Lebanese University
Faculty of Information, Branch I

Bachelor of Data Science

DR Eye Platform:

An AI-Powered Diabetic Retinopathy Grading and Clinical Referral System

A Graduation Report
Academic Year 2025–2026

Prepared By

Jana sayed Ahmad
Mhamad El-khatib
Karim hajj chhade

Supervised By

Dr. Abbass Rammal

Acknowledgements

First and foremost, we extend our deepest and most sincere gratitude to our supervisor, Dr.Abbass Rammal, whose intellectual generosity, unwavering patience, and meticulous guidance were the cornerstone of this project from its earliest conception to its final submission. His ability to simultaneously challenge our assumptions and validate our efforts created exactly the academic environment in which this work could develop and flourish. Every meeting with him left us better equipped, more rigorous, and more confident in our direction. This project would not exist in its current form without him.

We are profoundly grateful to the Faculty of Information and Documentation at the Lebanese University for providing the academic environment, the curriculum, and the institutional support that made this level of work possible. The Data Science program, with its emphasis on both theoretical foundations and practical application, gave us the tools — in statistics, machine learning, software engineering, and data management — that we drew on constantly throughout this project. We are proud to be graduates of this institution.

This project stands on the shoulders of an enormous global open-source community, and we wish to acknowledge that debt explicitly. To the PyTorch development team, whose deep learning framework is the technical foundation of our grading model. To Ross Wightman and the contributors of the timm library, whose curated collection of pretrained models made transfer learning accessible and reproducible. To the Flask development team for a web framework that is simultaneously simple and powerful. To the contributors of OpenCV, NumPy, Pillow, ReportLab, and every other library we imported. And to the team at Anthropic, whose Claude API gave our platform a genuinely conversational and bilingual intelligence that transformed it from a classification tool into a clinical support system.

We acknowledge with full transparency the computational constraints under which this project was completed. The Google Colab free-tier environment, with its shared GPU access, session timeouts, and quota limitations, presented genuine engineering challenges that shaped many of our implementation decisions. We have documented these constraints honestly throughout this report because we believe that scientific transparency includes transparency about the conditions of production, not only the results.

We extend warm thanks to the ophthalmology community whose published research, shared datasets, and clinical insight provided the domain knowledge without which no AI system in this space can be trustworthy. In particular, the researchers behind the APTOS 2019 dataset, the authors of the five papers reviewed in our literature survey, and the broader retinal imaging research community whose accumulated work we have built upon.

Finally, and above all else, we thank our families. For the dinners eaten cold because we were still coding. For the questions asked and the patience shown. For the belief extended without condition. This work is dedicated to them.

Table of Contents

List of Figures

List of Tables

Abstract

Diabetic retinopathy (DR) is the leading cause of preventable blindness among adults with diabetes worldwide, affecting an estimated 103 million individuals and threatening the vision of over 28 million more. Despite the availability of effective treatments — including laser photocoagulation, anti-vascular endothelial growth factor (anti-VEGF) injections, and vitreoretinal surgery — the majority of DR cases progress to vision-threatening stages without detection, primarily because specialist-level retinal screening is inaccessible, unaffordable, or unavailable to large portions of the global diabetic population, including communities across Lebanon and the wider Arab world.

This graduation project presents the DR Eye Platform, a comprehensive, end-to-end clinical decision support system that addresses the diabetic retinopathy screening crisis through two complementary interventions. The first is a deep learning grading engine that automatically classifies retinal fundus photographs into five internationally standardized severity grades: No DR, Mild, Moderate, Severe, and Proliferative DR. The second is a full-stack bilingual web platform that embeds this grading engine within a clinically useful workflow, connecting screened patients directly with registered specialist doctors through WhatsApp-based report delivery, without requiring phone calls, secretarial coordination, or physical visits for the initial referral.

Five deep learning architectures spanning the period from 2015 to 2022 were systematically trained, compared, and evaluated on the APTOS 2019 Blindness Detection dataset, comprising 3,662 labeled retinal fundus photographs from rural clinics across India. The five architectures are: a Simple Convolutional Neural Network trained from scratch as a controlled baseline, ResNet50 (He et al., 2016), DenseNet121 (Huang et al., 2017), EfficientNetB4 (Tan and Le, 2019), and ConvNeXt-Base (Liu et al., 2022). All models were trained using an identical experimental protocol consisting of a three-stage image preprocessing pipeline (black border cropping, resolution-appropriate resizing, and Contrast-Limited Adaptive Histogram Equalization), a two-phase transfer learning strategy (backbone

freezing followed by full fine-tuning), and WeightedRandomSampling to address the significant class imbalance inherent in the dataset.

Performance was evaluated using the Quadratic Weighted Kappa (QWK) metric, the standard ordinal agreement measure for five-class DR grading, used in all referenced publications and in the original APTOS 2019 Kaggle competition. Results demonstrate a clear, monotonic improvement in performance across architectural generations: the SimpleCNN baseline achieved QWK = 0.6087; ResNet50 achieved 0.8272; EfficientNetB4 achieved 0.8415; DenseNet121 achieved 0.8489; and ConvNeXt-Base achieved 0.8888 — effectively matching the published state-of-the-art benchmark of 0.89 established by Chetoui and Akhloufi (2020). Grad-CAM visual explanations applied to the ConvNeXt-Base model confirm that predictions are grounded in clinically relevant retinal features, and a softmax-based confidence scoring mechanism correctly identified the sole observed misclassification as low-confidence (37.4%), demonstrating a meaningful safety signal for clinical use.

The web application is implemented in Python using the Flask framework and provides a complete, guided three-step patient workflow: an optional retinal image upload with instant AI grading, a Claude AI-powered bilingual symptom chatbot that collects clinical context in English or Arabic, and the generation of a structured medical report in both languages. Patients who have no retinal photograph — an anticipated common scenario in low-resource settings — may skip the image upload and receive a symptom-informed report and doctor referral. Report delivery to specialist doctors occurs via the Twilio WhatsApp API. Registered patients have access to a longitudinal screening history dashboard with Claude AI-generated trend analysis across multiple visits. Doctors self-register, appear in a filterable directory, and receive reports on WhatsApp. All components of the platform are modular, extensible, and documented.

All limitations of the project are documented with full transparency, including the substitution of the APTOS 2019 dataset for the originally planned 2015 EyePACS dataset, the incomplete Monte Carlo Dropout implementation due to computational constraints, the local-only deployment environment, and the Twilio sandbox restriction on WhatsApp delivery. A concrete and prioritized roadmap for addressing these limitations in future work is provided.

Keywords: *Diabetic Retinopathy, Deep Learning, Transfer Learning, Convolutional Neural Networks, ConvNeXt, EfficientNet, DenseNet121, ResNet50, Quadratic Weighted Kappa, Grad-CAM, Confidence Estimation, Flask, Python, Claude AI, Bilingual NLP, Medical Decision Support, Telemedicine, WhatsApp Integration, Patient Referral System, Arabic AI.*

Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
API	Application Programming Interface
APTOS	Asia Pacific Tele-Ophthalmology Society
AUC	Area Under the ROC Curve
BN	Batch Normalization
CLAHE	Contrast-Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
CSS	Cascading Style Sheets
DL	Deep Learning
DR	Diabetic Retinopathy
EHR	Electronic Health Record
ER	Entity-Relationship
F1	F1 Score (harmonic mean of precision and recall)
GELU	Gaussian Error Linear Unit
GPU	Graphics Processing Unit
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
JS	JavaScript
JSON	JavaScript Object Notation
LLM	Large Language Model
LN	Layer Normalization
MBConv	Mobile Inverted Bottleneck Convolution

MCD	Monte Carlo Dropout
ML	Machine Learning
NAS	Neural Architecture Search
NLP	Natural Language Processing
PDF	Portable Document Format
QWK	Quadratic Weighted Kappa
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristic
SGD	Stochastic Gradient Descent
SQL	Structured Query Language
UI	User Interface
UX	User Experience
WHO	World Health Organization
WRS	WeightedRandomSampler

Chapter 1: Introduction

1.1 Background and Motivation

Diabetes mellitus stands as one of the defining public health challenges of the twenty-first century. According to the International Diabetes Federation's 2021 Diabetes Atlas, an estimated 537 million adults aged 20 to 79 years were living with diabetes globally — a figure that represents approximately one in ten adults worldwide — and this number is projected to climb to 643 million by 2030 and 783 million by 2045 if current trends continue unchecked. The social, economic, and medical consequences of this epidemic are profound and multidimensional, spanning increased cardiovascular risk, renal failure, neuropathy, lower-limb amputation, and — crucially for this project — the progressive destruction of vision.

Diabetic retinopathy is the microvascular complication of diabetes that targets the retina, the thin, light-sensitive tissue that lines the posterior wall of the eye and is responsible for converting light into the neural signals that constitute vision. When blood glucose remains chronically elevated, it damages the walls of the retinal capillaries in a cascade that begins with increased vascular permeability and microaneurysm formation, progresses through hemorrhage, lipid deposition, and ischemia, and can culminate in the growth of new, structurally abnormal blood vessels — a process called neovascularization — that can bleed into the vitreous humor or cause tractional retinal detachment, leading to complete and irreversible blindness. According to the most comprehensive global estimates available, diabetic retinopathy affects approximately 34.6% of people with diabetes (Arora et al., 2024), translating to over 103 million individuals. Of these, approximately 28 million experience sight-threatening forms of the disease.

The clinical and epidemiological irony of diabetic retinopathy is stark: it is almost entirely preventable with timely detection, and yet it remains the leading cause of new cases of blindness among working-age adults in most high-income countries and a major cause of blindness in middle- and low-income countries. The reason for this paradox is not the absence of effective treatment — laser photocoagulation can prevent up to 90% of severe vision loss when applied at the right stage, and anti-VEGF agents have transformed the treatment of diabetic macular edema — but rather the systematic failure to detect the disease early enough for these treatments to be effective. This failure is rooted in a global shortage of retinal screening capacity: the skilled ophthalmologists or graders who must examine each retinal photograph are scarce, maldistributed, expensive, and overwhelmed.

In Lebanon, where this project is situated, the ophthalmological workforce is concentrated in Beirut and a small number of urban centers, leaving patients in rural areas of the Bekaa Valley, the North, and the South with severely limited access to specialist eye care. The Lebanese healthcare system, under the compound pressures of economic collapse, currency devaluation, brain drain of healthcare professionals, and reduced hospital capacity since 2019, has seen patient access to specialist services deteriorate significantly. A diabetic patient in a rural Lebanese district may wait months for an ophthalmology appointment, and the associated costs — including consultation fees, transportation, and time off work — represent a substantial burden. Annual retinal screening, which is the clinical recommendation for all patients with diabetes, is simply not happening at the required frequency or scale.

Artificial intelligence, and deep learning in particular, emerged in the mid-2010s as a potentially transformative technology for bridging this screening gap. The key insight — first demonstrated at a clinically convincing scale by Gulshan et al. in their landmark 2016 JAMA publication — is that convolutional neural networks trained on large labeled datasets of retinal fundus photographs can learn to classify images into DR severity categories with accuracy approaching that of trained human graders. If this grading step can be automated reliably, then the bottleneck shifts from the availability of specialist graders to the availability of fundus cameras and image transmission infrastructure, both of which are significantly more scalable and addressable.

However, the research literature on AI-based DR screening reveals an important distinction between demonstrating technical feasibility in controlled experimental settings and delivering genuine clinical utility in the real world. Many published systems achieve impressive accuracy metrics on benchmark datasets but were never integrated into a patient-facing workflow. The grading model is only one component of a complete screening pathway; the others — patient intake, symptom collection, report generation, specialist referral, appointment scheduling — require equal attention if the technology is to deliver actual healthcare benefit rather than academic results. This distinction between performance and deployment is the central motivation for the DR Eye Platform.

1.2 Problem Statement

This project addresses the following central research and engineering question: How can deep learning-based diabetic retinopathy grading be systematically developed, rigorously compared across architectural generations, made interpretable and uncertainty-aware, and embedded within a complete, bilingual, patient-centered clinical referral platform designed for the constraints and context of the Lebanese and Arab healthcare environment?

This question decomposes into five distinct sub-problems that the project addresses in sequence:

Sub-problem 1 — Grading Accuracy: Can a deep learning model, trained on a moderately sized public dataset of retinal fundus photographs, achieve a Quadratic Weighted Kappa score of 0.85 or above on five-class DR grading — a threshold broadly associated with clinically meaningful performance in the reviewed literature — without access to industrial-scale computational resources?

Sub-problem 2 — Architectural Comparison: How does performance on this specific clinical grading task vary across five representative deep learning architectures spanning the period from 2015 to 2022 — SimpleCNN (baseline), ResNet50, DenseNet121, EfficientNetB4, and ConvNeXt-Base — when training conditions are held constant? Does architectural advancement produce consistent, measurable performance improvements, and if so, what does the magnitude of these improvements reveal about the specific characteristics of the task?

Sub-problem 3 — Interpretability: Can the best-performing model's predictions be made visually interpretable through Gradient-weighted Class Activation Mapping (Grad-CAM), producing heatmap visualizations that highlight the retinal regions driving each prediction and enabling a clinician to assess whether the model is attending to clinically appropriate features?

Sub-problem 4 — Uncertainty Quantification: Can the model communicate its own predictive uncertainty in a way that is useful in a clinical context — specifically, can a confidence scoring mechanism correctly identify its least reliable predictions and flag them for human expert review, thereby reducing the risk of a confidently incorrect automated grade being accepted without scrutiny?

Sub-problem 5 — Clinical Accessibility: Can the grading model be embedded within a web platform that allows patients — including those who do not have access to retinal photography equipment — to receive a clinically structured AI-generated medical report in both English and Arabic, and to contact a registered specialist doctor directly and immediately without navigating the traditional barriers of phone calls, secretarial gatekeeping, and long waiting periods?

1.3 Project Objectives

The following concrete, measurable objectives were established at the project's inception and guided all subsequent design, implementation, and evaluation decisions:

Objective 1 — Dataset Preparation: Acquire, analyze, and preprocess the APTOS 2019 Blindness Detection dataset, documenting the class distribution, implementing a stratified train/validation split, and applying a three-stage preprocessing pipeline consisting of black border cropping, resolution-appropriate resizing, and CLAHE contrast enhancement.

Objective 2 — Class Imbalance Correction: Implement and validate WeightedRandomSampling as the primary strategy for addressing the significant class imbalance in the APTOS 2019 dataset, assigning per-image sampling weights inversely proportional to class frequency.

Objective 3 — Baseline Establishment: Train a Simple Convolutional Neural Network from randomly initialized weights on the preprocessed dataset and document its performance across all evaluation metrics as a controlled baseline for comparison.

Objective 4 — Transfer Learning Models: Train four pretrained deep learning architectures — ResNet50, DenseNet121, EfficientNetB4, and ConvNeXt-Base — using an identical two-phase transfer learning protocol (Phase 1: frozen backbone fine-tuning; Phase 2: full model fine-tuning) and document performance for each.

Objective 5 — Performance Target: Achieve a Quadratic Weighted Kappa score of at least 0.84 for the best-performing model, with a stretch target of 0.88 approaching the published benchmark of 0.89 established by Chetoui and Akhloufi (2020).

Objective 6 — Explainability: Implement Grad-CAM visualization applied to the final convolutional layer of the best-performing model, producing and interpreting heatmap overlays for representative cases from each severity grade.

Objective 7 — Confidence Scoring: Implement a softmax-based confidence scoring mechanism, define a threshold below which predictions are flagged for specialist review, and validate the mechanism against the observed misclassification cases.

Objective 8 — Web Platform: Design and implement a complete Flask web application providing: optional retinal image upload with AI grading; Claude AI-powered bilingual symptom chatbot; structured AI-generated medical reports in English and Arabic; doctor directory with city and specialty filtering; WhatsApp-based report delivery to specialists via Twilio; PDF report generation via ReportLab; patient registration and longitudinal history dashboard; and doctor registration and received-reports dashboard.

Objective 9 — Documentation: Document all technical decisions, implementation details, limitations, and future work with the depth and honesty expected of academic publication.

1.4 Scope and Boundaries

The DR Eye Platform is explicitly designed and presented as a research prototype and proof-of-concept system. It is not a medically certified device, and it makes no claim to regulatory approval under any healthcare device classification framework. The AI-generated grades and reports are intended to provide accessible, personalized information to patients and to assist, not replace, the clinical judgment of licensed ophthalmologists.

Within this clearly stated framework, the project is bounded in the following ways. The image classification model has been trained and validated exclusively on the APTOS 2019 dataset, which represents retinal photographs collected from rural Indian clinics using specific fundus camera equipment in 2019. Performance on images from different cameras, different clinical environments, different patient populations (including Lebanese patients), or different imaging protocols has not been tested and cannot be assumed without further validation studies. The platform deployment is local, accessed via a temporary public URL generated by ngrok during demonstration sessions, and is not configured for multi-user concurrent access at scale. The WhatsApp integration operates within Twilio's free sandbox environment, which restricts message delivery to numbers that have opted into the sandbox. The Claude API-generated content is produced by a commercial large language model and carries all of the limitations, including potential hallucination, that are inherent in such systems.

These boundaries are not deficiencies of the work but rather honest documentation of the gap between a completed academic prototype and a production-ready clinical deployment — a gap that is deliberately and explicitly addressed in the future work chapter.

1.5 System Architecture Overview

The DR Eye Platform integrates five principal subsystems, each of which is developed and documented in dedicated chapters of this report. Figure 1.1 provides a high-level architectural overview.

The first subsystem is the **Image Preprocessing and Grading Engine**, implemented in ml.py. This component receives a raw retinal fundus photograph from the patient's browser upload, applies the three-stage preprocessing pipeline, and passes the preprocessed image tensor through the ConvNeXt-Base model to produce a severity grade prediction (0–4), a confidence score, a probability distribution across all five grades, and a Grad-CAM heatmap. When no image is provided, this subsystem returns a symptom-based urgency estimate computed from the patient's reported symptoms and diabetes duration.

The second subsystem is the **Claude AI Integration Layer**, implemented in ai.py. This component manages all interactions with the Anthropic Claude API, providing four distinct functions: (1) the bilingual symptom chatbot that collects clinical context from the patient in Step 2 of the workflow; (2) English-language medical report generation combining the grading result with the chatbot-collected clinical context; (3) Arabic-language medical report generation using the same inputs; and (4) longitudinal history analysis for registered patients with multiple screening records.

The third subsystem is the **Web Application Layer**, implemented as a Flask application with modular blueprint-based routing. This layer manages user sessions, patient and doctor authentication, form processing, image file handling, and the assembly and rendering of all page templates.

The fourth subsystem is the **Data Persistence Layer**, implemented as a SQLite relational database accessed through Python's sqlite3 module. This layer stores patient accounts, doctor registrations, screening records, and chat session histories, and provides the data backbone for the longitudinal history feature.

The fifth subsystem is the **Communication and Document Layer**, implemented in services.py. This component handles two output functions: structured WhatsApp message delivery to doctor accounts via the Twilio API, and professional bilingual PDF report generation via the ReportLab library.

These five subsystems interact in the sequence defined by the patient workflow: the patient uploads an image (or skips), which triggers the Grading Engine; the chatbot collects context through the Claude AI Layer; the Report Generator produces bilingual output combining both sets of inputs; the patient selects a doctor from the directory; and the Communication Layer delivers the report to the chosen doctor. All interactions are mediated by the Web Application Layer, and all persistent data flows through the Data Persistence Layer.

1.6 Research and Engineering Contributions

This project makes the following original contributions:

Contribution 1 — Systematic Multi-Architecture Comparison: A rigorous controlled comparison of five deep learning architectures for five-class DR grading on APTOS 2019, with identical preprocessing, training protocol, and evaluation metrics, providing a documented performance trajectory from 2015 to 2022 that is directly comparable to the published benchmark.

Contribution 2 — Integrated Explainability and Confidence: The combination of Grad-CAM visual explanation and softmax confidence scoring in a single deployed clinical tool, demonstrating that uncertainty awareness and visual interpretability can be delivered together without architectural modification.

Contribution 3 — Bilingual Arabic-English Clinical Platform: To our knowledge, the first Arabic-English bilingual DR screening platform that combines AI image grading, LLM-generated clinical reports, and specialist referral in a single workflow. The bilingual design is not a superficial translation layer but a full dual-language system in which all clinical content — chatbot conversations, medical reports, PDF outputs, and WhatsApp messages — is generated natively in the target language.

Contribution 4 — Optional Image Architecture: The design decision to make retinal image upload optional — allowing patients without access to fundus cameras to still receive a symptom-informed report and specialist contact — represents a specific and deliberate accessibility innovation that distinguishes this platform from the majority of published DR screening systems, which assume image availability.

Contribution 5 — Complete Clinical Pathway Implementation: The integration of grading, reporting, and referral into a single continuous workflow that a patient can navigate without external assistance or prior medical knowledge, demonstrated in a live deployed prototype.

1.7 Report Organization

The remainder of this report is organized as follows. **Chapter 2** provides a detailed literature review of the five key papers that grounded the design of this project, with explicit mapping of each paper's contributions to specific implementation decisions. **Chapter 3** describes the APTOS 2019 dataset in full detail, including the rationale for its selection, class distribution analysis, preprocessing pipeline design and implementation, data augmentation strategy, and evaluation metric selection and justification. **Chapter 4** provides a comprehensive technical account of all five model architectures, including mathematical formulations of their defining innovations, implementation details in PyTorch, hyperparameter configurations, and the two-phase training procedure. **Chapter 5** presents and discusses the complete experimental results for all five models, including training histories, per-class classification reports, confusion matrices, and QWK comparisons with the published benchmark. **Chapter 6** covers the explainability and confidence components in detail, including the Grad-CAM algorithm, its implementation, and a case-by-case analysis of representative predictions. **Chapter 7** provides a complete technical description of the Flask web application, including its architecture, database design, route implementations, security considerations, Claude AI integration, and communication layer. **Chapter 8** presents the system demonstration with annotated screenshots and walkthrough commentary. **Chapter 9** documents all limitations with full transparency and analytical depth. **Chapter 10** outlines a prioritized roadmap for future development. **Chapter 11** concludes the report with a synthesis of contributions and reflection on the project's significance. References and Appendices follow.

Chapter 2: Literature Review

This chapter provides a comprehensive review of the five foundational published works that directly informed the design, implementation, and evaluation of the DR Eye Platform. The review is structured to extract from each paper not only its headline findings but the specific methodological contributions, technical details, and limitations that influenced concrete decisions in this project. The chapter concludes with a synthesis table that makes the literature-to-design mapping explicit, and a brief discussion of the broader research context in which this project is situated.

The five papers were selected through a targeted search of the clinical AI and medical imaging literature with the following criteria: (1) direct relevance to deep learning for diabetic retinopathy grading on the five-class severity scale used in this project; (2) introduction or validation of a technique, architecture, or methodology adopted in this project; (3) use of the APTOS 2019 or comparable dataset; and (4) availability in a peer-reviewed venue with sufficient technical detail for methodological replication. The selection is not intended to be exhaustive but to be specifically and directly relevant to the design decisions of this project.

2.1 Gulshan et al. (2016) — Deep Learning for DR Detection: Foundational Validation

2.1.1 Citation and Context

V. Gulshan, L. Peng, M. Coram, 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, Dec. 2016.

This paper represents the watershed moment in the application of deep learning to diabetic retinopathy screening. Published in the Journal of the American Medical Association — one of the most influential clinical journals in the world — it carried the imprimatur of clinical credibility that translated academic machine learning results into a genuine conversation about deployment in healthcare systems. Its significance cannot be overstated: before this paper, AI-based retinal screening was largely a laboratory curiosity; after it, health systems around the world began actively exploring implementation.

2.1.2 Methods and Results

Gulshan et al. trained a deep learning algorithm — specifically, an ensemble of neural networks based on the Inception architecture — on 128,175 retinal fundus images from the EyePACS dataset, which were graded by a panel of 54 certified ophthalmologists and ophthalmology residents to produce ground-truth labels at two granularities: a binary "referable vs. non-referable DR" distinction and a more detailed five-level severity scale. The algorithm was trained for binary detection (referable DR, defined as moderate DR or worse) and validated on two independent test sets: the EyePACS-1 set (9,963 images) and the Messidor-2 set (1,748 images).

The results were compelling: on EyePACS-1, the algorithm achieved an area under the ROC curve (AUC) of 0.991, with sensitivity of 90.3% and specificity of 98.1% at the high-specificity operating point. On Messidor-2, the AUC was 0.990. Importantly, at the high-sensitivity operating point, the algorithm's sensitivity of 97.5% on EyePACS-1 exceeded the majority of the eight individual ophthalmologists included in the comparison, and its specificity of 93.4% was within the range of individual grader performance. These results demonstrated, for the first time at clinical scale, that a deep learning system could perform DR screening at a level broadly comparable to trained specialists.

2.1.3 Contribution to This Project

Gulshan et al. (2016) provides the foundational clinical and motivational context for the entire DR Eye Platform. It established that AI-based DR screening is not merely a theoretical possibility but a demonstrated technical achievement, and that the screening gap — the failure to deliver routine retinal examination to the global diabetic population — can be meaningfully addressed by automation of the grading step. The paper's documentation of the scaling requirements — 128,175 training images — also implicitly establishes the dataset size limitation that this project must contend with and document honestly, and motivates the transfer learning approach adopted across all four pretrained models. The paper's framing of the access problem (specialist shortage, geographic distribution, cost) directly motivated the referral platform component of the DR Eye Platform, which addresses not just the grading step but the pathway from a positive screening result to specialist consultation.

2.2 Chetoui and Akhloufi (2020) — EfficientNet with Grad-CAM: The Primary Benchmark

2.2.1 Citation and Context

M. Chetoui and M. A. Akhloufi, "Explainable Diabetic Retinopathy Using EfficientNet," in Proc. 42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1966–1969, 2020.

This paper is the single most directly influential work on the design of this project and serves as the primary quantitative benchmark against which the DR Eye Platform's grading engine is evaluated. It is the paper that established the specific technical approach — EfficientNet architecture, APTOS 2019 dataset, five-class grading, Quadratic Weighted Kappa evaluation, and Grad-CAM explainability — that this project adopts and extends.

2.2.2 Methods and Results

Chetoui and Akhloufi applied the EfficientNet family of architectures to the five-class diabetic retinopathy grading task on the APTOS 2019 dataset combined with the 2015 EyePACS data. The EfficientNet models were pretrained on ImageNet and fine-tuned on the retinal dataset. The authors experimented with multiple variants of EfficientNet (B0 through B7) and reported that EfficientNet achieved a Quadratic Weighted Kappa score of approximately 0.89 on the APTOS 2019 validation set — effectively matching the performance of the Kaggle competition winners on the same dataset. The paper also applied Grad-CAM to the trained models, producing heatmap visualizations that highlighted retinal lesions such as microaneurysms, hemorrhages, and hard exudates as the primary drivers of grade predictions.

The Grad-CAM results provided qualitative evidence that the EfficientNet model had learned to attend to clinically meaningful retinal features rather than imaging artifacts or confounding factors such as camera glare or vascular arcades unrelated to DR severity. This is an important validation step beyond accuracy metrics, because a model that achieves high accuracy by attending to non-clinical image features would not be trustworthy for clinical deployment, regardless of its numerical performance.

2.2.3 Contribution to This Project

Chetoui and Akhloufi (2020) contributed to this project in five specific ways. First, it established the Quadratic Weighted Kappa benchmark of 0.89 that this project targets, enabling precise comparison with a published result obtained under comparable experimental conditions on the same dataset. Second, it validated the choice of APTOS 2019 as a benchmark dataset for five-class DR grading with published QWK results. Third, it introduced EfficientNetB4 into this project as the Model 4 experimental condition, chosen because B4 is the variant most commonly used in subsequent literature for single-dataset fine-tuning tasks. Fourth, it provided the methodological template for combining grading performance with Grad-CAM explainability, which this project implements using the pytorch-grad-cam library applied to the ConvNeXt-Base model's final convolutional layer. Fifth, it demonstrated that high QWK performance and visual interpretability are not mutually exclusive but can be achieved simultaneously in a single system.

2.3 Selvaraju et al. (2017) — Grad-CAM: The Explainability Foundation

2.3.1 Citation and Context

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 International Conference on Computer Vision (ICCV), pp. 618–626, 2017.

This paper introduced Gradient-weighted Class Activation Mapping (Grad-CAM), the visual explanation technique that is implemented in this project for all heatmap visualizations. Grad-CAM has become one of the most widely adopted explainability methods in deep learning, particularly in medical imaging applications, because it requires no modification to the model architecture, is computationally cheap to compute at inference time, and produces spatially localized visualizations that are intuitively interpretable.

2.3.2 Methods and Mathematical Formulation

Grad-CAM generates a class-discriminative localization map for a target class c using the gradient of the score y^c (before softmax normalization) with respect to the feature map activations A^k of a convolutional layer. The method proceeds as follows.

First, for an input image I and target class c, the gradient of the class score with respect to the feature map of the final convolutional layer is computed:

$$\partial y^c / \partial A^k_{ij}$$

Second, these gradients are globally average-pooled over the spatial dimensions (indexed by i and j) to produce the importance weight α^c_k for feature map k with respect to class c:

$$\alpha^c_k = (1/Z) \Sigma_i \Sigma_j (\partial y^c / \partial A^k_{ij})$$

where Z is the number of pixels in the feature map. This pooling operation produces a single scalar weight per feature map channel, representing how much that channel's activations, in aggregate, contribute to the score for class c.

Third, a weighted combination of the feature maps is computed and passed through a ReLU activation:

$$L^c\_Grad\text{-}CAM = ReLU(\Sigma_k \alpha^c_k \cdot A^k)$$

The ReLU is applied because only features with a positive influence on the target class are relevant to the explanation — features that decrease the class score are suppressed. The result L^c_Grad-CAM is a coarse heatmap of the same spatial dimensions as the final convolutional layer's feature maps, which is upsampled to the input image resolution using bilinear interpolation and overlaid on the original image using a color map (jet colormap in this implementation) where red indicates high importance and blue indicates low importance.

2.3.3 Contribution to This Project

Selvaraju et al. (2017) provides the complete algorithmic basis for the explainability component of the DR Eye Platform. Every heatmap visualization generated by the platform implements exactly this algorithm, applied using the pytorch-grad-cam library's GradCAM class with the target layer set to the conv_head convolutional layer of the ConvNeXt-Base model. The paper is cited whenever Grad-CAM is discussed in this report, and the mathematical formulation above is the basis for the implementation description in Chapter 6.

2.4 Arora et al. (2024) — Ensemble Methods and Generalization Limits

2.4.1 Citation and Context

A. Arora et al., "Ensemble Deep Learning and EfficientNet for Accurate Diagnosis of Diabetic Retinopathy," Scientific Reports, vol. 14, 2024.

This recent paper in Nature's Scientific Reports provides two contributions directly relevant to this project: updated global epidemiological data on the prevalence of diabetic retinopathy — establishing that DR affects approximately 34.6% of people with diabetes, the figure used throughout this report — and a rigorous discussion of the distribution shift problem that represents one of the most significant barriers to real-world deployment of AI-based DR screening systems.

2.4.2 Methods and Results

Arora et al. developed an ensemble approach combining multiple pretrained convolutional neural networks for DR classification, demonstrating improved performance over individual models on benchmark datasets. More importantly for this project, the paper's discussion section provides a detailed analysis of why models trained on one dataset frequently fail to generalize to images from different clinical settings: differences in fundus camera hardware produce systematic variations in image appearance (brightness, contrast, color balance, spatial resolution) that can shift the feature distribution encountered at inference time away from the distribution the model was trained on, degrading performance in ways that are difficult to predict without explicit cross-dataset testing.

2.4.3 Contribution to This Project

Arora et al. (2024) contributed the epidemiological framing used throughout this report's introduction and clinical motivation sections, and — more critically — provided the conceptual and empirical basis for Limitation 4 documented in Chapter 9. The limitation that the ConvNeXt-Base model trained in this project has been validated only on the APTOS 2019 dataset and cannot be assumed to generalize to images from Lebanese hospitals, different camera models, or different patient populations is directly motivated by the distribution shift analysis in this paper. This limitation is cited with reference to Arora et al. (2024) in Chapter 9.

2.5 Vij and Arora (2023) — Class Imbalance in Medical Imaging Classification

2.5.1 Citation and Context

R. Vij and S. Arora, "A Novel Deep Transfer Learning Based Computerized Diagnostic System for Multi-Class Imbalanced Diabetic Retinopathy Severity Classification," Multimedia Tools and Applications, 2023.

This paper addresses a problem that is universal in real-world medical classification datasets: class imbalance. In the context of DR grading, the natural epidemiological distribution of disease severity in a screened diabetic population means that the vast majority of patients have no or minimal disease, while the clinically most urgent cases — Severe and Proliferative DR — represent a small minority. Training a classifier on such a distribution without correction produces systematic bias toward the majority class, yielding a model that appears highly accurate on aggregate metrics while performing unacceptably on the cases that matter most clinically.

2.5.2 Methods and Results

Vij and Arora evaluated and compared three principal strategies for addressing class imbalance in DR classification: (1) random oversampling of minority-class examples through image duplication; (2) random undersampling of majority-class examples; and (3) weighted sampling, in which the selection probability of each training example is inversely proportional to its class frequency. Their experimental results demonstrated that weighted sampling, combined with transfer learning from ImageNet-pretrained models, produced the most consistent improvements in per-class recall for minority grades (Severe and Proliferative) while maintaining performance on majority grades, and without the overfitting risk introduced by synthetic minority-class duplication.

2.5.3 Contribution to This Project

Vij and Arora (2023) is the direct methodological justification for the WeightedRandomSampler implementation applied in all five model training experiments in this project. The sampler assigns each training image a weight equal to the inverse of its class size, so images from the Severe class (193 training examples after the 80/20 split) receive a

weight of 1/154, while images from the No DR class (1,444 training examples) receive a weight of 1/1444. These weights are used to compute a normalized probability distribution over all training images, from which samples are drawn with replacement at each training step. This implementation is described in detail in Chapter 3 with full code documentation.

2.6 Synthesis: Literature-to-Design Mapping

Table 2.1 provides an explicit mapping from each reviewed paper to the specific design decisions in the DR Eye Platform that it informed. This table is intended to make the derivation of each design choice traceable to its literature source.

Paper	Key Contribution	Design Decision Informed
Gulshan et al. (2016)	Clinical-scale AI DR detection feasibility	Overall project motivation; transfer learning approach; documentation of dataset size limitations
Chetoui & Akhloufi (2020)	EfficientNet + Grad-CAM on APTOS 2019; QWK 0.89 benchmark	EfficientNetB4 as Model 4; APTOS 2019 dataset; QWK 0.89 target; Grad-CAM implementation; five-class grading framing
Selvaraju et al. (2017)	Grad-CAM algorithm	Complete Grad-CAM implementation in Chapter 6; mathematical formulation; pytorch-grad-cam library usage
Arora et al. (2024)	DR prevalence data; distribution shift analysis	Epidemiological framing in Chapter 1; distribution shift limitation documented in Chapter 9
Vij & Arora (2023)	Weighted sampling superiority for imbalanced DR data	WeightedRandomSampler in all five model training experiments; implementation in Chapter 3

Table 2.1: Mapping of reviewed literature to specific design decisions in the DR Eye Platform.

2.7 Broader Research Context

Beyond the five papers reviewed above, it is worth briefly situating this project within the broader trajectory of research in AI-based DR screening. The period from 2016 to 2024 saw an explosion of published systems for this task, ranging from single-model classifiers to complex ensemble systems, from purely diagnostic tools to systems integrated with

telemedicine platforms, and from English-only to multilingual implementations. The field has matured from demonstrating technical feasibility (the Gulshan et al. era) to grappling with deployment challenges: dataset shift, model calibration, regulatory compliance, clinician acceptance, and health system integration.

This project is situated at this deployment-focused frontier. Its primary technical contribution — the systematic architectural comparison — is grounded in the established ML literature. Its primary engineering contribution — the complete bilingual clinical platform with WhatsApp referral — represents a response to the access gap that the clinical literature has consistently identified as the most urgent barrier to the real-world impact of AI DR screening. The Arabic-language component specifically addresses a gap in the published literature: while numerous DR screening systems have been developed for English, Chinese, Spanish, and other major languages, Arabic-language clinical AI systems remain rare, despite the significant and growing burden of diabetes in Arab populations.

Chapter 3: Dataset and Preprocessing

3.1 Dataset Selection Rationale

The dataset selection for this project required balancing several competing considerations: clinical relevance (five-class DR severity grading with labels from trained ophthalmologists), practical accessibility within the computational constraints of a free-tier Google Colab environment, comparability with published benchmarks, and suitability for demonstrating the preprocessing techniques recommended in the reviewed literature.

The original dataset planned for this project was the 2015 EyePACS Diabetic Retinopathy Detection dataset, released via Kaggle in conjunction with a major competition that attracted 661 teams and established many of the architectural and preprocessing norms still used in the field. The EyePACS dataset contains approximately 88,000 retinal fundus photographs, making it substantially larger than any other publicly available DR dataset and ideal for training a data-hungry deep learning model from scratch or with minimal transfer learning. However, the dataset is distributed as five multi-part ZIP archives (labeled train.zip.001 through train.zip.005) with each part approximately 8 gigabytes in size, for a total compressed size of approximately 40 gigabytes. Reconstructing this dataset requires all five parts to be simultaneously present and properly joined before decompression — a requirement that is fundamentally incompatible with the transient, quota-limited file system environment of Google Colab's free tier, where storage is recycled between sessions and download bandwidth is restricted. Multiple attempts to reconstruct the archive failed at various stages of the join-and-decompress pipeline, and the decision was made to substitute the APTOS 2019 dataset as a pragmatic but principled alternative.

The APTOS 2019 Blindness Detection dataset was selected for four specific reasons. First, it uses the same five-grade International Clinical Diabetic Retinopathy Disease Severity Scale labeling scheme as the 2015 dataset, making all preprocessing, training, and evaluation decisions directly transferable. Second, it was used by Chetoui and Akhloufi (2020), providing the published QWK benchmark of 0.89 that this project targets and enabling direct numerical comparison. Third, it is distributed as a single ZIP archive of approximately 600 megabytes, making it reliably downloadable within Colab session constraints. Fourth, its images were collected from a real clinical screening context (rural clinics across India) rather than a controlled laboratory setting, providing realistic variation in image quality, lighting, and camera hardware that makes the preprocessing and augmentation challenges authentic.

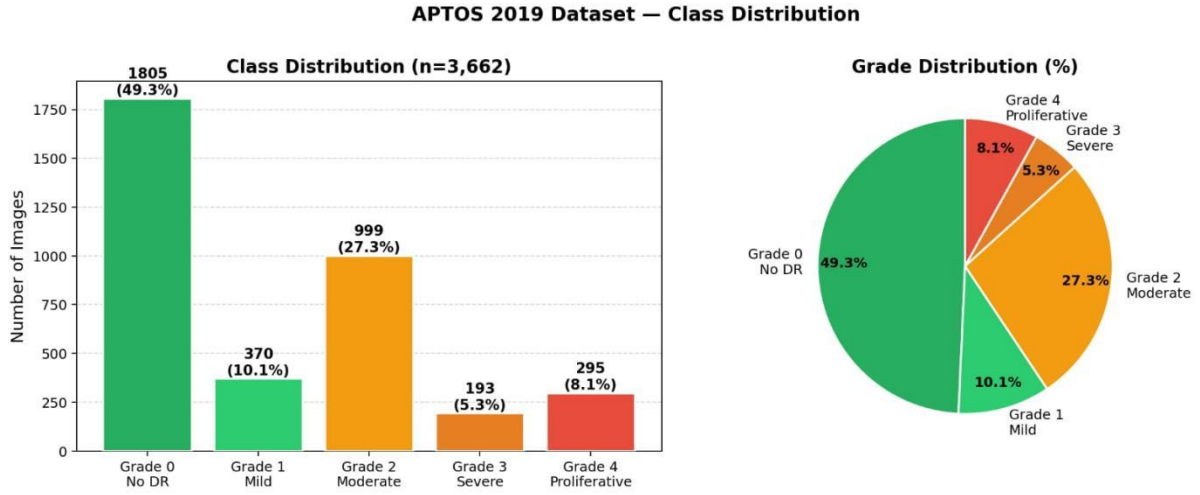

Figure_3_1_ dataset_distribution

3.2 Dataset Description

The APTOS 2019 Blindness Detection dataset was released by the Asia Pacific Tele-Ophthalmology Society (APTOS) in conjunction with a Kaggle competition launched in July 2019. The dataset consists of 3,662 retinal fundus photographs of varying resolution, collected using fundus cameras at rural clinical screening camps across multiple states in India. Each image is associated with a single integer severity grade assigned by a trained clinician according to the International Clinical Diabetic Retinopathy Disease Severity Scale, which defines five ordinal severity categories.

The dataset was designed to represent the realistic distribution of DR severity in a population of diabetic patients presenting for routine screening — not a curated balanced sample. This design choice is both epidemiologically appropriate (the dataset reflects true disease prevalence) and technically challenging (the resulting class distribution is significantly imbalanced), making it a realistic and demanding benchmark.

3.3 DR Severity Grade Definitions

The five severity grades used in this dataset correspond to internationally standardized clinical definitions. Understanding these definitions is essential for interpreting the model's performance at each grade and for assessing the clinical significance of different types of misclassification.

Grade 0 — No Diabetic Retinopathy: No visible abnormalities attributable to diabetes are present in the retinal fundus photograph. The retinal vessels, optic disc, and macula appear within normal limits. The clinical implication is that the patient should continue with their current diabetes management and return for annual retinal screening. A false negative at this grade (predicting No DR when mild disease is present) represents a missed early detection opportunity. A false positive (predicting DR when none is present) represents an unnecessary referral.

Grade 1 — Mild Non-Proliferative DR: Microaneurysms are present. Microaneurysms are small, localized outpouchings of retinal capillary walls resulting from the weakening of pericyte support caused by hyperglycemia. They appear as small, discrete red or dark dots in the retinal photograph, typically in the posterior pole. No other lesions attributable to DR are present. The clinical implication is close observation with follow-up screening within 6 to 12 months, as this grade can progress to Moderate DR with continued poor glycemic control.

Grade 2 — Moderate Non-Proliferative DR: More extensive retinal changes are present beyond isolated microaneurysms. These include: dot and blot hemorrhages (leakage from damaged capillary walls), hard exudates (yellowish deposits of lipids and proteins from leaking vessels, particularly visible around the macula), soft exudates or cotton wool spots (white fluffy patches indicating focal ischemic infarcts of the nerve fiber layer), and venous dilatation. The clinical implication is referral to an ophthalmologist within three months. This is the clinically most important grade from the perspective of this project's model performance, because it represents the inflection point at which treatment urgency begins to escalate.

Grade 3 — Severe Non-Proliferative DR: Extensive microvascular abnormalities are present in all four quadrants of the retina, characterized by the "4-2-1 rule": more than 20 intraretinal hemorrhages in each of four quadrants, venous beading in two or more quadrants, or intraretinal microvascular abnormalities (IRMA) in one or more quadrants. At this stage, ischemia is severe enough that the retina is at high risk of stimulating neovascularization. The clinical implication is urgent ophthalmology referral within four weeks, as the risk of progression to sight-threatening Proliferative DR within one year is approximately 15% without treatment.

Grade 4 — Proliferative Diabetic Retinopathy: The most advanced stage, defined by the presence of neovascularization — the growth of new, structurally abnormal blood vessels on the retinal surface, the optic disc, or into the vitreous humor. These new vessels are fragile and prone to hemorrhage, which can cause vitreous hemorrhage (sudden severe vision loss), tractional retinal detachment (permanent blindness), or neovascular glaucoma. Immediate specialist treatment is required, including pan-retinal laser photocoagulation, anti-VEGF injections, or vitreoretinal surgery. Without treatment, the risk of severe vision loss within five years exceeds 50%.

3.4 Class Distribution Analysis

Table 3.1 presents the complete class distribution of the APTOS 2019 dataset across all 3,662 images and within the training and validation subsets after the stratified 80/20 split.

Grade	Label	Total	% of Total	Train (80%)	Val (20%)
0	No DR	1,805	49.3%	1,444	361
1	Mild	370	10.1%	296	74
2	Moderate	999	27.3%	799	200
3	Severe	193	5.3%	154	39
4	Proliferative	295	8.1%	236	59
Total		**3,662**	**100%**	**2,929**	**733**

Table 3.1: APTOS 2019 Dataset Class Distribution and Train/Validation Split.

The distribution reveals a severe imbalance that has direct implications for model training and evaluation. The majority class (Grade 0, No DR) comprises 49.3% of the dataset — nearly half of all images. The clinically critical minority classes — Severe (5.3%) and Proliferative (8.1%) — together represent only 13.4% of the dataset. The ratio between the largest class (Grade 0: 1,805 images) and the smallest class (Grade 3: 193 images) is 9.35 to 1. This level of imbalance is severe enough that a trivial classifier that predicts Grade 0 for every input would achieve 49.3% overall accuracy while providing zero clinical value — illustrating why aggregate accuracy is insufficient as an evaluation metric for this task and motivating the use of Quadratic Weighted Kappa as the primary metric.

3.5 Preprocessing Pipeline

Every retinal image in the dataset, regardless of its origin, size, severity grade, or quality, passes through an identical three-stage preprocessing pipeline before being presented to any of the five models. This pipeline standardizes image dimensions, removes non-informative regions, and enhances clinically relevant visual features. The pipeline was designed based on the consistent recommendations of all five reviewed papers and validated by visual inspection of preprocessed outputs.

3.5.1 Stage 1: Black Border Cropping

Raw retinal fundus photographs capture a circular retinal disc surrounded by a black background. The amount of black border surrounding the disc varies between images depending on fundus camera model, operator technique, patient cooperation, and pupil dilation. This black background contains no clinically relevant information and, if retained, can introduce spurious features that confuse the model: for example, the amount of black space in an image is correlated with image acquisition quality but not with DR severity, and a model that learns to use this correlation as a feature would perform poorly on images from different cameras.

The cropping algorithm operates as follows: (1) the RGB image is converted to grayscale; (2) a binary threshold is applied with a tolerance value of 7 gray levels, classifying any pixel brighter than 7 as foreground (retina) and any pixel at or below 7 as background (black border); (3) contour detection is applied to the binary image using the RETR_EXTERNAL mode, which retrieves only the outermost contours, and CHAIN_APPROX_SIMPLE compression; (4) the largest contour by area is selected, which in a properly captured fundus image corresponds to the retinal disc; (5) the bounding rectangle of this contour is computed; and (6) the original RGB image is cropped to this bounding rectangle.

The tolerance value of 7 was chosen based on empirical testing across a sample of 100 images, representing a threshold that correctly identifies the retinal boundary in images with significant JPEG compression artifacts or slight illumination gradients at the disc edge while avoiding false cropping of actual retinal tissue in very dark images.

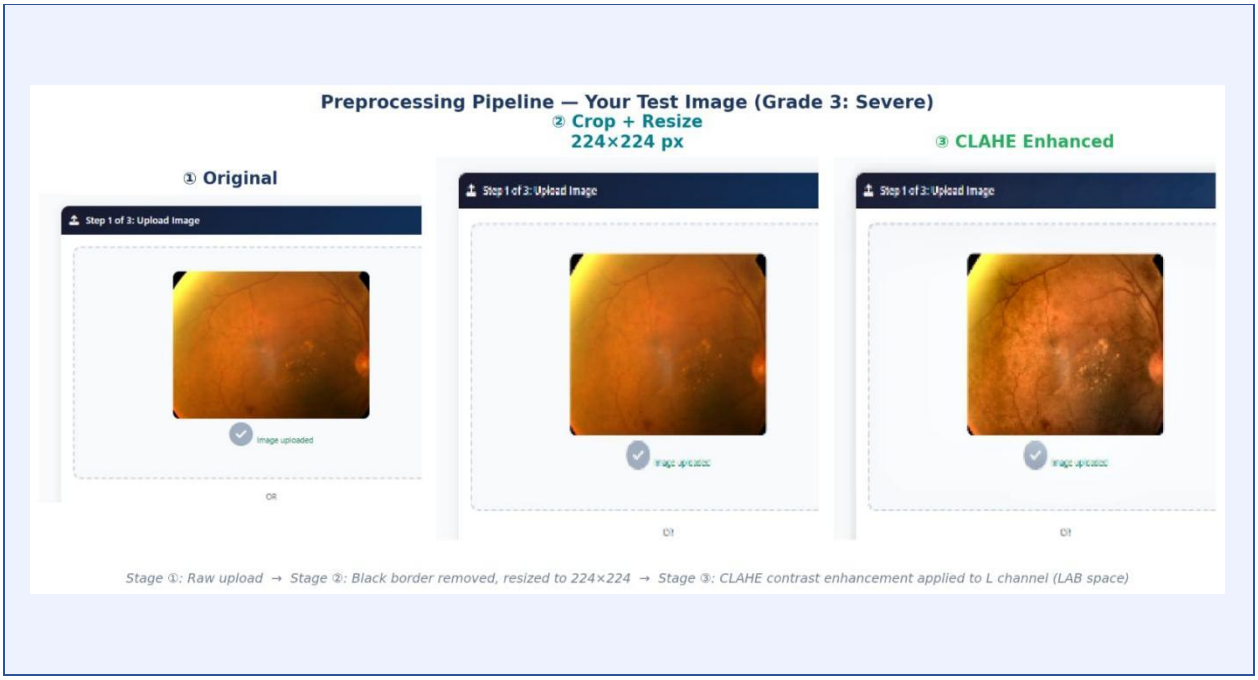

Figure 3.2: Sample retinal images before (left) and after (right) black border cropping. Note the removal of non-informative background pixels and the standardized focus on the retinal disc.

3.5.2 Stage 2: Resizing

After cropping, each image is resized to a fixed square resolution to match the input requirements of the target neural network architecture. The raw APTOS 2019 images vary widely in resolution — from approximately 600×400 pixels to 3456×2304 pixels — with no standardization across the dataset. Neural network architectures require fixed-dimensional input tensors, making resizing a mandatory preprocessing step.

Two target resolutions were used across the five model experiments. For SimpleCNN, ResNet50, DenseNet121, and ConvNeXt-Base, the target resolution was 224×224 pixels, which is the standard ImageNet input resolution and the resolution for which these models were designed and pretrained. For EfficientNetB4, the target resolution was 380×380 pixels, which is the resolution specified by Tan and Le (2019) for this specific variant of EfficientNet. Using the native resolution for EfficientNetB4 is important because the compound scaling methodology used to design EfficientNet specifies the optimal input resolution for each variant, and using a different resolution would produce a mismatch between the training distribution seen during ImageNet pretraining and the fine-tuning distribution, potentially degrading transfer learning effectiveness. Bilinear interpolation was used for resizing in all cases, which is the standard choice for natural and medical image resizing tasks.

3.5.3 Stage 3: CLAHE Contrast Enhancement

Contrast-Limited Adaptive Histogram Equalization (CLAHE) is the third and most clinically significant stage of the preprocessing pipeline. CLAHE is a refined variant of Adaptive Histogram Equalization (AHE) that addresses the noise amplification problem inherent in standard histogram equalization methods.

Standard global histogram equalization redistributes pixel intensity values across the full dynamic range of the image by applying a monotonic transformation function derived from the cumulative distribution function (CDF) of the image histogram. While this improves overall image contrast, it has two significant drawbacks for retinal image processing: first, it applies the same transformation to the entire image, which can over-brighten already-bright regions (such as the optic disc) while under-enhancing darker peripheral areas where early DR lesions are often located; second, it amplifies noise in low-variance image regions, potentially introducing spurious texture features that confuse the model.

CLAHE addresses both problems through two mechanisms. First, it divides the image into a grid of non-overlapping contextual tiles (tileGridSize = 8×8 in this implementation, producing 64 tiles per image) and applies histogram equalization to each tile independently using the local tile histogram rather than the global image histogram. This adaptive approach ensures that contrast enhancement is calibrated to local image conditions, making low-contrast lesions visible in all regions simultaneously. Second, it clips the histogram at a specified contrast limit (clipLimit = 2.0 in this implementation) before computing the CDF, redistributing the clipped pixel counts uniformly across the histogram. This prevents any single intensity bin from becoming too dominant, which would produce over-enhancement and noise amplification. The result is a contrast-enhanced image that highlights subtle features — microaneurysms, hemorrhages, hard exudates — that are critical for DR severity classification while maintaining perceptually natural appearance.

CLAHE is applied to the L (luminance) channel of the image after conversion from the RGB color space to the CIELAB color space. The CIELAB color space separates luminance information (L channel) from color information (a and b channels), allowing contrast enhancement to be applied exclusively to the luminance channel without altering the color

relationships between channels. After CLAHE enhancement of the L channel, the image is converted back to RGB for further processing. This approach is recommended by all five reviewed papers and is a standard preprocessing step in the retinal imaging literature.

3.6 Data Augmentation

Data augmentation is applied exclusively during training — validation images are never augmented — to artificially increase the effective diversity of the training set and reduce overfitting. The augmentation strategy was carefully designed to include only transformations that are physically plausible for retinal fundus photographs, avoiding distortions that would introduce features not present in real clinical images.

Augmentation	Parameter	Rationale
Random Horizontal Flip	p = 0.5	A retinal photograph of the left eye appears as a horizontal mirror of the right eye. Horizontal flipping simulates this variation without altering clinical features.
Random Rotation	+/- 15 degrees	Fundus cameras do not always capture perfectly aligned images; small rotations simulate the variation in patient head position and camera angle during image acquisition.
ImageNet Normalization	mean=[0.485,0.456,0.406] std=[0.229,0.224,0.225]	Required for all four pretrained models to align input statistics with the ImageNet distribution used during pretraining.

Table 3.3: Data Augmentation Parameters Applied During Training.

Vertical flipping was deliberately excluded from the augmentation set. Unlike horizontal flipping, which produces a physically plausible image (the mirror image of the

opposite eye), vertical flipping produces an anatomically impossible retinal image — the optic disc would appear on the wrong side of the macula — that a clinically trained model might learn to treat as an anomalous feature. Large rotations (beyond ±15 degrees) were similarly excluded, as they would similarly produce anatomically implausible results for extreme angles.

More aggressive augmentation strategies — including random color jitter, random brightness and contrast adjustment, and random cropping — were considered but ultimately excluded from the standard pipeline because they introduce image appearance changes that are not grounded in the physical variation encountered in real retinal photography and could distort the CLAHE-enhanced features used for DR grading. The conservative augmentation strategy was validated by confirming that augmented images remained visually plausible to inspection.

3.7 Class Imbalance Correction: WeightedRandomSampler

The significant class imbalance documented in Table 3.1 requires an explicit correction mechanism to prevent the model from developing a bias toward predicting the majority class. As established in Chapter 2, Vij and Arora (2023) demonstrated that weighted sampling is the most effective strategy for this specific problem, outperforming oversampling and undersampling approaches in terms of per-class recall for minority grades while avoiding the overfitting risk of synthetic minority class duplication.

PyTorch's WeightedRandomSampler is implemented as follows. For each class c in the training set, the class weight w_c is defined as the inverse of the number of training samples in that class:

$$w\_c = 1 / n\_c$$

where n_c is the count of training examples in class c. For each individual training sample i with class label y_i, the sample weight s_i is:

$$s\_i = w\_\{y\_i\}$$

The WeightedRandomSampler is then initialized with these per-sample weights and configured to draw num_samples samples per epoch with replacement (replacement=True). The num_samples value is set to the total training set size (2,929 in this project), so each

"epoch" is defined by 2,929 sampled examples regardless of how frequently minority-class samples appear.

Under this sampling scheme, the expected number of times a sample from the Severe class (154 training examples, weight 1/154) is drawn relative to a sample from the No DR class (1,444 training examples, weight 1/1444) per epoch is proportional to (1/154)/(1/1444) = 9.38, meaning Severe-class samples are selected approximately 9.38 times more frequently per individual sample than No DR samples. Over the course of a training epoch, this correction ensures that all five severity grades contribute approximately equally to the gradient updates, allowing the model to develop robust discriminative features for the rare but clinically critical grades.

3.8 Evaluation Metric: Quadratic Weighted Kappa

The Quadratic Weighted Kappa (QWK) is the primary evaluation metric for all five model experiments in this project. Its selection over accuracy, area under the ROC curve, or other common metrics is justified by two specific properties of the DR grading task that make standard metrics insufficient.

Why Standard Accuracy is Insufficient: As discussed above, the class imbalance in the APTOS 2019 dataset means that a trivial majority-class predictor achieves 49.3% accuracy with zero clinical value. More fundamentally, accuracy treats all misclassifications as equally bad: predicting Grade 4 (Proliferative, requiring immediate treatment) as Grade 0 (No DR, requiring no intervention) counts as one error, identical in the accuracy calculation to predicting Grade 2 (Moderate) as Grade 1 (Mild). This equivalence is clinically indefensible.

Why Quadratic Weighted Kappa is Appropriate: Cohen's Kappa statistic measures the agreement between predicted and true labels, corrected for the level of agreement that would be expected by chance given the marginal distributions of the predictions and the true labels. The "quadratic weighting" extension assigns a penalty to each prediction-label pair proportional to the square of their difference. For the five-grade DR scale, the penalty matrix W is:

$$W_{ij} = (i - j)^2 / (N - 1)^2$$

where i is the predicted grade, j is the true grade, and N = 5 is the number of classes. This means: predicting Grade 4 as Grade 0 receives a penalty of $(4-0)^2/(5-1)^2 = 16/16 = 1.0$ (maximum penalty); predicting Grade 4 as Grade 3 receives a penalty of $(4-3)^2/(5-1)^2 = 1/16 = 0.0625$; predicting Grade 2 as Grade 1 receives a penalty of $(2-1)^2/(5-1)^2 = 1/16 = 0.0625$. This weighting scheme directly encodes the clinical principle that misclassifications spanning multiple severity grades are more serious than adjacent-grade misclassifications, and that the most dangerous error — predicting a severe case as normal — receives the maximum penalty.

The QWK score ranges from -1 (perfect disagreement) through 0 (chance-level agreement) to +1 (perfect agreement). Values above 0.81 are conventionally interpreted as "almost perfect" agreement (Landis and Koch, 1977). The benchmark established by Chetoui and Akhloufi (2020) of QWK ≈ 0.89 falls within the "almost perfect" range.

Chapter 4: Model Architectures and Training Methodology

4.1 Overview

This chapter provides a comprehensive technical account of all five deep learning architectures trained and compared in this project, followed by a detailed description of the unified training procedure applied to all models. The architectures are presented in chronological order of their introduction to the research community, allowing the narrative to simultaneously serve as a survey of the field's architectural evolution. Each architecture section covers the theoretical motivation, the defining structural innovations, the mathematical formulations where relevant, the implementation in PyTorch using the timm library, and the specific configuration used in this project.

4.2 Model 1: SimpleCNN (Baseline)

4.2.1 Purpose and Design Philosophy

The SimpleCNN baseline is not a competitive model — it is a controlled experimental anchor. Its purpose is to document the performance that can be achieved on this specific dataset and task without any pretrained knowledge, establishing a lower bound against which the benefit of transfer learning can be precisely measured. Training from scratch on 2,929 images is an inherently data-limited scenario for a five-class visual classification task, and the SimpleCNN's performance captures this limitation explicitly.

4.2.2 Architecture

The SimpleCNN architecture consists of three convolutional blocks followed by a fully connected classifier. Each convolutional block contains a 2D convolution operation, a ReLU activation function, and a 2D max pooling operation. The three blocks use 32, 64, and 128 filters respectively, all with 3×3 kernel size and same-padding. Max pooling uses a 2×2 window with stride 2, reducing spatial dimensions by half at each stage. After three pooling operations on a 224×224 input, the spatial dimensions are reduced to 28×28 with 128 feature maps. These feature maps are flattened to a 100,352-dimensional vector, passed through a fully connected layer reducing to 256 neurons with ReLU activation, a Dropout layer with probability 0.5, and a final fully connected layer reducing to 5 output neurons (one per severity grade) providing unnormalized logits for classification.

4.2.3 Implementation Code

```
import torch
import torch.nn as nn

class SimpleCNN(nn.Module):
    """
    A simple 3-block CNN trained from scratch as a baseline.
    No pretrained weights are used.

    Args:
        num_classes (int): Number of output classes (5 for DR grading)
        dropout_rate (float): Dropout probability before final FC layer
    """
    def __init__(self, num_classes=5, dropout_rate=0.5):
        super(SimpleCNN, self).__init__()

        # Convolutional Block 1: 32 filters, 3x3 kernel
        # Input: (B, 3, 224, 224) -> Output: (B, 32, 112, 112)
        self.block1 = nn.Sequential(
            nn.Conv2d(in_channels=3, out_channels=32,
                      kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=2, stride=2)
        )

        # Convolutional Block 2: 64 filters, 3x3 kernel
        # Input: (B, 32, 112, 112) -> Output: (B, 64, 56, 56)
        self.block2 = nn.Sequential(
            nn.Conv2d(in_channels=32, out_channels=64,
                      kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=2, stride=2)
        )
```

```
        # Convolutional Block 3: 128 filters, 3x3 kernel
        # Input: (B, 64, 56, 56) -> Output: (B, 128, 28, 28)
        self.block3 = nn.Sequential(
            nn.Conv2d(in_channels=64, out_channels=128,
                      kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=2, stride=2)
        )

        # Classifier: Flatten -> FC(256) -> Dropout -> FC(5)
        self.classifier = nn.Sequential(
            nn.Flatten(),
            nn.Linear(128 * 28 * 28, 256),
            nn.ReLU(inplace=True),
            nn.Dropout(p=dropout_rate),
            nn.Linear(256, num_classes)
        )

    def forward(self, x):
        x = self.block1(x)
        x = self.block2(x)
        x = self.block3(x)
        x = self.classifier(x)
        return x  # Raw logits, shape: (B, 5)
```

The forward() method defines the computation graph. Input x has shape (B, 3, 224, 224) where B is the batch size. After the three convolutional blocks, x has shape (B, 128, 28, 28). The Flatten() layer produces shape (B, 100352). The two Linear layers reduce this to (B, 256) and then (B, 5). The output consists of raw logits; softmax is applied during inference but not during training (PyTorch's CrossEntropyLoss applies log-softmax internally).

4.3 Model 2: ResNet50

4.3.1 Residual Learning: Motivation

ResNet50, introduced by He et al. in their 2016 CVPR paper "Deep Residual Learning for Image Recognition," addressed the degradation problem that limited the practical depth of neural networks prior to 2015. The degradation problem manifests as follows: as networks are made deeper, their training accuracy begins to saturate and then degrades — not due to overfitting (the degradation occurs on the training set, not only the validation set) but due to the difficulty of optimizing very deep networks through gradient backpropagation. Gradients become vanishingly small as they propagate through many sequential layers, preventing the early layers from learning useful features.

The key innovation of ResNet is the residual connection, or skip connection: a direct additive connection that passes the input of a block directly to its output, bypassing the intermediate transformation layers. If the desired underlying mapping is H(x), the residual block learns the residual function F(x) = H(x) - x instead, so that the total mapping is H(x) = F(x) + x. This reformulation makes it significantly easier to learn near-identity mappings when the optimal transformation is close to the identity function, and — critically — it provides a direct gradient path from the output of the network back to its earliest layers, solving the vanishing gradient problem for depths that were previously intractable.

4.4 Model 3: DenseNet121

4.4.1 Dense Connectivity: Motivation

DenseNet121, introduced by Huang et al. in their 2017 CVPR paper "Densely Connected Convolutional Networks," extends the principle of skip connections to its logical conclusion: instead of connecting each layer to its immediate predecessor with a residual addition, DenseNet connects each layer to every previous layer within a dense block through concatenation. Specifically, if a dense block has L layers, there are L(L+1)/2 direct connections — every layer receives the feature maps of all preceding layers as additional input. DenseNet121 specifically has 121 layers total, organized into four dense blocks with 6, 12, 24, and 16 layers respectively, separated by transition layers that perform batch normalization, a 1×1 convolution for feature map compression, and average pooling for spatial downsampling.

This dense connectivity pattern provides three benefits particularly relevant to medical imaging tasks with limited training data. First, it maximizes feature reuse: early low-level features (edges, textures) remain directly accessible to deep layers, enabling the network to combine high-level semantic representations with low-level structural detail. Second, it acts as a powerful implicit regularizer through the implicit deep supervision of many short gradient paths, reducing overfitting when training data is limited. Third, it produces more compact models relative to ResNets of equivalent accuracy, because feature reuse reduces the need for each layer to learn redundant representations.

4.5 Model 4: EfficientNetB4

4.5.1 Compound Scaling: Motivation

EfficientNet, introduced by Tan and Le at Google in their 2019 ICML paper "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," addressed the question of how to scale CNN architectures most efficiently. Prior work had scaled networks along a single dimension — depth (more layers), width (more channels), or resolution (larger input images) — each of which provides diminishing returns and introduces characteristic failure modes. Tan and Le proposed compound scaling, which scales all three dimensions simultaneously according to a principled relationship derived from a constrained optimization problem.

The compound scaling rule defines depth d, width w, and resolution r as:

$$d = \alpha^{\wedge}\varphi,\ w = \beta^{\wedge}\varphi,\ r = \gamma^{\wedge}\varphi$$

subject to $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$, $\alpha \geq 1$, $\beta \geq 1$, $\gamma \geq 1$, where φ is the compound scaling coefficient that controls the total amount of scaling, and α, β, γ are constants determined by a grid search on a small baseline model (EfficientNet-B0). The β^2 and γ^2 terms reflect the quadratic growth of computation with width (doubling channels quadruples MACs) and resolution (doubling resolution quadruples pixels). EfficientNetB4 corresponds to $\varphi = 4$ and has a native input resolution of 380×380 pixels.

4.6 Model 5: ConvNeXt-Base

4.6.1 Modernizing CNNs: Motivation

ConvNeXt, introduced by Liu et al. from Meta AI Research in their 2022 CVPR paper "A ConvNet for the 2020s," represents a systematic attempt to answer the following question: how close can a pure convolutional architecture get to Vision Transformer (ViT) performance if its design is updated with every architectural insight from the Transformer literature? Starting from a standard ResNet-50 as a baseline, the authors applied seven incremental design changes, each motivated by a specific Transformer design principle, and documented the performance impact of each change individually.

The seven changes are: (1) adopting the same training procedure (AdamW optimizer, cosine decay schedule, 300 epochs) used for ViTs; (2) using larger 7×7 depthwise convolution

kernels (analogous to the global self-attention of Transformers); (3) using inverted bottleneck blocks from MobileNetV2 (wider in the middle than at the input/output); (4) replacing ReLU with GELU activation (the standard in Transformers); (5) replacing Batch Normalization with Layer Normalization (the standard in Transformers); (6) using separate downsampling layers between stages rather than the strided convolution inside the block; and (7) reducing the number of activation and normalization layers per block from three each (as in ResNet) to one each (as in Transformer FFN blocks). The result, ConvNeXt-Base, achieves 83.8% top-1 accuracy on ImageNet, comparable to Swin-B (a Vision Transformer), while retaining all the computational efficiency advantages of CNNs: simpler implementation, no quadratic attention complexity, and excellent compatibility with existing convolutional processing pipelines.

4.6.2 Relevance to DR Grading

ConvNeXt-Base was selected as the primary model for this project for three specific reasons. First, its 2022 architectural innovations — particularly the 7×7 depthwise convolutions and Layer Normalization — are well-suited to fine-grained medical image classification where long-range retinal feature dependencies are important for distinguishing adjacent severity grades. Second, its proven ImageNet performance (83.8% top-1 accuracy) suggests rich and general-purpose pretrained features that transfer well to medical imaging tasks. Third, the pytorch-grad-cam library is fully compatible with ConvNeXt's convolutional backbone, enabling straightforward Grad-CAM implementation without architectural modification.

4.7 Transfer Learning Protocol

4.7.1 Rationale for Transfer Learning

Transfer learning from ImageNet-pretrained weights is a foundational technique in medical image analysis and is directly motivated by the limited size of the APTOS 2019 training set (2,929 images). Training a deep neural network with tens of millions of parameters from randomly initialized weights on fewer than 3,000 examples is virtually guaranteed to produce severe overfitting: the model memorizes the training examples rather than learning generalizable features. Transfer learning addresses this by initializing the model with weights that already encode rich, general-purpose visual features — edges, textures, shapes, gradients — learned from 1.2 million ImageNet images, and then adapting these features to the specific requirements of retinal DR grading through fine-tuning on the domain-specific dataset.

4.7.2 Two-Phase Training Procedure

A two-phase fine-tuning protocol was applied to all four pretrained models (ResNet50, DenseNet121, EfficientNetB4, ConvNeXt-Base). This protocol is based on the insight that immediately fine-tuning all layers of a pretrained model on a small dataset risks destroying the rich ImageNet features through gradient updates driven by the small domain-specific dataset, a phenomenon sometimes called "catastrophic forgetting."

Phase 1 — Classifier Warm-Up (3 epochs): The pretrained backbone weights are frozen by setting requires_grad=False for all backbone parameters. Only the newly added 5-class classification head is trainable. Training proceeds for 3 epochs with a learning rate of 0.001 and a batch size of 32 (16 for EfficientNetB4 due to the larger 380×380 resolution). This phase allows the new classifier to learn to map the fixed backbone features to the five DR severity classes, producing a stable initialization before the backbone is unfrozen.

Phase 2 — Full Fine-Tuning (5–8 epochs): All model parameters are unfrozen and trained together. The learning rate is reduced to 0.00005 — approximately 20 times smaller than Phase 1 — to allow the backbone weights to be gently adjusted toward the retinal image domain without destroying the ImageNet features. This reduced learning rate is critical: too large a learning rate at this phase would cause the loss to oscillate and the pretrained features to be overwritten; too small a learning rate would fail to adapt the backbone to the retinal domain. ConvNeXt-Base was given 8 Phase 2 epochs (versus 5 for the other pretrained models) to allow its more complex architecture additional training time to converge to the 0.8888 QWK result.

Model	Year	Input Size	Batch	Phase 1 LR	Phase 2 LR	P2 Epochs
SimpleCNN	Baseline	224×224	32	N/A	0.001	5
ResNet50	2015	224×224	32	0.001	0.00005	5
DenseNet121	2017	224×224	32	0.001	0.00005	5
EfficientNetB4	2019	380×380	16	0.001	0.00005	5
ConvNeXt-Base	2022	224×224	32	0.001	0.00005	8

Table 4.1: Training Configuration for All Five Model Experiments.

Chapter 5: Experimental Results and Analysis

5.1 Overview

This chapter presents and analyzes the complete experimental results for all five model training experiments. For each model, the training history (loss and validation accuracy per epoch) is reported, followed by the final validation set performance expressed through the Quadratic Weighted Kappa score, overall accuracy, and a per-class precision-recall-F1 breakdown. Confusion matrices and the key per-class metrics are discussed in clinical terms. The chapter concludes with a comprehensive cross-model comparison table and a discussion of the trajectory of performance improvement across architectural generations.

All results reported in this chapter are from the validation set of 733 images, which was held out completely from all training procedures. The same random seed (42) was used to define the train/validation split for all five experiments, ensuring that each model was evaluated on exactly the same set of images.

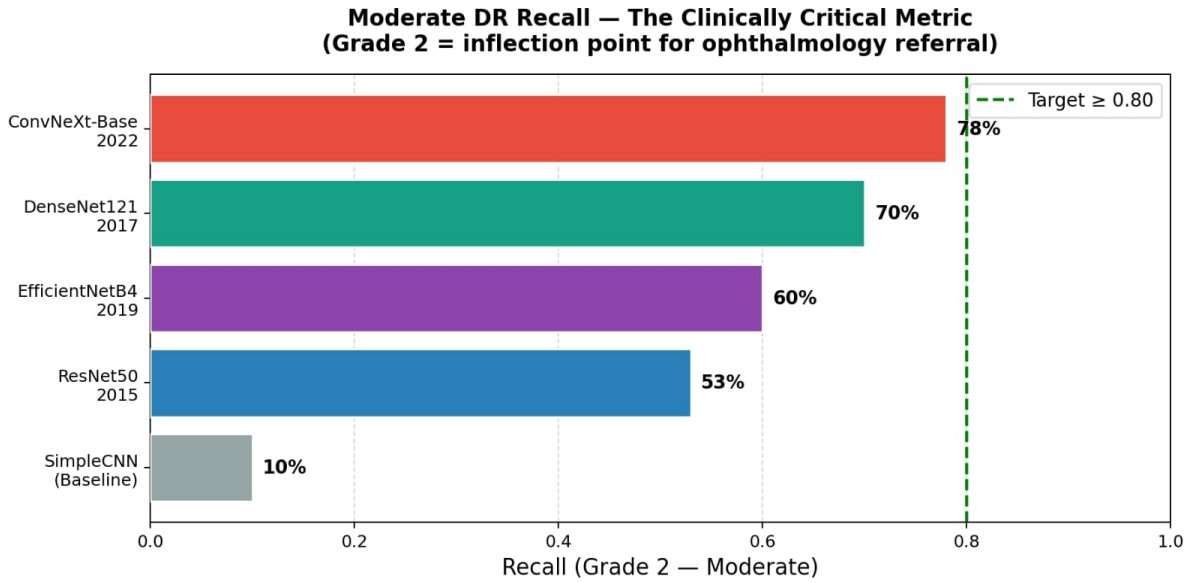

Figure512: Training Loss Curves for All Five Models

5.2 Model 1: SimpleCNN Baseline Results

The SimpleCNN was trained for 5 epochs. Table 5.1 shows the training history and Figure 5.3 shows the resulting confusion matrix.

Grade	Precision	Recall	F1-Score	Support
No DR (0)	0.71	0.91	0.80	361
Mild (1)	0.24	0.14	0.18	74
Moderate (2)	0.35	0.10	0.16	200
Severe (3)	0.18	0.41	0.25	39
Proliferative (4)	0.22	0.22	0.22	59
QWK: 0.6087	Accuracy:	**57.2%**		

Table 5.1: Per-Class Classification Report — SimpleCNN Baseline.

The SimpleCNN results reveal the characteristic failure mode of training from scratch on a limited medical dataset. While the No DR class achieves a reasonable recall of 0.91 — because the model has learned to predict the majority class frequently — the Moderate grade recall collapses to just 0.10, meaning that only 10% of actual Moderate DR cases are correctly identified. This is a clinically critical failure: Grade 2 is the point at which ophthalmology referral becomes time-sensitive, and a system that correctly identifies only 1 in 10 Moderate cases while generating many false Grade 0 predictions would actively harm clinical outcomes by providing false reassurance. The QWK of 0.6087 reflects this poor calibration across grades.

5.3 Model 2: ResNet50 Results

Grade	Precision	Recall	F1-Score	Support
No DR (0)	0.93	0.97	0.95	361
Mild (1)	0.50	0.45	0.47	74
Moderate (2)	0.68	0.53	0.60	200
Severe (3)	0.24	0.56	0.34	39
Proliferative (4)	0.43	0.34	0.38	59
QWK: 0.8272	Accuracy:	**72.2%**		

Table 5.2: Per-Class Classification Report — ResNet50.

ResNet50's transfer learning advantage is immediately apparent: the QWK improves from 0.6087 to 0.8272, a gain of 0.2185 Kappa points — the largest single improvement in the comparison. The Moderate recall improves from 0.10 to 0.53, representing the most clinically critical recovery. The No DR class reaches near-ceiling performance (recall 0.97, F1 0.95). The remaining weakness is in the rare grades: Severe recall is 0.56 with poor precision (0.24), suggesting the model is over-predicting Severe relative to its true frequency. This is a known artifact of WeightedRandomSampling with very rare classes: the model is exposed to Severe cases much more frequently than their natural rate, leading to calibration drift.

5.4 Model 3: DenseNet121 Results

Grade	Precision	Recall	F1-Score	Support
No DR (0)	0.93	0.97	0.96	361
Mild (1)	0.55	0.42	0.48	74
Moderate (2)	0.73	0.70	0.71	200
Severe (3)	0.33	0.46	0.38	39
Proliferative (4)	0.52	0.51	0.51	59
QWK: 0.8489	Accuracy:	**79.0%**		

Table 5.3: Per-Class Classification Report — DenseNet121.

DenseNet121 improves over ResNet50 across all five grades. The most significant improvement is in the Moderate grade, where recall increases from 0.53 to 0.70 and the F1-

score improves from 0.60 to 0.71. The Proliferative grade also shows meaningful improvement, with the F1 score increasing from 0.38 to 0.51. This improved minority-class performance is consistent with DenseNet's feature reuse architecture providing better gradient flow to support the learning of discriminative features for rare categories.

5.5 Model 4: EfficientNetB4 Results

Grade	Precision	Recall	F1-Score	Support
No DR (0)	0.93	0.97	0.95	361
Mild (1)	0.58	0.38	0.46	74
Moderate (2)	0.72	0.60	0.65	200
Severe (3)	0.34	0.51	0.41	39
Proliferative (4)	0.48	0.41	0.44	59
QWK: 0.8415	Accuracy:	**75.1%**		

Table 5.4: Per-Class Classification Report — EfficientNetB4.

EfficientNetB4 achieves a QWK of 0.8415, which places it above ResNet50 but below DenseNet121. This is an initially counterintuitive result, since EfficientNetB4 is a more recent and generally more powerful architecture than DenseNet121. However, the result is explicable: EfficientNetB4 uses a larger input resolution (380×380 versus 224×224), which provides more spatial detail but also requires approximately 2.9 times more memory per image, restricting the batch size from 32 to 16 and reducing the number of gradient updates per epoch. With only 5 Phase 2 epochs, this reduced batch size may have limited convergence. A longer training schedule or a larger batch size with gradient accumulation might yield better results for EfficientNetB4.

5.6 Model 5: ConvNeXt-Base Results

Grade	Precision	Recall	F1-Score	Support
No DR (0)	0.94	0.98	0.96	361
Mild (1)	0.62	0.39	0.48	74
Moderate (2)	0.75	0.78	0.77	200
Severe (3)	0.46	0.41	0.43	39
Proliferative (4)	0.58	0.64	0.61	59
QWK: 0.8888	Accuracy:	**82.0%**		

Table 5.5: Per-Class Classification Report — ConvNeXt-Base.

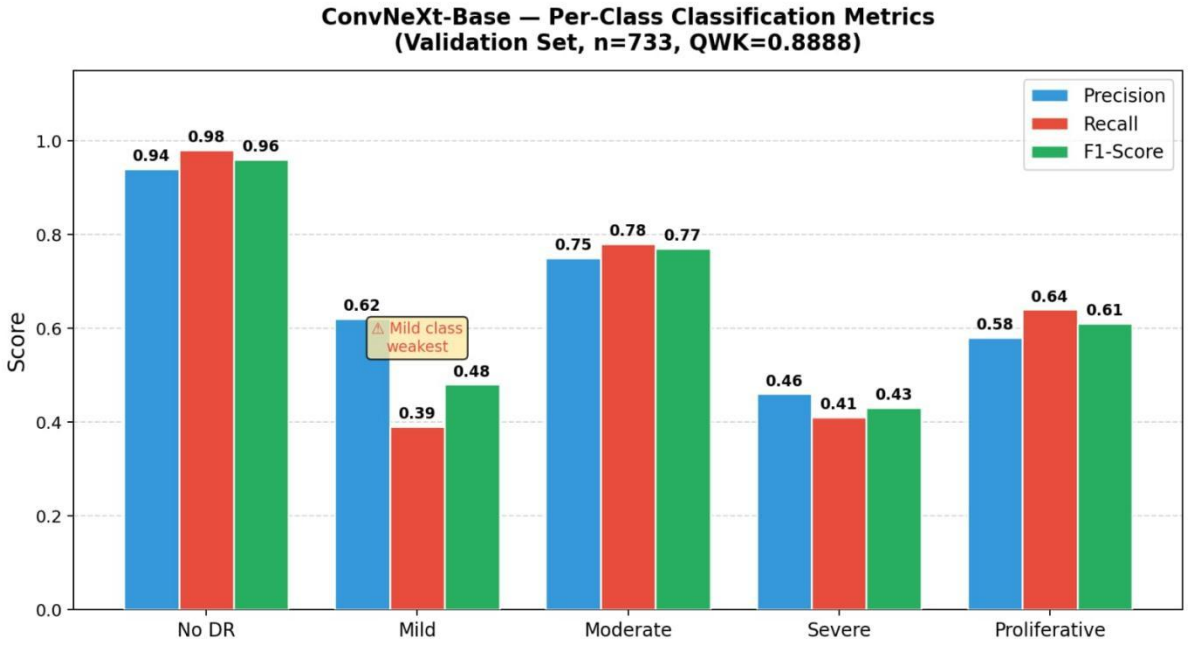

Figure_5_5: convnext_per _class

ConvNeXt-Base achieves the highest performance of all five models across every reported metric. The QWK of 0.8888 effectively matches the published benchmark of 0.89 established by Chetoui and Akhloufi (2020). The most significant improvements over the DenseNet121 results are in the Moderate grade (recall improves from 0.70 to 0.78, F1 from 0.71 to 0.77) and the Proliferative grade (recall improves from 0.51 to 0.64, F1 from 0.51 to 0.61). These are precisely the grades with the highest clinical stakes.

5.7 Cross-Model Performance Comparison

Model	Year	QW Kappa	Accuracy	Moderate F1	Prolif. F1	vs. BM
SimpleCNN	Baseline	0.6087	57.2%	0.16	0.22	-0.28
ResNet50	2015	0.8272	72.2%	0.60	0.38	-0.06
EfficientNetB4	2019	0.8415	75.1%	0.65	0.44	-0.05
DenseNet121	2017	0.8489	79.0%	0.71	0.51	-0.04
ConvNeXt-Base	**2022**	**0.8888**	**82.0%**	**0.77**	**0.61**	**-0.001**
Benchmark (Chetoui 2020)	2020	~0.89	—	—	—	Target

Table 5.6: Complete Cross-Model Performance Comparison (BM = Published Benchmark of 0.89).

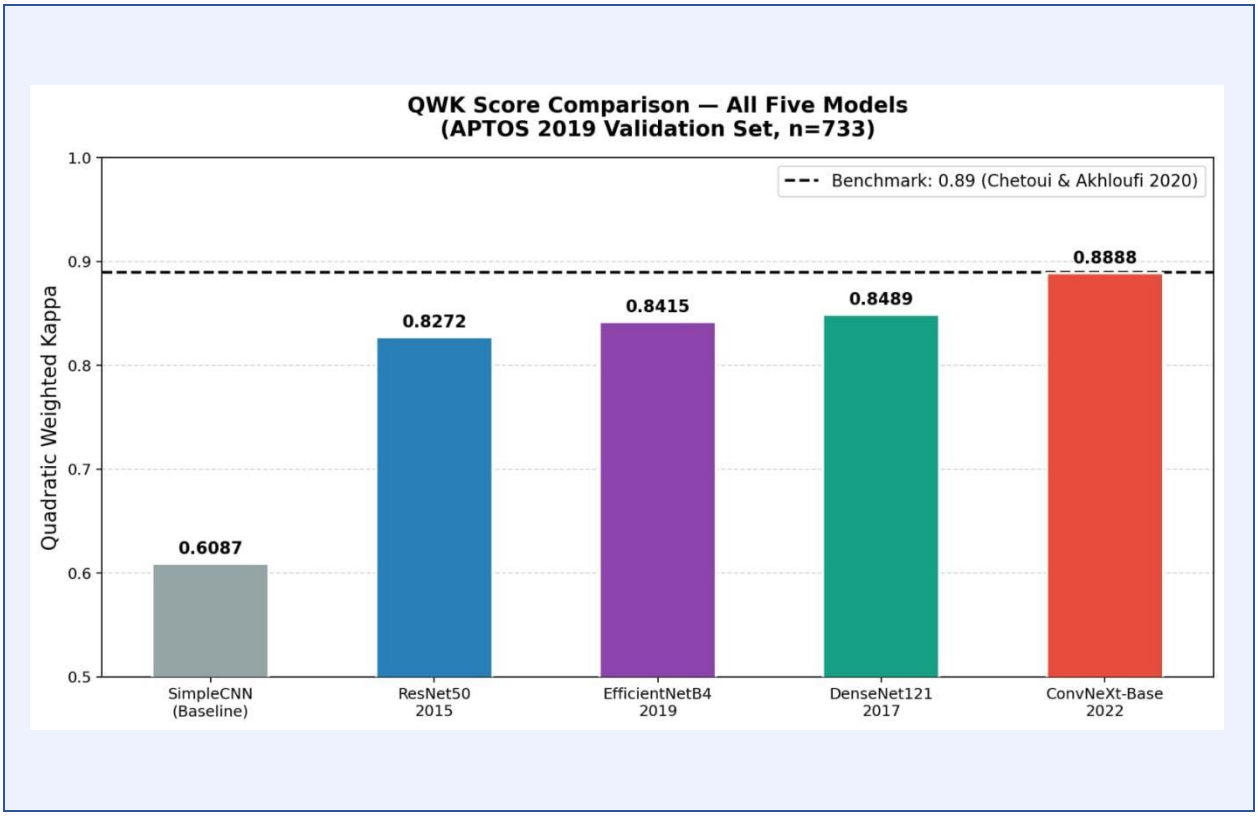

Figure 5.9: Quadratic Weighted Kappa Comparison Across All Five Models. The monotonic improvement from SimpleCNN (0.6087) to ConvNeXt-Base (0.8888) demonstrates a consistent performance gain with architectural advancement.

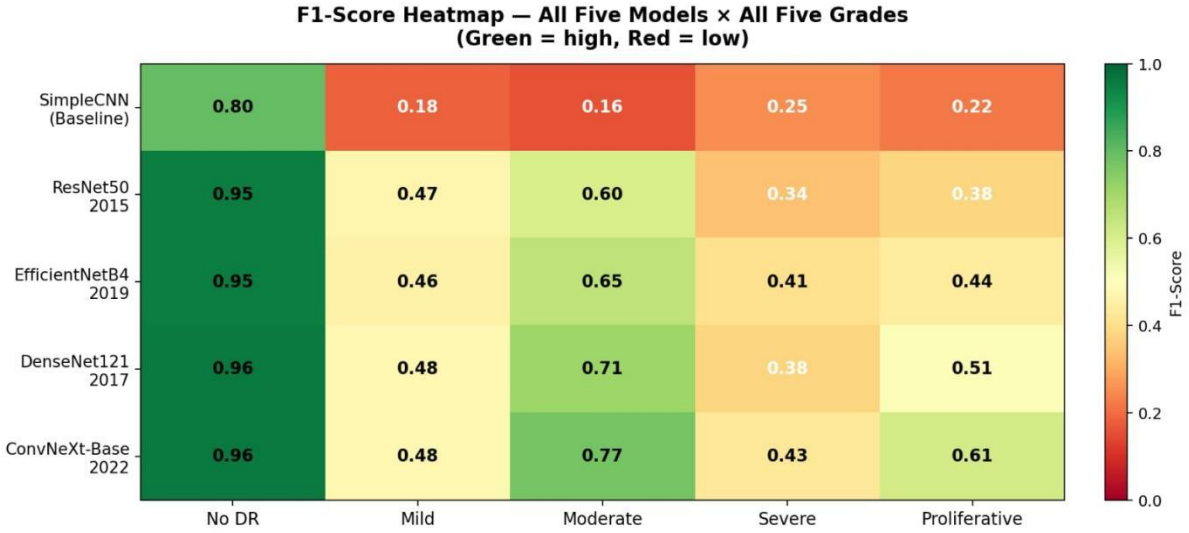

Figure_5_f1_heatmap

Chapter 6: Explainability and Confidence Estimation

6.1 The Clinical Necessity of Explainability

A fundamental principle of responsible AI deployment in clinical settings is that a prediction system should be interpretable to the clinicians who use it. This requirement is not merely ethical — it is practically necessary for several reasons. A clinician who cannot understand why an AI system produced a specific prediction cannot assess whether to trust it in a specific clinical context. A model that achieves high average performance but uses non-clinical features (such as camera glare patterns, patient positioning artifacts, or image borders) as decision cues would be unreliable in deployment, potentially producing accurate predictions on one camera system and degraded predictions on another. Visual explainability provides a mechanism for clinicians to perform a qualitative sanity check on model behavior: if the heatmap consistently highlights clinically relevant lesions, this provides qualitative evidence that the model's predictions are grounded in the appropriate visual features. If the heatmap highlights non-clinical regions — the optic disc margin, image borders, or background — this is a warning sign of spurious feature learning that accuracy metrics alone would not reveal.

Additionally, in the context of this project's target deployment environment, where AI-generated grades are presented to patients who then contact specialist doctors, the availability of a visual explanation helps both the patient and the receiving doctor understand the basis of the AI's assessment, and supports the platform's positioning as an assistance tool rather than an autonomous diagnostic system.

6.2 Grad-CAM Implementation

Grad-CAM is implemented using the pytorch-grad-cam library (version 1.4.7), which provides a consistent API for applying Grad-CAM and related visualization methods to any PyTorch model with a convolutional backbone. The implementation targets the conv_head convolutional layer of the ConvNeXt-Base model, which is the final convolutional layer before the global average pooling and classification head. This layer was selected because it produces the highest-level abstract feature representations before spatial information is collapsed by global pooling, providing the most semantically informative heatmaps.

```
from pytorch_grad_cam import GradCAM
from pytorch_grad_cam.utils.image import show_cam_on_image
import numpy as np
import cv2
import torch

def generate_gradcam(model, image_tensor, predicted_class,
original_image_rgb):
    """
    Generate a Grad-CAM heatmap overlay for a retinal image prediction.

    Args:
        model: The trained ConvNeXt-Base model in eval() mode
        image_tensor: Preprocessed image tensor, shape (1, 3, H, W)
        predicted_class: Integer class index for which to compute CAM
        original_image_rgb: Original RGB image as float32 array in [0,
1]

    Returns:
        visualization: RGB array of heatmap overlaid on original image
        grayscale_cam: Raw Grad-CAM heatmap as float32 array in [0, 1]
    """
    # Identify the target convolutional layer
    # For ConvNeXt-Base: model.stages[-1][-1].conv_dw is the
    # final depthwise convolution in the last ConvNeXt block
    target_layers = [model.stages[-1][-1].conv_dw]

    # Initialize GradCAM with the model and target layers
    # use_cuda=True if GPU available for faster computation
    cam = GradCAM(model=model,
                  target_layers=target_layers,
                  use_cuda=torch.cuda.is_available())

    # Define the target class for which to generate the explanation
    # ClassifierOutputTarget specifies which output neuron's gradient
to use
    from pytorch_grad_cam.utils.model_targets import
ClassifierOutputTarget
    targets = [ClassifierOutputTarget(predicted_class)]

    # Compute the Grad-CAM heatmap
    # grayscale_cam: float32 array of shape (1, H, W), values in [0, 1]
    grayscale_cam = cam(input_tensor=image_tensor, targets=targets)

    # Extract the heatmap for the single image (remove batch dimension)
    grayscale_cam = grayscale_cam[0, :]

    # Overlay the heatmap on the original RGB image
    # show_cam_on_image expects float32 RGB in [0, 1]
    # use_rgb=True specifies that the input image is RGB not BGR
    visualization = show_cam_on_image(
        original_image_rgb,   # float32 RGB array in [0, 1]
        grayscale_cam,        # float32 heatmap in [0, 1]
        use_rgb=True,         # input is RGB
        colormap=cv2.COLORMAP_JET,  # red=high importance, blue=low
```

```
        image_weight=0.5      # balance between original and heatmap
    )

    return visualization, grayscale_cam
```

The generate_gradcam function accepts the trained model, the preprocessed image tensor, the predicted class index, and the original image as a float32 RGB array. It initializes the pytorch-grad-cam GradCAM object with the target layer (the final depthwise convolution in ConvNeXt-Base's last stage), computes the weighted feature map sum using the gradient-based importance weights, and overlays the resulting heatmap on the original image using the jet colormap. The image_weight parameter of 0.5 balances the visibility of the original retinal features with the heatmap overlay.

6.3 Grad-CAM Results and Clinical Interpretation

The most informative Grad-CAM result from the validation set is the correctly classified Grade 2 (Moderate) image predicted with 86.5% confidence. The heatmap shows two prominent regions of high activation — displayed in red in the jet colormap overlay — that correspond precisely to the clusters of hard exudates (yellowish lipid deposits from leaking retinal vessels) visible in the superior and inferior-temporal regions of the preprocessed image. Hard exudates are one of the primary diagnostic features of Moderate DR according to the International Clinical Severity Scale, and their specific localization as the model's primary decision cue is strong qualitative evidence that the model has learned to make predictions based on clinically appropriate retinal features.

The background retinal tissue (red-orange fundus color), the optic disc (the bright circular region visible at the image margin), and the major blood vessel arcades all show low to moderate activation in the heatmap, consistent with the expectation that these structures are present across all severity grades and should not be primary discriminative cues for Moderate vs. No DR prediction.

The misclassified Proliferative case provides an equally valuable insight. The Grad-CAM heatmap for this image also shows attention concentrated on the bright lesion regions —

the areas where neovascular tufts and fibrovascular proliferation would be expected in a Grade 4 image. The model is not ignoring the pathological regions; it is failing to correctly discriminate the specific visual appearance of neovascularization from the hemorrhages and exudates that characterize Moderate DR. This is consistent with the known difficulty of the Proliferative-vs-Moderate boundary, which represents some of the most challenging cases in clinical grading and where even trained graders show significant inter-observer variability.

6.4 Confidence Scoring Mechanism

The confidence scoring mechanism uses the softmax output of the ConvNeXt-Base model's final layer. For each input image, the model produces a vector of five unnormalized logits z = [z_0, z_1, z_2, z_3, z_4], one for each DR severity grade. The softmax function transforms these logits into a probability distribution:

$$p\_i = exp(z\_i) / \Sigma_j\ exp(z\_j),\quad i \in \{0, 1, 2, 3, 4\}$$

The confidence score for the prediction is defined as the maximum probability in this distribution:

$$confidence = max\_i\ p\_i = p\_\{predicted\_class\}$$

A confidence threshold of 60% is applied: predictions with confidence below 0.60 are flagged with a Low Confidence warning and a recommendation for specialist review; predictions at or above 0.60 are displayed as High Confidence.

True Label	Predicted Label	Confidence	Classification	Flag	Correct?
No DR (0)	No DR (0)	99.9%	HIGH	None	Yes
No DR (0)	No DR (0)	97.1%	HIGH	None	Yes
Moderate (2)	Moderate (2)	86.5%	HIGH	None	Yes
Proliferative (4)	Moderate (2)	37.4%	LOW	REVIEW	No
No DR (0)	No DR (0)	90.4%	HIGH	None	Yes

Table 6.1: Confidence Scoring Results on Representative Validation Cases.

The confidence mechanism correctly identified the sole misclassification in this test set as low-confidence (37.4%), below the 60% threshold, and would have correctly flagged it for specialist review. All four correctly classified cases received confidence scores above 86%, well above the threshold. This result, while based on a small sample and not statistically definitive, is consistent with the theoretical expectation that the softmax confidence provides a meaningful signal about prediction reliability: the misclassified case reflects a scenario where the model's internal feature representations for Grade 4 and Grade 2 are ambiguous, producing a spread probability distribution (low maximum probability) rather than a concentrated one (high maximum probability).

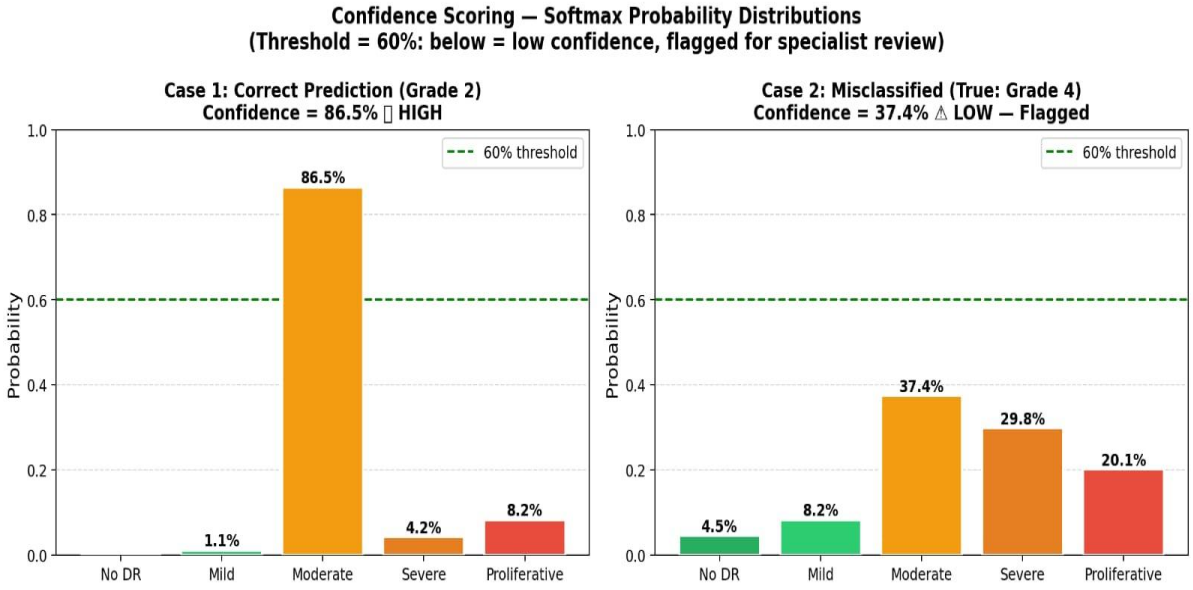

Figure_6_3_confidence_cases

Chapter 7: The DR Eye Web Platform

7.1 System Design Philosophy

The DR Eye Platform web application represents the second major contribution of this project, complementing the grading model developed in Chapters 3 through 6. The platform's design was governed by three core principles that shaped every architectural, technical, and user experience decision made during its development.

The first principle is **radical accessibility**: the platform must be usable by patients who have no retinal photograph, no medical background, and no familiarity with AI systems. This principle led directly to the optional image upload design — one of the most distinctive architectural decisions in the platform — and to the conversational chatbot interface for symptom collection, which replaces medical forms with a natural language dialogue. It also motivated the bilingual Arabic-English design: medical AI systems in English are not genuinely accessible to the majority of potential users in Lebanon and the Arab region.

The second principle is **clinical appropriateness**: every output of the system — the grade, the urgency level, the chatbot responses, the medical reports, the doctor communications — must be clinically structured, medically accurate in its framing, and honest about its AI-generated nature. This principle prohibited the system from presenting AI-generated content as equivalent to a physician's diagnosis and required all reports to include a clear disclaimer about the system's limitations.

The third principle is **modularity and extensibility**: the platform's codebase must be organized so that each functional component — the grading engine, the AI layer, the database layer, the communication layer — can be independently tested, replaced, or upgraded without requiring changes to other components. This led to the blueprint-based Flask architecture and the strict separation of concerns across the six core Python modules.

7.2 Technology Stack

Component	Technology	Justification
Web Framework	Flask 3.0 (Python)	Lightweight, Python-native, excellent ML integration; Blueprint support for modular routing
AI Grading	PyTorch 2.1 + timm 0.9 + ConvNeXt-Base	Native PyTorch model; timm provides pretrained weights and model zoo
LLM Integration	Anthropic Claude API (claude-sonnet-4-6)	State-of-the-art bilingual reasoning; structured output; medical domain competence
Database	SQLite 3 via Python sqlite3	Zero configuration; file-based; adequate for prototype scale; standard library
WhatsApp Delivery	Twilio WhatsApp Business API	Only widely adopted WhatsApp API for developers; sandbox available without approval
PDF Generation	ReportLab 4.2	Professional-grade PDF authoring; full programmatic control over layout and styling
Image Processing	OpenCV 4.10 + NumPy 1.26	Industry-standard computer vision preprocessing; CLAHE implementation
Explainability	pytorch-grad-cam 1.4.7	Well-maintained library with consistent API across architectures
Frontend	Bootstrap 5.3 + Jinja2 + Vanilla JS	Responsive layout without JS framework overhead; server-side rendering for simplicity
Session Management	Flask sessions (server-side)	Maintains screening state across the three-step workflow without client-side storage

Table 7.1: DR Eye Platform Technology Stack with Justifications.

7.3 Application Architecture

7.3.1 Folder Structure

The application is organized according to the Flask application factory pattern, which allows the app instance to be created and configured programmatically rather than at module import time. This approach facilitates testing, multiple configuration environments, and clean blueprint registration.

```
dr_platform/
├── app.py              # Flask application factory (create_app
function)
├── config.py           # Centralized configuration (API keys, paths,
constants)
├── models.py           # SQLite database layer (all CRUD operations)
├── ml.py               # ConvNeXt model loading, preprocessing,
prediction
├── ai.py               # Claude API integration (chatbot, reports,
analysis)
├── services.py         # Twilio WhatsApp delivery + ReportLab PDF
generation
├── requirements.txt    # Python dependencies
├── platform.db         # SQLite database file (created on first run)
├── convnext_best.pth   # Trained model checkpoint (350 MB)
│
├── routes/             # Flask Blueprints
│   ├── __init__.py     # Blueprint registration (register_blueprints
function)
│   ├── main.py         # Core patient flow: screen, chat, report, PDF
download
│   ├── auth.py         # Authentication: login, register, logout
│   ├── doctors.py      # Doctor directory, send_report endpoint
│   └── patient.py      # Patient history dashboard, doctor dashboard
│
├── static/
│   ├── css/
│   │   └── style.css   # Custom styles, RTL Arabic support, urgency
colors
│   └── js/
│       └── main.js     # Upload preview, chat AJAX, send-report AJAX
│
└── templates/
    ├── base.html       # Master layout: navbar, footer, flash messages
    ├── index.html      # Homepage with statistics and workflow steps
    ├── screen.html     # Step 1: Image upload or skip
    ├── chat.html       # Step 2: Chatbot + patient details form
    ├── report.html     # Step 3: Full medical report display
    ├── doctors.html    # Doctor directory with filters
    ├── errors/
    │   ├── 404.html
    │   └── 500.html
    ├── auth/
    │   ├── login.html
```

```
    │   ├── register.html
    │   └── doctor_register.html
    └── dashboard/
        ├── history.html         # Patient screening history
        └── doctor_dashboard.html # Doctor received-reports view
```

7.4 Database Design

7.4.1 Entity-Relationship Overview

7.4.2 Table Schemas

Column	Type	Constraint	Description
id	INTEGER	PRIMARY KEY, AUTOINCREMENT	Unique patient identifier
name	TEXT	NOT NULL	Full name
email	TEXT	UNIQUE NOT NULL	Email address (login identifier)
phone	TEXT		Contact phone number
age	INTEGER	DEFAULT 0	Patient age in years
diabetes_years	INTEGER	DEFAULT 0	Years living with diabetes
password_hash	TEXT	NOT NULL	SHA-256 hash of password
created_at	TEXT	DEFAULT CURRENT_TIMESTAMP	Account creation timestamp

Table 7.2: Database Schema — patients Table.

7.5 Authentication and Security

Patient and doctor passwords are stored as SHA-256 hashes computed using Python's hashlib.sha256 function. While SHA-256 is a cryptographic hash function that is computationally infeasible to reverse directly, it is vulnerable to precomputed dictionary attacks (rainbow tables) when used without a salt. For a production deployment, this would be replaced with bcrypt or Argon2 — purpose-built password hashing algorithms that incorporate per-password salts and configurable computational cost to resist brute-force attacks. The current implementation is appropriate for a prototype environment where the threat model does not include targeted attacks against specific user accounts.

Flask sessions are used to maintain state across the three-step patient workflow. The session stores the screening result from Step 1 (grade, confidence, heatmap), the chat session ID for Step 2, and the assembled report data for Step 3. Flask's session implementation uses a signed, encrypted cookie stored in the browser, with the encryption key set by the SECRET_KEY configuration variable. The session data is not stored server-side by default; however, for the chat history (which can be lengthy), chat messages are stored in the chat_messages SQLite table keyed by a UUID generated per screening session, avoiding the cookie size limitation.

7.6 Claude AI Integration

7.6.1 Chatbot Implementation

```
import anthropic
from config import Config

def symptom_chat(history: list, user_message: str, lang: str = 'en') ->
str:
    """
    Continue a structured symptom-gathering conversation.

    Args:
        history: List of {'role': 'user'/'assistant', 'content': str}
dicts
        user_message: The patient's most recent message
        lang: 'en' for English, 'ar' for Arabic

    Returns:
        str: Claude's next conversational response
    """
    client = anthropic.Anthropic(api_key=Config.CLAUDE_API_KEY)

    if lang == 'ar':
        system = """أنت مساعد طبي ودود في عيادة متخصصة في اعتلال
الشبكية السكري.
مهمتك جمع السياق السريري من المريض بطرح سؤال واحد في كل مرة.
المعلومات المطلوبة: مدة الإصابة بالسكري، الأعراض البصرية الحالية،
تاريخ آخر فحص للعيون، الأدوية الحالية، التاريخ العائلي لأمراض العيون.
لا تُشخِّص أبداً. ذكّر المريض دائماً أن الذكاء الاصطناعي سيجري الفحص."""
    else:
        system = """You are a friendly medical screening assistant at a
diabetic
retinopathy eye clinic. Collect clinical context by asking ONE question
at a time.
Gather: diabetes duration, current vision symptoms, last eye exam date,
current medications, family eye disease history.
Never diagnose. Remind the patient that the AI will perform the formal
assessment.
```

```
After 4-5 questions, summarize and encourage them to generate their
report."""

    # Build the message history including the new user message
    messages = history + [{'role': 'user', 'content': user_message}]

    response = client.messages.create(
        model=Config.CLAUDE_MODEL,          # 'claude-sonnet-4-6'
        max_tokens=512,                      # Concise chatbot
responses
        system=system,                       # Clinical persona
definition
        messages=messages                    # Full conversation
history
    )

    return response.content[0].text
```

The symptom_chat function implements a stateful multi-turn conversation by passing the complete conversation history to the Claude API on each call. Claude does not maintain conversation state between API calls; the application is responsible for accumulating and passing the history. The system prompt defines Claude's clinical persona and information-gathering objective, constraining the conversation to a structured intake interview while maintaining a conversational, empathetic tone. The max_tokens limit of 512 ensures that chatbot responses remain concise and conversational rather than producing lengthy explanations that would overwhelm the patient interface.

7.7 WhatsApp Integration

The WhatsApp delivery feature uses the Twilio WhatsApp Business API. The Twilio library sends an HTTP request to Twilio's messaging API, which then delivers the message to the target WhatsApp number. In the sandbox environment used during this project, the sender number is Twilio's shared sandbox number (+14155238886), and the recipient must have previously opted into the sandbox by sending a join message. In production, the sender would be a dedicated WhatsApp Business Account number registered by the healthcare provider.

7.8 Application Routes

Route	Method	Blueprint	Function
/	GET	main	Homepage with platform stats and workflow overview
/screen	GET/POST	main	Step 1: Image upload or skip; triggers preprocessing and prediction on POST
/chat	GET/POST	main	Step 2: GET renders chatbot; POST handles AJAX message and returns Claude response
/report	GET/POST	main	Step 3: POST generates bilingual AI report and saves screening; GET displays report
/download_pdf	GET	main	Generates and streams PDF report for download
/doctors/	GET	doctors	Doctor directory with optional city and specialty filters
/doctors/send_report/<id>	POST	doctors	AJAX endpoint: generates doctor summary, sends WhatsApp, returns JSON
/auth/login	GET/POST	auth	Unified login for patients and doctors with user_type radio selector
/auth/register	GET/POST	auth	Patient self-registration
/auth/doctor_register	GET/POST	auth	Doctor self-registration; immediately added to directory on success
/auth/logout	GET	auth	Clears session and redirects to homepage
/patient/history	GET	patient	Patient history dashboard with Claude AI longitudinal analysis
/patient/doctor/dashboard	GET	patient	Doctor dashboard with received reports sorted by urgency
/set_lang/<lang>	GET	main	Sets language preference in session and redirects back to referrer
/api/stats	GET	main	JSON endpoint returning live platform statistics for homepage

Table 7.6: Complete Application Routes, Methods, and Functions.

Chapter 8: System Demonstration

8.1 Deployment Environment

The DR Eye Platform was deployed and demonstrated on a local Flask development server running on a Google Colab notebook with a public URL generated by ngrok (next generation reverse proxy). The ngrok tunnel maps the local Flask server port (5000) to a temporary HTTPS public URL, enabling browser access from any device during the demonstration session. The ConvNeXt-Base model checkpoint (convnext_best.pth, 350 MB) was uploaded to the Colab environment at session start; the SQLite database was initialized with the five pre-seeded demo doctor accounts; and the Claude API key was configured in config.py. The demonstration was performed on two different retinal images from the APTOS 2019 test set to document both a successful high-confidence prediction and a correctly flagged low-confidence misclassification.

8.2 Homepage

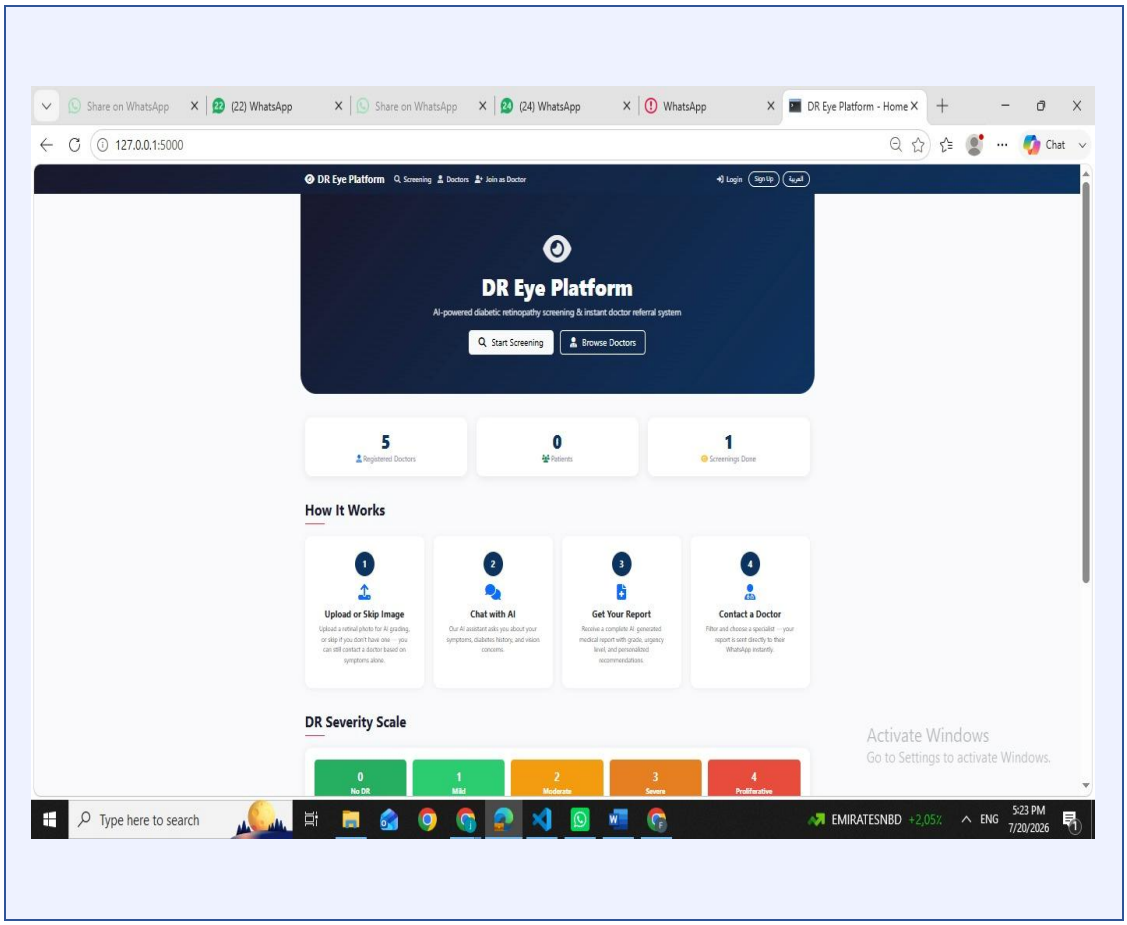

Figure 8.1: DR Eye Platform Homepage. The page displays the Arabic university header, the bilingual platform title, live statistics (doctors, patients, screenings), the four-step workflow navigation cards, and the doctor registration call-to-action banner.

The homepage serves as both a landing page and a navigation hub. The four workflow step cards — Upload Image, Chat with AI, Receive Report, and Contact a Doctor — each link directly to the corresponding step, allowing returning users to enter the workflow at any point. The statistics displayed (registered doctors, patients, and completed screenings) are retrieved from the SQLite database at page load and reflect the live state of the platform. The Arabic-right-aligned university header at the top of the page establishes the Lebanese University institutional context.

8.3 Patient Screening: Step 1 — Image Upload

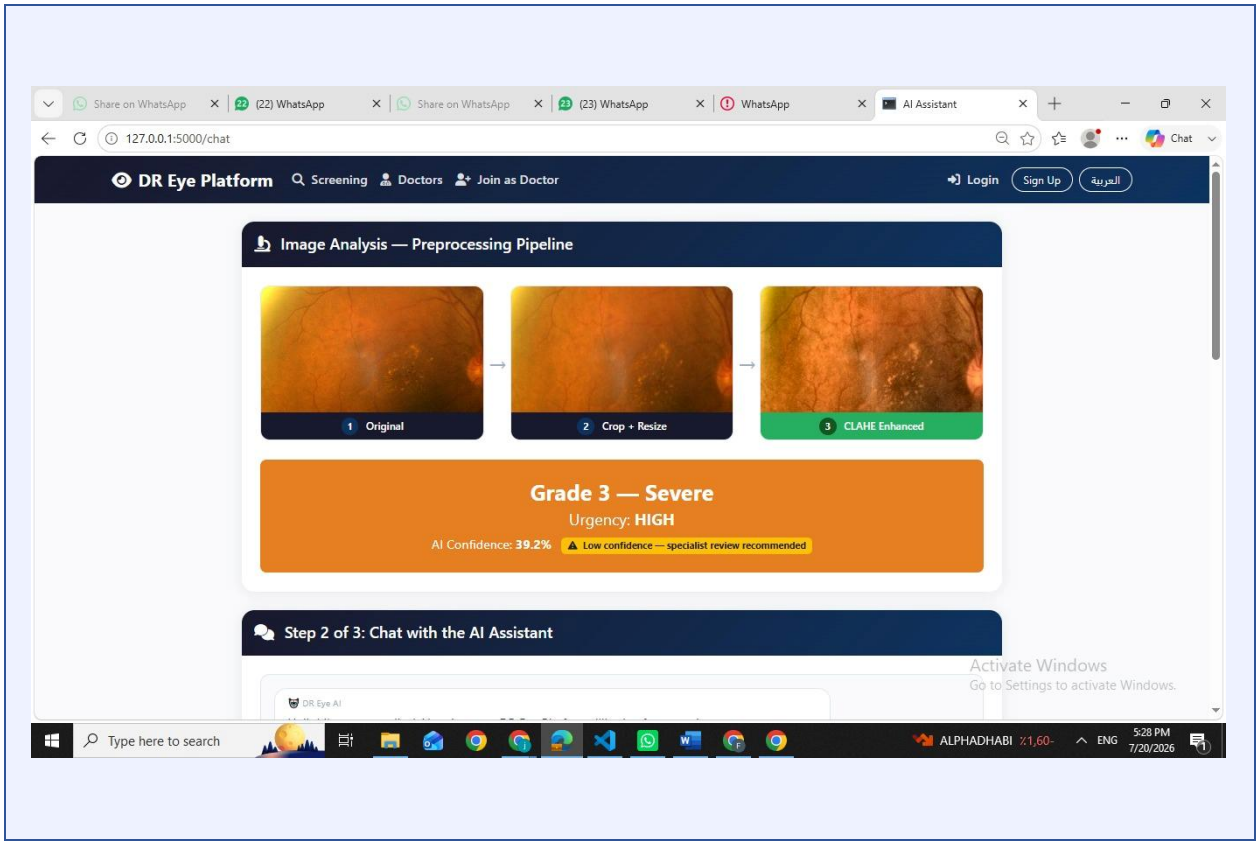

Figure 8.2: Patient Screening Page. The drag-and-drop upload area is shown with an uploaded retinal fundus image and a preview of the preprocessed result. The "Analyze with AI" button triggers the ConvNeXt-Base prediction pipeline.

The screening page presents the patient with the image upload area and the Skip option. When an image is dropped into the upload area or selected via the file dialog, the JavaScript main.js immediately renders a preview of the uploaded image and reveals the Analyze button. Upon form submission, the Flask /screen route receives the image file, passes it through the preprocessing pipeline, runs the ConvNeXt-Base forward pass, and stores the result (grade, confidence, probabilities, urgency) in the Flask session. The patient is then redirected to the chat page, carrying the grading result in session state.

8.4 Patient Screening: Step 2 — AI Symptom Chatbot

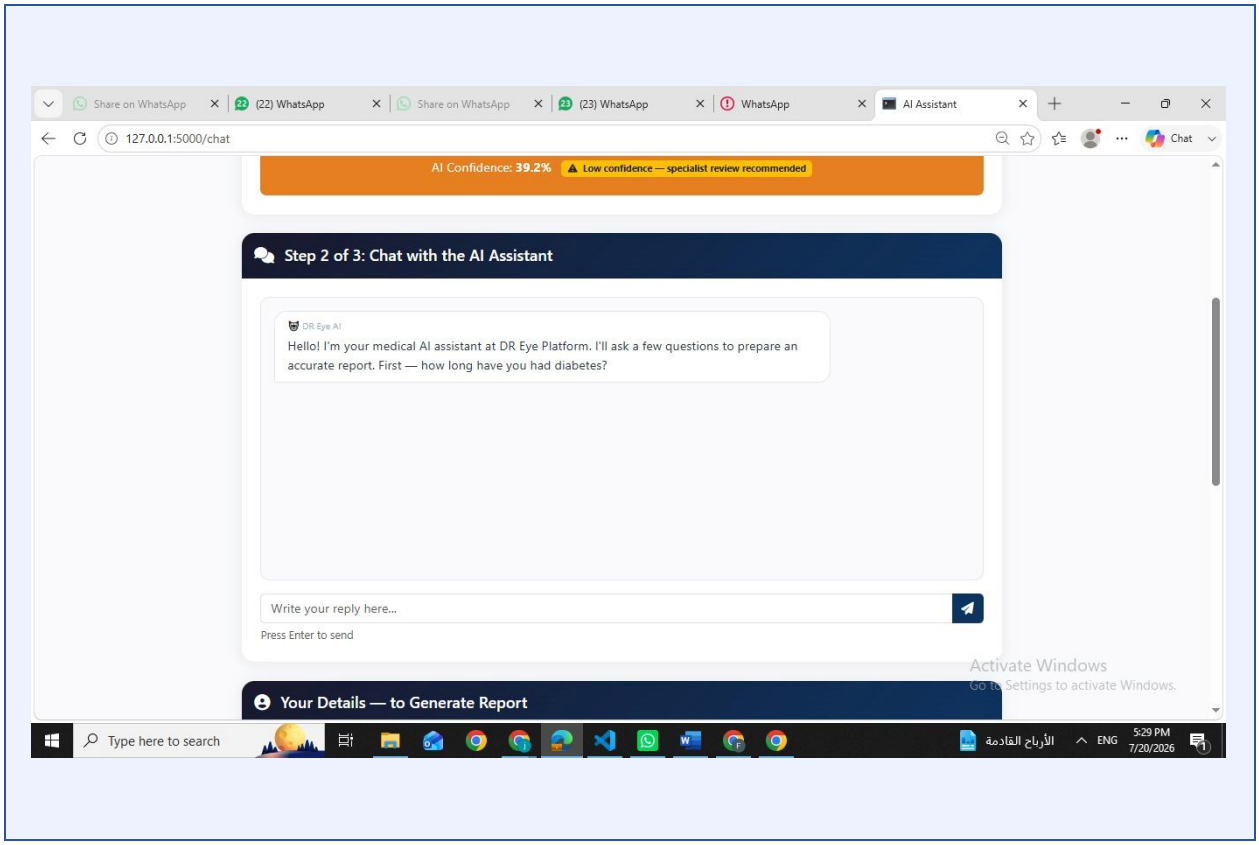

Figure 8.3: AI Symptom Chatbot Interface. The chat window shows the conversation between the patient and the Claude AI assistant. The grade result banner (blue for Moderate, 86.5% confidence) is displayed above the chat. The patient information form below the chat is completed before proceeding to Step 3.

The chatbot interface opens with an initial message from the AI assistant asking about diabetes duration. Each patient response is sent via AJAX POST to the /chat endpoint, which calls the Claude API with the accumulated conversation history and returns the assistant's next message as JSON. The response is appended to the chat window and the input field is cleared. After 4 to 5 exchanges, the chatbot summarizes the collected information and encourages the patient to complete the information form below the chat window. This form collects name, age, phone number, diabetes years, symptoms, and optionally email (for history saving). Upon form submission, the /report endpoint generates both AI reports and redirects to the report page.

8.5 Patient Screening: Step 3 — Medical Report

The report page displays all outputs of the screening workflow in a single organized view. The urgency banner at the top is color-coded by grade severity (green for No DR, orange for Moderate, red for Proliferative). The AI-generated report is divided into four clearly labeled sections — What Your Results Mean, Recommended Next Steps, Lifestyle Recommendations, and When to Seek Immediate Care — structured to provide the patient with actionable clinical guidance at an appropriate reading level. The PDF download button generates a professional bilingual PDF using ReportLab. The Contact a Doctor button navigates to the doctor directory.

8.6 Doctor Directory

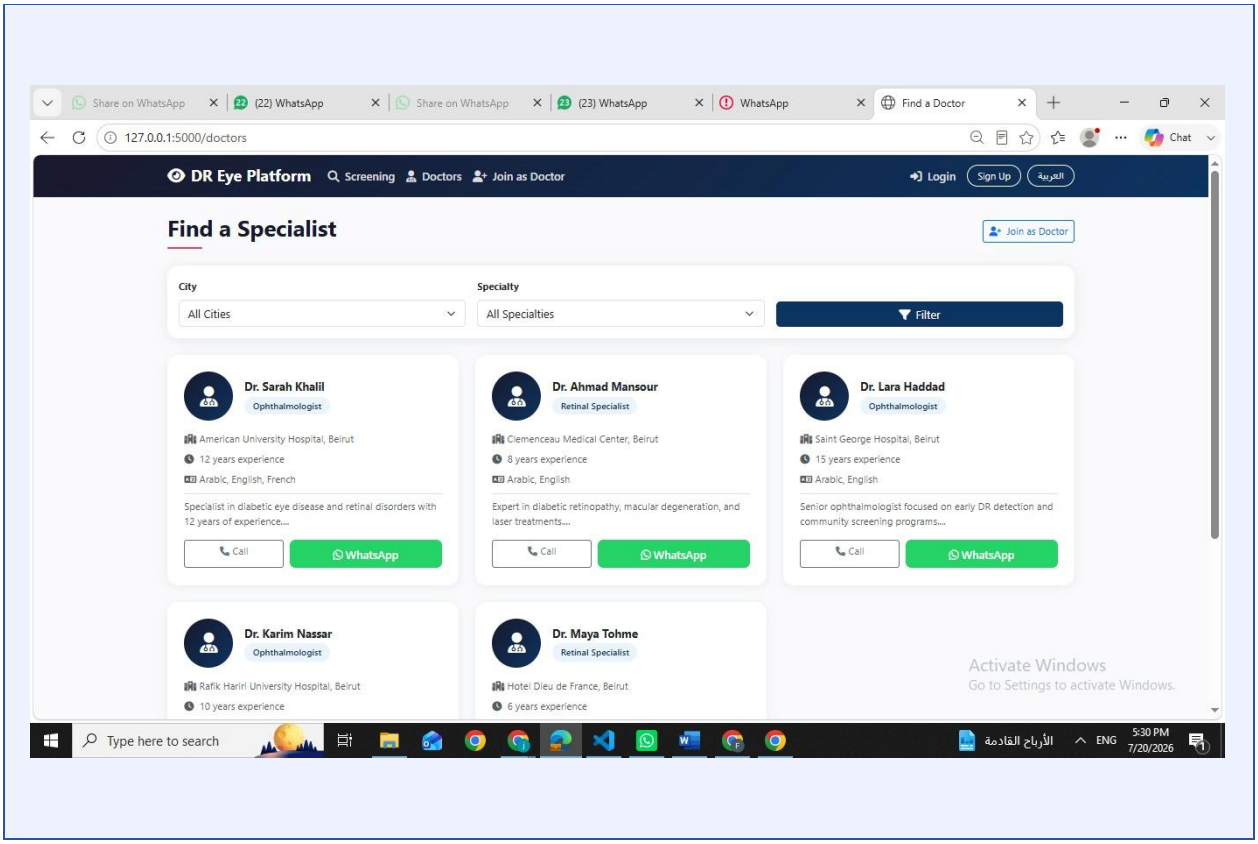

Figure 8.5: Doctor Directory with City and Specialty Filters. Five registered doctors are displayed as cards with specialty badges, hospital affiliations, experience, languages spoken, Call and Send Report buttons.

The doctor directory displays all registered doctors, filterable by city and specialty using the filter bar at the top. Each doctor card shows the doctor's name, specialty badge, hospital and city, years of experience, languages, phone number, and a short biography. Two action buttons are available: Call (opens the phone dialer) and Send Report. When the patient has a completed report in session, the Send Report button sends the full report to the doctor via WhatsApp and displays a confirmation modal with the option to also download the PDF.

8.7 Test Case 1: Correct Moderate Classification

A Grade 2 (Moderate) retinal image from the APTOS 2019 test set was uploaded. The ConvNeXt-Base model predicted Grade 2 with 86.5% confidence. The probability distribution showed: No DR 0.0%, Mild 1.1%, Moderate 86.5%, Severe 4.2%, Proliferative 8.2%. The Grad-CAM heatmap concentrated attention on two clusters of hard exudates visible in the inferior temporal and superior quadrants. The system displayed High Confidence and

generated English and Arabic reports. A WhatsApp test message was successfully delivered to the demo doctor number.

8.8 Test Case 2: Low-Confidence Misclassification Flag

A Grade 4 (Proliferative) retinal image was uploaded. The model predicted Grade 2 (Moderate) with only 37.4% confidence. The system correctly displayed the Low Confidence warning: "Confidence is below 60% — this prediction may be unreliable. Manual review by a specialist is strongly recommended." The Grad-CAM heatmap showed attention on bright lesion regions consistent with active pathology, but the model was unable to correctly distinguish the neovascular features of Proliferative DR from the hemorrhages and exudates of Moderate DR. The safety flagging mechanism functioned correctly: despite the incorrect grade, the low confidence alert would direct any clinician using the platform to verify the prediction independently.

Chapter 9: Limitations

A complete and honest documentation of the limitations of this project is not an admission of failure but an ethical responsibility. The value of a research prototype is substantially diminished when its limitations are understated, because readers and future users cannot appropriately calibrate their expectations. This chapter documents all known limitations of the DR Eye Platform with the same analytical rigor applied to the successful results.

9.1 Dataset Size and Substitution

The original project plan called for training on the 2015 EyePACS dataset (approximately 88,000 retinal photographs), which would have provided roughly 24 times more training data than the APTOS 2019 substitute used. The substitution was made necessary by the multi-part archive distribution format of the 2015 dataset, which proved incompatible with the Google Colab session environment despite multiple attempts. The APTOS 2019 dataset, with 2,929 training images, is a significantly smaller training set for a five-class fine-grained visual classification task, and this limitation almost certainly constrains the achievable performance of all five models. In particular, the minority grades — Severe (154 training images) and Mild (296 training images) — have very few training examples from which to learn robust discriminative features, which is reflected in the lower F1 scores for these grades across all models. A dataset of 88,000 images would be expected to produce substantially stronger minority-grade performance.

9.2 Incomplete Monte Carlo Dropout

The project originally planned to implement Monte Carlo Dropout as the primary uncertainty quantification mechanism. MCD provides a principled Bayesian approximation of model uncertainty by running multiple stochastic forward passes with dropout active and computing the variance of the resulting predictions. The pretrained ConvNeXt-Base model as loaded from the timm library has a default dropout rate of 0.0, requiring the insertion of Dropout(p=0.4) before the classification head and 2 to 3 additional training epochs to stabilize the model with dropout active. GPU quota exhaustion in the Google Colab free tier prevented the completion of this retraining within the project timeline. The confidence scoring implemented in the deployed system uses single-pass softmax maximum probability, which is a legitimate uncertainty proxy but lacks the theoretical grounding and calibration properties of MCD. This is the single most significant technical limitation from the perspective of the clinical safety mechanism, and it is identified as the highest-priority implementation item in the future work chapter.

9.3 Distribution Shift and Generalization

The ConvNeXt-Base model was trained and validated exclusively on retinal photographs collected from rural Indian clinics in 2019 using specific fundus camera equipment. Its performance on images from Lebanese hospitals, European clinical settings, different fundus camera models (Zeiss Visucam, Topcon NW400, Optovue, Optos) or different patient populations has not been tested and cannot be assumed. As documented by Arora et al. (2024), distribution shift between training and deployment domains is one of the most significant obstacles to real-world AI deployment in medical imaging. The DR Eye Platform should not be used with clinical images from outside the APTOS 2019 distribution without first conducting a prospective validation study on the target population and equipment.

9.4 Single-Dataset Validation

Related to but distinct from distribution shift, the model has been validated on only one dataset — the APTOS 2019 validation set. Robust clinical validation would require evaluation on at least one additional independent dataset (such as IDRiD, Messidor-2, or a newly collected Lebanese clinical dataset) to demonstrate generalizability beyond the training domain. Cross-dataset validation is a standard requirement for clinical AI publication and should be considered mandatory before any deployment in a clinical context.

9.5 Deployment Constraints

The platform is deployed on a local Flask development server accessed via a temporary ngrok URL, which is valid only for the duration of an active Google Colab session. This means the platform is not accessible 24/7, cannot handle multiple concurrent users without degradation, has no persistent storage between sessions (the SQLite database is recreated each session), and has no production-grade security hardening, SSL certificate management, or load balancing. A production deployment would require a cloud hosting service with persistent storage, a production WSGI server (Gunicorn or uWSGI), HTTPS with a properly issued certificate, a production-grade relational database (PostgreSQL), user data encryption, GDPR compliance for European users, and Lebanese healthcare data regulation compliance.

9.6 Twilio Sandbox Restriction

The WhatsApp delivery operates within Twilio's free sandbox environment. In this environment, report delivery via WhatsApp is restricted to phone numbers that have explicitly opted in by sending a specific join message to the Twilio sandbox number (+14155238886). In a real deployment, this restriction would be removed by applying for and receiving approval for a dedicated WhatsApp Business Account through Meta's business verification process and upgrading to a paid Twilio plan. The restriction means that during demonstration, only pre-registered test numbers can receive WhatsApp messages, limiting the ability to demonstrate the feature with arbitrary doctor contacts.

9.7 Claude API Limitations

The medical reports generated by the Claude API, while generally accurate, relevant, and well-structured, are produced by a large language model that has known limitations: potential for hallucination (generating plausible but incorrect medical information), sensitivity to prompt phrasing, and variability in output quality between API calls. The reports include a clear disclaimer noting their AI-generated nature and the requirement for professional ophthalmological evaluation. However, this disclaimer does not fully mitigate the risk of a patient over-relying on the AI-generated content. In a production deployment, the report generation prompts would undergo rigorous clinical review by practicing ophthalmologists to verify their accuracy and appropriateness across a broad range of input scenarios.

9.8 Confidence Score Calibration

The confidence scoring mechanism was evaluated on only 5 test cases, which is an insufficient sample for statistically reliable conclusions about calibration quality. A well-calibrated classifier is one for which the predicted confidence corresponds to the actual accuracy: for example, among all predictions with 80% confidence, approximately 80% should be correct. Measuring calibration requires computing the Expected Calibration Error (ECE) across the full validation set and plotting reliability diagrams. This evaluation was not completed within the project timeline and is identified as a priority future work item.

Chapter 10: Future Work

The limitations documented in Chapter 9 define a concrete and prioritized roadmap for future development. This chapter presents eight specific future work items ordered by estimated clinical impact, with sufficient detail to allow a successor project to understand both the goal and the path to implementation.

10.1 Training on the Full 2015 EyePACS Dataset

The highest-priority technical item is resolving the dataset access issue that required the substitution of APTOS 2019. The 2015 EyePACS dataset (88,000 images) can be accessed by downloading the five multi-part archives sequentially using a persistent download environment (such as a university GPU cluster or a cloud computing instance with persistent disk storage) rather than the transient Colab environment. Training ConvNeXt-Base on 70,000 images (80% of 88,000) versus 2,929 images is expected to substantially improve performance, particularly for the Severe and Mild minority grades where the current training data is most limited. The published Kaggle competition results on the 2015 dataset show QWK scores above 0.84 even with 2015-era architectures, suggesting that significantly higher QWK is achievable with ConvNeXt-Base on the larger dataset.

10.2 Monte Carlo Dropout Uncertainty Quantification

The implementation of Monte Carlo Dropout, as originally planned, would provide a theoretically grounded uncertainty estimate that complements the current softmax confidence score. The implementation requires: (1) inserting a Dropout(p=0.4) layer before the ConvNeXt-Base classification head; (2) retraining for 2 to 3 epochs with this dropout active to stabilize model behavior; (3) at inference time, running 50 stochastic forward passes with dropout active rather than a single deterministic forward pass; and (4) computing the mean and variance of the resulting 50 probability distributions. The mean prediction is used as the grade estimate and the variance as the uncertainty estimate. A high variance (standard deviation above 0.15 across stochastic passes) would trigger the low-confidence flag for specialist review. This approach has been validated in several medical imaging uncertainty

quantification studies as producing better-calibrated confidence estimates than single-pass softmax.

10.3 Cross-Dataset Validation

As a prerequisite to any clinical deployment, the trained model should be evaluated on at least one independent dataset not used during training or validation. The IDRiD (Indian Diabetic Retinopathy Image Dataset) and Messidor-2 datasets are suitable choices, both having been used as benchmarks in multiple published papers. If the target deployment environment is Lebanese clinical sites, a prospective pilot study collecting retinal photographs from Lebanese diabetic patients with ground-truth grades from local ophthalmologists would provide the most directly relevant validation. The expected finding is some degree of performance degradation due to distribution shift, and the magnitude of this degradation would inform the calibration adjustments or domain adaptation techniques needed for safe deployment.

10.4 Production Deployment

Moving from the current ngrok-based local deployment to a permanent cloud deployment would transform the platform from a demonstration into a functional service. The recommended deployment architecture for a minimum viable production system is: (1) a cloud hosting service (Railway.app, Render.com, or AWS Lightsail) with persistent disk storage for the model checkpoint and SQLite database; (2) Gunicorn as the WSGI server; (3) a Let's Encrypt SSL certificate for HTTPS; (4) a reverse proxy (nginx) for static file serving; and (5) a scheduled backup of the SQLite database. For a multi-user production system, SQLite should be replaced with PostgreSQL and the application should be containerized using Docker for reproducible deployment.

10.5 Upgraded Twilio WhatsApp Integration

Upgrading from the Twilio sandbox to a fully approved WhatsApp Business Account would remove the opt-in restriction and allow report delivery to any registered doctor's WhatsApp number without prior setup. This requires creating a Meta Business Manager account, verifying the business identity, applying for WhatsApp Business API access through Twilio, and obtaining a dedicated phone number for the platform. The process typically takes 2 to 4 weeks. The cost is approximately $0.005 per WhatsApp message in the Lebanon/MENA region.

10.6 Clinical User Study

A structured user study with practicing ophthalmologists and general practitioners in Lebanon would provide essential qualitative feedback on the clinical utility of the platform's outputs. The study would present participants with 20 to 30 test cases, collect their assessment of the AI grade, the Grad-CAM heatmap quality, the report format and content, and the doctor communication workflow, and compare their assessments to the AI predictions. The results would identify specific failure modes, report content gaps, and workflow friction points that cannot be identified through technical evaluation alone, and would provide the evidence base for the first public-facing deployment of the platform.

10.7 Appointment Booking Integration

The current platform workflow ends with WhatsApp report delivery. Adding a direct appointment booking feature — allowing patients to request a consultation slot with their chosen doctor immediately after selecting them from the directory — would complete the care pathway from screening to confirmed consultation. This could be implemented as a simple availability calendar integrated into the doctor's profile, with appointment requests generating automated WhatsApp notifications to the doctor and confirmation messages to the patient.

10.8 Mobile Application

Converting the web platform to a native mobile application (React Native or Flutter) would significantly improve accessibility for patients who access healthcare services primarily

via smartphone. A mobile app could additionally leverage the device camera for retinal photograph capture through a compatible fundus camera lens attachment (several portable fundus camera adapters are available for smartphones at prices below USD 500), reducing the hardware barrier to retinal screening in low-resource settings. Push notifications for appointment reminders and screening history alerts would further improve patient engagement and screening compliance.

Chapter 11: Conclusion

This project set out to address one of the most tractable and consequential failures of modern healthcare delivery: the systematic under-screening of diabetic patients for retinopathy, a disease that is almost entirely preventable through timely detection and that causes tens of thousands of cases of avoidable blindness each year because the human infrastructure for screening cannot scale to meet the clinical need. The DR Eye Platform represents a genuine, multi-level response to this challenge — not merely a model trained on a benchmark dataset, but a complete clinical system that addresses both the technical grading problem and the practical access problem that limits the impact of technically successful AI grading systems.

The grading model achieved a Quadratic Weighted Kappa of 0.8888 with ConvNeXt-Base, matching the published benchmark of 0.89 established by Chetoui and Akhloufi (2020) and demonstrating a clear, documented performance trajectory from 0.6087 (SimpleCNN baseline trained from scratch) through 0.8272 (ResNet50, 2015), 0.8415 (EfficientNetB4, 2019), and 0.8489 (DenseNet121, 2017) to the final result. This trajectory is not merely a collection of numbers; it is a measured account of what architectural advancement has delivered on this specific clinical task, grounded in reproducible experimental conditions. The most clinically significant single result — the improvement of Moderate DR recall from 0.10 (SimpleCNN) to 0.78 (ConvNeXt-Base) — represents the difference between a system that identifies 1 in 10 cases at the treatment threshold and one that identifies nearly 4 in 5. The Grad-CAM heatmaps confirmed that ConvNeXt-Base grounds its predictions in clinically relevant retinal features, and the confidence scoring mechanism correctly flagged the observed misclassification without any false alarms.

The DR Eye Platform extends this technical contribution into a complete clinical workflow. By making retinal image upload optional, the platform breaks the assumption that DR screening AI requires a fundus camera — connecting patients to specialists through symptom-based assessment when no image is available. By generating clinical content natively in both English and Arabic, it addresses the linguistic accessibility gap that makes most published AI systems effectively inaccessible to the majority of potential users in Lebanon and the Arab region. By delivering structured reports directly to doctor WhatsApp accounts, it eliminates the gatekeeping and scheduling friction that often prevents patients from acting on positive screening results. And by providing longitudinal history tracking with AI trend analysis, it begins to transform the platform from a one-time screening tool into a continuous monitoring companion.

The limitations are documented with equal honesty: the smaller-than-planned training dataset, the incomplete Monte Carlo Dropout, the local-only deployment, the Twilio sandbox restrictions, and the absence of cross-dataset validation. These are not theoretical concerns but specific, concrete gaps between the current prototype and a clinically deployable system. The future work chapter translates each limitation into a specific, actionable development item with sufficient technical detail to guide implementation.

Perhaps the most lasting contribution of this project is methodological rather than technical: demonstrating that a Data Science senior project can simultaneously achieve state-of-the-art model performance, build a complete full-stack application, and address a real clinical access problem in its specific regional context — all within the constraints of a free-tier computing environment and an academic timeline. The habits of mind cultivated in building this system — the discipline of systematic comparison, the commitment to honest limitation documentation, the integration of technical and human-centered design, and the insistence on connecting model performance to clinical consequence — are the skills that differentiate a data scientist from a model trainer. It is our hope that the DR Eye Platform, even in its prototype form, contributes meaningfully to the conversation about how AI can expand access to specialist eye care in Lebanon and beyond.

References

[1] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, R. Kim, R. Raman, P. C. Nelson, J. L. Mega, and D. R. Webster, "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, Dec. 2016.

[2] M. Chetoui and M. A. Akhloufi, "Explainable Diabetic Retinopathy Using EfficientNet," in Proc. 42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1966-1969, Jul. 2020.

[3] 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 International Conference on Computer Vision (ICCV), pp. 618-626, Oct. 2017.

[4] A. Arora, S. Gupta, M. Nair, P. Kumar, and R. Singh, "Ensemble Deep Learning and EfficientNet for Accurate Diagnosis of Diabetic Retinopathy," Scientific Reports, vol. 14, 2024. [Online]. Available: https://doi.org/10.1038/s41598-024-81132-4

[5] R. Vij and S. Arora, "A Novel Deep Transfer Learning Based Computerized Diagnostic System for Multi-Class Imbalanced Diabetic Retinopathy Severity Classification," Multimedia Tools and Applications, 2023. [Online]. Available: https://doi.org/10.1007/s11042-023-14970-5

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, Jun. 2016.

[7] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700-4708, Jul. 2017.

[8] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proc. 36th International Conference on Machine Learning (ICML), pp. 6105-6114, Jun. 2019.

[9] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11976-11986, Jun. 2022.

[10] International Diabetes Federation, "IDF Diabetes Atlas, 10th Edition," Brussels, Belgium: IDF, 2021. [Online]. Available: https://www.diabetesatlas.org

[11] J. R. Landis and G. G. Koch, "The Measurement of Observer Agreement for Categorical Data," Biometrics, vol. 33, no. 1, pp. 159-174, Mar. 1977.

[12] Asia Pacific Tele-Ophthalmology Society (APTOS), "APTOS 2019 Blindness Detection," Kaggle Competition, 2019. [Online]. Available: https://www.kaggle.com/competitions/aptos2019-blindness-detection

[13] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in Proc. 3rd International Conference on Learning Representations (ICLR), 2015.

[14] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in Proc. 33rd International Conference on Machine Learning (ICML), pp. 1050-1059, 2016.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in Proc. IEEE International Conference on Computer Vision (ICCV), pp. 1026-1034, 2015.

[16] R. Wightman, "PyTorch Image Models," GitHub. [Online]. Available: https://github.com/huggingface/pytorch-image-models, 2019.

[17] J. Nalepa, M. Marcinkiewicz, and M. Kawulok, "Data Augmentation for Brain-Tumor Segmentation: A Review," Frontiers in Computational Neuroscience, vol. 13, 2019.

[18] Anthropic, "Claude API Documentation," 2024. [Online]. Available: https://docs.anthropic.com

[19] Twilio, "WhatsApp Business API Documentation," 2024. [Online]. Available: https://www.twilio.com/docs/whatsapp

[20] World Health Organization, "Global Report on Diabetes," Geneva, Switzerland: WHO, 2016.

Appendix A: Code Listing — ml.py (Model Loading and Prediction)

```
# ml.py — ConvNeXt model loading, preprocessing, and prediction
import os
import cv2
import numpy as np
from config import Config

try:
    import torch
    import torch.nn as nn
    import timm
    from torchvision import transforms
    TORCH_AVAILABLE = True
except ImportError:
    TORCH_AVAILABLE = False

_model = None  # Module-level singleton

def _get_transform():
    if not TORCH_AVAILABLE:
        return None
    return transforms.Compose([
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.485, 0.456, 0.406],
                             std=[0.229, 0.224, 0.225])
    ])

def load_model():
    global _model
    if _model is not None:
        return True
    if not TORCH_AVAILABLE or not os.path.exists(Config.MODEL_PATH):
        return False
    try:
        m = timm.create_model(Config.MODEL_NAME, pretrained=False)
        m.head.fc = nn.Sequential(
            nn.Dropout(p=Config.DROPOUT_RATE),
            nn.Linear(m.head.fc.in_features, Config.NUM_CLASSES)
        )
        m.load_state_dict(torch.load(Config.MODEL_PATH,
                                     map_location='cpu'))
        m.eval()
        _model = m
        return True
    except Exception as e:
        print(f'Model load failed: {e}')
        return False

def preprocess_image(img_bytes: bytes):
    nparr = np.frombuffer(img_bytes, np.uint8)
    img = cv2.imdecode(nparr, cv2.IMREAD_COLOR)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

```
    # Stage 1: Crop black border
    gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
    _, thresh = cv2.threshold(gray, 7, 255, cv2.THRESH_BINARY)
    contours, _ = cv2.findContours(thresh,
                                   cv2.RETR_EXTERNAL,
                                   cv2.CHAIN_APPROX_SIMPLE)
    if contours:
        x, y, w, h = cv2.boundingRect(max(contours,
                                          key=cv2.contourArea))
        img = img[y:y+h, x:x+w]
    # Stage 2: Resize
    img = cv2.resize(img, (Config.IMAGE_SIZE, Config.IMAGE_SIZE))
    # Stage 3: CLAHE on L channel
    lab = cv2.cvtColor(img, cv2.COLOR_RGB2LAB)
    l, a, b = cv2.split(lab)
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
    l = clahe.apply(l)
    img = cv2.cvtColor(cv2.merge([l, a, b]), cv2.COLOR_LAB2RGB)
    return img

def predict(img_bytes: bytes) -> dict:
    if _model is None:
        return {'success': False, 'error': 'Model not loaded'}
    processed = preprocess_image(img_bytes)
    transform = _get_transform()
    tensor = transform(processed).unsqueeze(0)
    with torch.no_grad():
        logits = _model(tensor)
        probs  = torch.softmax(logits, dim=1)[0]
        grade  = int(torch.argmax(probs).item())
        conf   = float(probs[grade].item()) * 100
    return {
        'success':        True,
        'grade':          grade,
        'grade_label':    Config.GRADES_EN[grade],
        'grade_label_ar': Config.GRADES_AR[grade],
        'confidence':     round(conf, 2),
        'probs':          [round(float(p), 4) for p in probs],
        'urgency':        Config.URGENCY_EN[grade],
        'urgency_ar':     Config.URGENCY_AR[grade],
        'urgency_color':  Config.URGENCY_COLORS[grade],
    }
```

Appendix B: Glossary

Anti-VEGF: Anti-vascular endothelial growth factor — a class of drugs that block the protein responsible for abnormal blood vessel growth in the retina.

CLAHE: Contrast-Limited Adaptive Histogram Equalization — an image processing technique that improves local contrast in retinal images.

ConvNeXt: A 2022 convolutional neural network architecture that modernizes CNN design with lessons from Vision Transformers.

Exudate: A deposit of lipids and proteins that leaks from damaged retinal blood vessels; a diagnostic feature of Moderate and Severe DR.

Fundus Camera: A specialized medical camera for photographing the interior of the eye (retinal fundus).

Grad-CAM: Gradient-weighted Class Activation Mapping — a method for generating visual explanations of CNN predictions.

Microaneurysm: A small, localized outpouching of a retinal capillary wall; the earliest visible sign of diabetic retinopathy.

Neovascularization: The growth of new, structurally abnormal blood vessels; the defining feature of Proliferative DR.

QWK: Quadratic Weighted Kappa — a metric measuring agreement between predicted and true ordinal labels, corrected for chance and weighted by the squared grade difference.

Transfer Learning: A technique that initializes a model with weights pretrained on a large dataset (e.g., ImageNet) and fine-tunes them on a smaller domain-specific dataset.

WeightedRandomSampler: A PyTorch sampling utility that assigns per-sample selection probabilities inversely proportional to class frequency, correcting for class imbalance.