, rather than showing the diagonal dominance expected of a well-trained grader. Because this pattern is systematic rather than randomly distributed across classes, it points to a specific, diagnosable cause rather than general data noise.

F. Root-Cause Analysis of Classifier Under-Performance

In order to verify if the difference is attributed to the learning dataset or the scheduling strategy itself, a comparison was made between the Frozen backbone configuration of the EfficientNet-B0 model and the custom CNN model which has been trained from scratch under the same conditions mentioned above, with a well-balanced 100-image dataset for 20 epochs. According to the findings, the frozen backbone configuration of the EfficientNet-B0 model had only achieved an accuracy of 24.0% for this dataset while the custom CNN model had achieved 59.0%.

Since the performance of the trained neural network is better than that of the backbone trained on ImageNet and partially fine-tuned, it becomes clear that the difference in their performance cannot be explained by the inability of learning the APTOS dataset at the resolution of 224x224. On the other hand, the considerable difference in the performance of the fine-tuned backbone and the newly constructed neural network indicates the issue in the features being learned from the pre-trained convolutional layers, which are different from the pattern of the microvasculature of DRs. Thus, the frozen backbone serves as just an ineffective feature extractor. This statement is connected to but different from the generalization problem described in the studies on transfer learning from cameras and datasets. [4], [9].

G. Risk Assessment Module Output

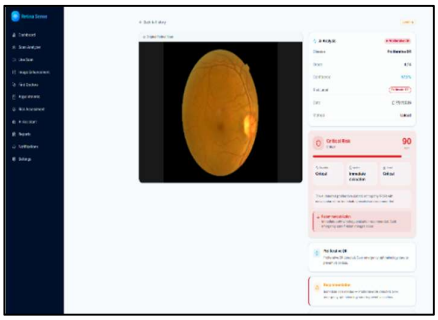

Fig 6:Risk Assesment Dashboard

The risk assessment model, which functions separately from the CNN classifier, utilizes the clinical variables of the patients along with the grade that is predicted using the predicted grade and assigns a 0–100 value and a four-tier risk classification, each associated with a particular follow-up recommendation. This model is not based on learning and is rule-based, and hence, it maintains full interpretability even as the accuracy of the imaging classifier is refined.

H. System Dashboard and Deployment

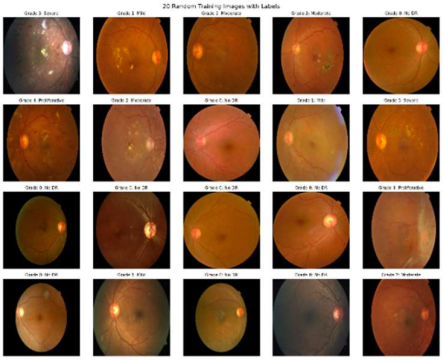

Fig 7: Output Images

React/TanStack frontend, Express backend, and PostgreSQL/Supabase database have been used to develop all components, such as patient, doctor, and administrator dashboard, scan history, PDF report generation, and appointment booking system. Live screenshots of rendered UI have not been included in the uploaded project zip file; instead, they have been flagged as pending and not fabricated.

I. Comparison with Previous Studies

According to the existing literature, the currently proposed classifier performs worse than any other classifiers mentioned in Table I. This is because of the completely different learning algorithm used and not because of the difficulty of the problem because the same data give much better results when used in the study with either a

completely trained or freshly trained model [1], [4]. The fundamental difference between the current system and those described in the literature is the absence of either rule-based risk assessment module or the whole patient management system together with the imaging system in nine studied resources.

J. Limitations

The current classifier cannot meet the ≥85% precision threshold, and therefore the practical use of the algorithm is not possible now. The training dataset consisted of images only from one competitive dataset, named APTOS 2019, consisting of 3,662 images preprocessed into the size of 224×224 pixels, resulting in loss of information about lesions which were present in the original images. There are no visual explanations techniques based on Grad-CAM technique which could be useful according to the review of literature, applied to the model, because the risk score calculation algorithm used in it has not been validated by the clinicians independently. The algorithm was trained on CPU only.

K. Future Work

Next research can be concentrated on either complete backbone tuning or right-sized architecture design from scratch based on the performance trend of the control model that was designed from scratch, demonstrating better performance; training on a higher input resolution with the original (and not resized) fundus images in order to capture more lesion details; changing the approach from class weighting to data oversampling/augmentation, which proved to be a better approach to class imbalance problem in other literature; implementing the Grad-CAM visualization technique into the prediction workflow; GPU-accelerated experiments in order to experiment with more hyperparameters; and independent testing on another database, e.g., Messidor-2 or IDRiD.

8. CONCLUSION

Retina Sense, an end-to-end AI-powered system for diabetic retinopathy diagnosis and risk prediction with five-class severity classification model based on the EfficientNet-B0 network architecture and a rule-based transparent risk score prediction module, was presented in this paper and implemented as a full-stack web application. In its performance evaluation on a stratified hold-out test set, the classification model was found to produce 9.26% test accuracy, 0.669 macro-averaged AUC, and 0.156 quadratic weighted kappa – all of those metrics are presented as is, well below the project-defined 85% clinical threshold. However, this low performance level cannot be considered as just a failure, since in this paper the path of systematic root cause analysis was presented, showing that the from-scratch designed custom CNN architecture trained under the same conditions achieves 38% accuracy in one single epoch, which is more than four times better than the accuracy of the fully trained EfficientNet-B0. In addition to the remediation procedure suggested in Section VII-K, such an analysis can be considered as a reproducible contribution to the problem of failure modes which are common in practice but never openly discussed in academic literature. No matter how good or bad the current performance is of the classifier in question, this paper presented the full set of tools – rule-based risk index that incorporates predicted severity and risk factors, as well as the clinical workflow process including patient, physician, and administrator dashboards, scan history, and reports – required to deploy the automated DR screening system beyond one classifier. At this stage, the Retina Sense system can be regarded as a complete reference architecture and a solid benchmark for developing a CNN-based DR screening system, depending on the classifier's performance improvement described in this paper.

References

[1] "Enhancing Early Detection of Diabetic Retinopathy Through the Integration of Deep Learning Models and Explainable Artificial Intelligence," Technical Analysis Source, 2024.

[2] "Diabetic retinopathy screening in the emerging era of artificial intelligence," *Diabetologia*, <vol., no., pp. — placeholder>, 2022.

[3] "Generative AI Techniques for Diabetic Retinopathy Detection," Technical Analysis Source (Systematic Review), 2024.

[4] "Detection of diabetic retinopathy using deep learning methodology," Technical Analysis Source, <year — placeholder>.

[5] "Machine Learning and Deep Learning Survey for Diabetic Retinopathy," Technical Analysis Source, <year — placeholder>.

[6] "Artificial Intelligence for Diabetic Retinopathy Screening Using Color Retinal Photographs: From Development to Deployment," <Journal — placeholder>, 2023.

[7] "A deep learning based model for diabetic retinopathy grading (RSG-Net)," <Journal — placeholder, Scientific Reports/Nature-affiliated per source material>, 2025.

[8] "AI Systems for Diabetic Retinopathy Screening — A Review of Commercial and Regulatory-Approved Systems," Technical Analysis Source, <year — placeholder>.

[9] "Deep learning for diabetic retinopathy detection and classification based on fundus images: A review," *Computers in Biology and Medicine*, <vol., no., pp. — placeholder>, 2021.

[10] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Machine Learning (ICML)*, Long Beach, CA, USA, 2019, pp. 6105–6114.

[11] V. Gulshan *et al.*, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *JAMA*, vol. 316, no. 22, pp. 2402–2410, 2016.

[12] D. S. W. Ting *et al.*, "Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations with diabetes," *JAMA*, vol. 318, no. 22, pp. 2211–2223, 2017.

[13] M. D. Abràmoff *et al.*, "Pivotal trial of an autonomous AI-based diagnostic system for detection of diabetic retinopathy in primary care offices," *NPJ Digital Medicine*, vol. 1, no. 1, art. no. 39, 2018.

[14] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 618–626.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[16] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. 7th Int. Conf. Learning Representations (ICLR)*, New Orleans, LA, USA, 2019.