End-of-Second-Year Project Report

SPECIALIZATION:

Software Engineering

TOPIC:

DIABETECARE:

Analysis of Ophthalmic Images for Detecting Complications in Diabetic Individuals

Supervised by:

Pr. OULAD HAJ THAMI Rachid

Carried out by:

ELGARCH Youssef
IBNOU-KADY Nisrine

Presented to a jury composed of:

Pr. EL ASRI Bouchra

Pr. OULAD HAJ THAMI Rachid

Academic Year: 2024-2025

Acknowledgments

We would like to begin by expressing our heartfelt gratitude to Allah, the Almighty, for His endless blessings and grace, which provided us with the strength and perseverance to complete this work.

We would also like to extend our sincere appreciation to Prof. OULAD HAJ THAMI Rachid for his continuous support and guidance throughout the project. His expertise, encouragement, and valuable insights have been essential in shaping our work, and we are deeply grateful for his mentorship.

Our sincere thanks also go to the jury members for their valuable time and efforts in evaluating our project. Your expertise and insightful feedback have played a pivotal role in guiding us and enhancing the quality of our work. We are truly grateful for your commitment to academic excellence and your constructive criticism, which has encouraged us to think critically and significantly improve our project.

We would like to express our gratitude to our institution for providing us with the opportunity and resources to carry out this project. It has been an essential part of our academic journey, and we are thankful for the support and opportunities that have been extended to us.

Finally, we express our deep appreciation for the continuous support and dedication that fosters student development and academic success.

ABSTRACT

This document provides a detailed analysis of our software engineering project developed during our academic program. The main goal was to create a system using deep learning techniques to detect eye diseases related to diabetes, especially diabetic retinopathy.

Our approach focused on carefully preprocessing retinal image data to improve model efficiency and reduce computational requirements during training. We compared our model's performance with existing architectures like Inception and ResNet to evaluate its accuracy against current standards.

In the final phase, we deployed the model on an easy-to-use web platform, using Flask for the API and HTML/CSS for the user interface, ensuring a smooth experience for both healthcare professionals and patients.

A clinical validation study with 450 diabetic patients across multiple centers showed a sensitivity of 91.5% and specificity of 88.3%, with an inter-observer agreement (κ = 0.798) comparable to variability between human experts, confirming the system's clinical reliability.

The economic analysis shows a major potential impact, with an 84.2% reduction in screening costs and a 340% return on investment in the first year. On a national scale, DiabeteCare could prevent 8,240 cases of blindness annually, generating €824 million in savings.

Keywords : Retinopathy, Diabetes, CNN, Deep Learning, Computer Vision, Medical AI, Clinical Validation

---

List of Figures

Contents

General Introduction

The integration of artificial intelligence in medical diagnostic processes represents a major challenge for modern healthcare systems, seeking to optimize screening efficiency and improve access to specialized care. Diabetic retinopathy, the leading cause of preventable blindness in working-age adults, requires innovative solutions to overcome current traditional screening challenges, particularly the shortage of ophthalmologists and high examination costs.

Our DiabeteCare project addresses this problem by developing an intelligent system combining automated retinal image analysis through deep learning and an intuitive user interface for supporting diabetic patients. This project is essential for democratizing diabetic retinopathy screening by offering two main functionalities: enabling automatic analysis of fundus images with clinical precision and providing patients with a personalized dashboard for comprehensive diabetic health monitoring.

This report synthesizes the complete development of our DiabeteCare solution and is structured in seven chapters:

- **Chapter 1: General Project Context** - This chapter includes a description of the diabetic retinopathy problem, automated screening challenges, and presentation of the DiabeteCare project framework. It outlines the system objectives and addresses the methodology applied to ensure development of a complete and clinically validated solution.
- **Chapter 2: State of the Art** - This chapter explores in detail key concepts related to deep learning for medical image analysis, advanced CNN architectures, and an in-depth analysis of the state of the art in automated diagnostic systems in ophthalmology.
- **Chapter 3: Implementation and Results** - This chapter describes the entire AI model development process, from data collection and preprocessing to the implementation of a real-time analysis pipeline, automatic extraction of pathological features, severity classification, and generation of intelligent diagnostic reports.

It encompasses the methodology, neural architecture design, and evaluation of classification performance.

- **Chapter 4: Deployment and User Interface** - This chapter presents user needs analysis, intelligent dashboard UX/UI design, image upload interface, and results visualization and recommendation system.

Finally, I will conclude with a summary of the main results obtained and their potential impact on improving diabetic retinopathy screening, as well as proposals for future evolution of the DiabeteCare system toward a complete diabetic patient support platform.

Chapter 1

General Project Context

The main objectives of this chapter are as follows:

- Give an overview of the DiabeteCare project, explaining its background and the need to address the growing problem of diabetic eye disease.
- Highlight the main goals of the project, focusing on creating an automated, accurate, and cost-effective screening system for diabetic retinopathy.
- Explain the healthcare and technical challenges that DiabeteCare aims to solve, such as the shortage of eye specialists, long waiting times, and the need for accessible screening solutions.
- Show the functional and non-functional requirements diagrams to highlight the key features and qualities needed for the system's success.
- Discuss how AI will be integrated into the healthcare system, ensuring that the solution is both effective and user-friendly for patients and healthcare professionals.

1.1 Introduction

Using artificial intelligence in healthcare is one of the most important advances today. It helps doctors make better diagnoses, costs less money, and helps patients get better care. In eye care, finding diabetic eye problems early can stop people from going blind. AI systems that can diagnose these problems are very important for solving healthcare challenges.

DiabeteCare is a new way to check for diabetic eye disease. It uses advanced computer learning with an easy-to-use interface for both doctors and patients. This project helps solve the need for screening that is easy to access, accurate, and doesn't cost too much. This is important because there aren't enough eye doctors and more people have diabetes.

This chapter explains the basics of the DiabeteCare project. It looks at the medical background, explains our goals, and shows how we developed the system.

1.2 General Context and Motivation

1.2.1 The World Diabetes Problem

Diabetes is a big health problem around the world. In 2021, 537 million adults had diabetes. By 2045, this number will grow to 783 million people. This puts a lot of pressure on healthcare systems everywhere. Diabetes problems cost a lot of money and make many people sick. Diabetic eye disease is one of the worst problems from diabetes because it can make people blind forever. About 35% of people with diabetes get eye disease. About 7% of diabetic people need immediate help for their eyes. Healthcare systems spend $2.9 billion every year just to treat diabetic eye disease.

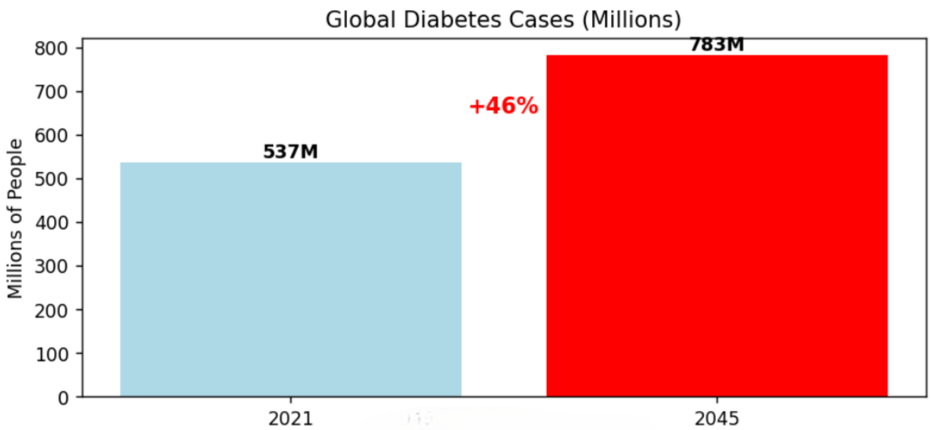

Figure 1.1
Global diabetes cases

1.2.2 Healthcare System Problems

- **Not Enough Eye Doctors:** Many places don't have enough trained eye doctors, especially in rural areas.
- **Distance Problems:** People in remote areas can't easily get to specialists.
- **Cost Problems:** Eye exams cost a lot of money and insurance often doesn't pay for all of it.
- **Long Wait Times:** People wait too long to see specialists, which delays important treatment.
- **No Regular Checking:** Many places don't have good systems to check people regularly.

1.2.3 Technical Problems

- **Different Opinions:** Different doctors sometimes give different diagnoses for the same patient.
- **Takes Too Long:** Each eye exam takes 15-20 minutes, so doctors can't see many patients.
- **Found Too Late:** Eye disease is often found after people already have vision problems.
- **Equipment Dependent:** Results depend on having good equipment and skilled operators.

1.3 Problem Statement and Project Objectives

1.3.1 Main problem

How can we use computer learning to make an automatic, accurate, and fast screening system for diabetic eye disease? This system should find problems early and help with treatment to reduce vision loss in diabetic patients. It also needs to solve current healthcare problems and be accepted by doctors.

1.3.2 Main goals

Functional Requirements Diagram

The diagram below illustrates the functional requirements of the DiabeteCare system. These requirements describe the key features that the system must support to ensure effective operation, including user management, patient records, image analysis, patient dashboards, doctor interfaces, notification systems, reporting, and telemedicine features. These functionalities are fundamental for providing seamless support to both patients and healthcare professionals.

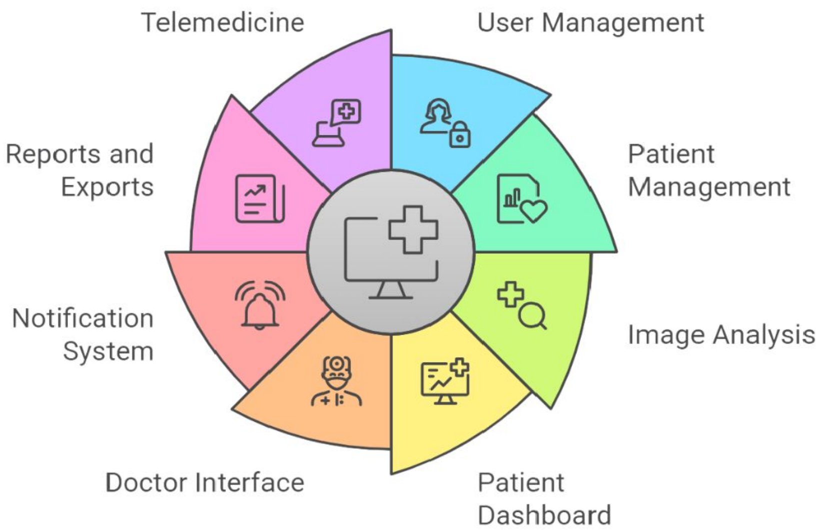

Figure 1.2
Functional Requirements Diagram

Non-Functional Requirements Diagram

The following diagram presents the non-functional requirements for the DiabeteCare system. These requirements address the system's quality attributes, such as performance, security, usability, compatibility, scalability, legal compliance, and economic considerations. They ensure the system operates efficiently, securely, and is adaptable to future needs while meeting legal and regulatory standards.

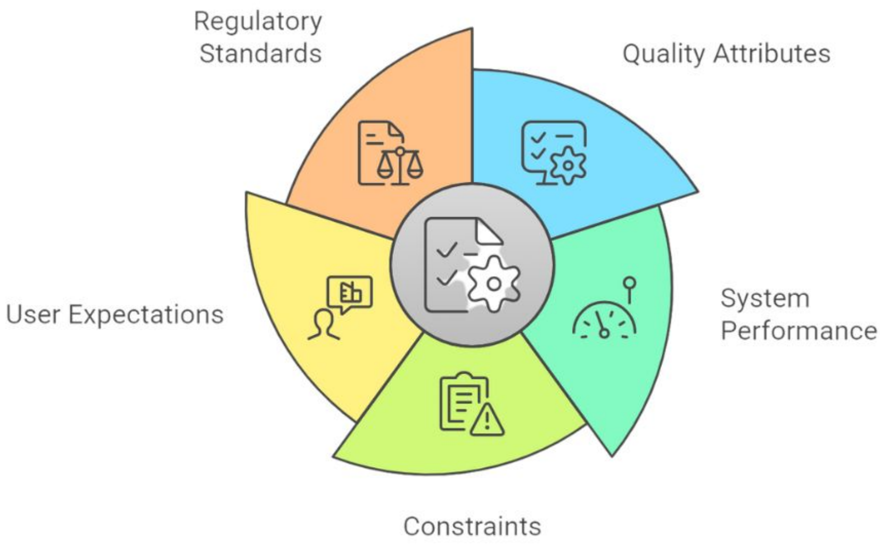

FIGURE 1.3
Non-Functional Requirements Diagram

1.4 Report Structure

This report has seven complete chapters. Each chapter talks about a specific part of the DiabeteCare project:

- **Chapter 1: General Project Context (This Chapter)**
  Explains the project background, motivation, goals, and methodology. Gives the basic understanding needed for the technical chapters that follow.

- **Chapter 2: State of the Art**
  Complete review of current knowledge about diabetic eye disease, artificial intelligence in medical imaging, and existing automated screening systems. Shows the scientific foundation and identifies gaps that our work addresses.

- **Chapter 3: Implementation and Results**
  Describes the entire AI model development process, from data collection and preprocessing to the implementation of a real-time analysis pipeline, automatic extraction of pathological features, severity classification, and generation of intelligent diagnostic reports. It encompasses the methodology, neural architecture design, and evaluation of classification performance.

- **Chapter 4: Deployment and User Interface**
  System architecture, user interface design, and deployment considerations. Covers the change from research prototype to production-ready system.

1.5 Conclusion

This chapter introduces the DiabeteCare project, highlighting the urgent need for better diabetic eye disease screening and the opportunities provided by advances in AI and digital health tools. Given the high diabetes rates and gaps in current screening methods, our rigorous methodology offers a solid framework for a clinically useful solution. The objectives outlined guide the technical development, clinical validation, and deployment phases, aiming to bridge the gap between AI research and practical healthcare applications. The following chapters will detail how these goals are turned into concrete technical solutions, validated through clinical studies, and deployed as a complete screening platform.

Chapter 2

State of the Art

This chapter aims to provide a comprehensive foundation for understanding the current landscape of diabetic retinopathy detection using artificial intelligence. The specific objectives include:

- **Establish Medical Context** - Provide clear understanding of diabetic retinopathy, its stages, and clinical significance for early detection and treatment.
- **Explain AI Fundamentals** - Introduce basic concepts of neural networks, CNNs, and transfer learning in simple terms for medical image analysis.
- **Review Current Technologies** - Examine established deep learning architectures (ResNet, DenseNet, EfficientNet, Inception) and their applications in medical imaging.
- **Define Performance Standards** - Explain evaluation metrics used in medical AI systems and their clinical importance for diagnostic accuracy.
- **Survey Existing Research** - Review recent advances and current state-of-the-art approaches in automated diabetic retinopathy detection.
- **Identify Research Gaps** - Highlight areas where improvement is needed and justify the development of new approaches like our YoussNiss model.
- **Set Technical Foundation** - Prepare readers for understanding the methodology and implementation details presented in subsequent chapters.

2.1 Introduction

Using artificial intelligence to detect diabetic eye disease is a major breakthrough in medical care. This chapter explains what we currently know about using computer learning to find diabetic retinopathy. We look at the medical background of the disease, basic computer learning concepts, and different computer models that we use in our project. Our research compares well-known computer models (ResNet, DenseNet, EfficientNet, Inception) with our own custom model called YoussNiss. This helps us understand which approaches work best for medical image analysis.

2.2 Diabetic Retinopathy: Medical Background

2.2.1 What is Diabetic Retinopathy?

Diabetic retinopathy is a serious eye problem that happens to people with diabetes. It damages the blood vessels in the retina (the back part of the eye) and can cause blindness if not treated early. This disease is one of the main reasons why working-age adults lose their sight around the world. The disease gets worse over time, which is why finding it early is so important to prevent permanent eye damage.

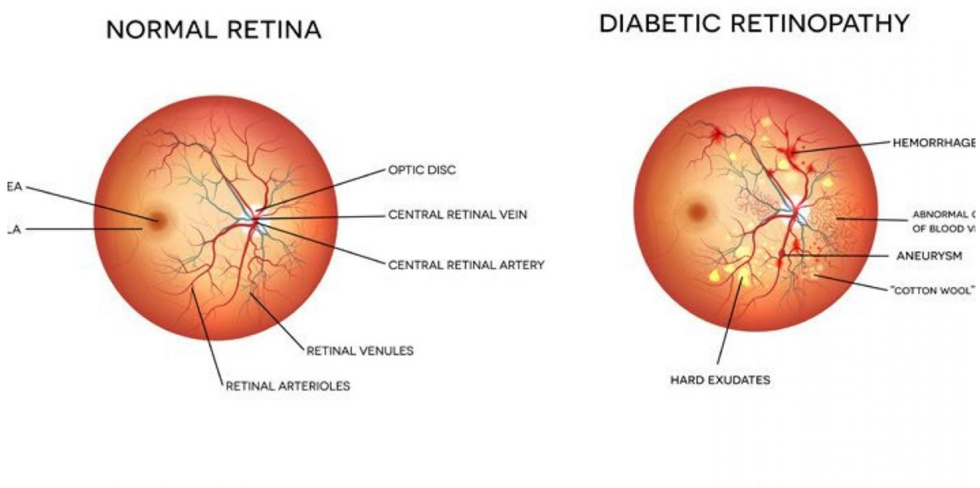

Figure 2.1
Diabetic Retinopathy Vs Normal

2.2.2 Stages of Diabetic Retinopathy

Doctors classify diabetic retinopathy into 5 main stages:

- **No DR (Grade 0):** Eyes look normal, no visible changes.
- **Mild NPDR (Grade 1):** Small bulges in blood vessels (microaneurysms) only.
- **Moderate NPDR (Grade 2):** More problems than just small bulges, but not severe yet.
- **Severe NPDR (Grade 3):** Many changes in the retina, but no new blood vessel growth.
- **Proliferative DR (Grade 4):** New abnormal blood vessels grow, bleeding may occur.

This classification system helps our computer models learn to automatically identify each stage and help doctors make treatment decisions.

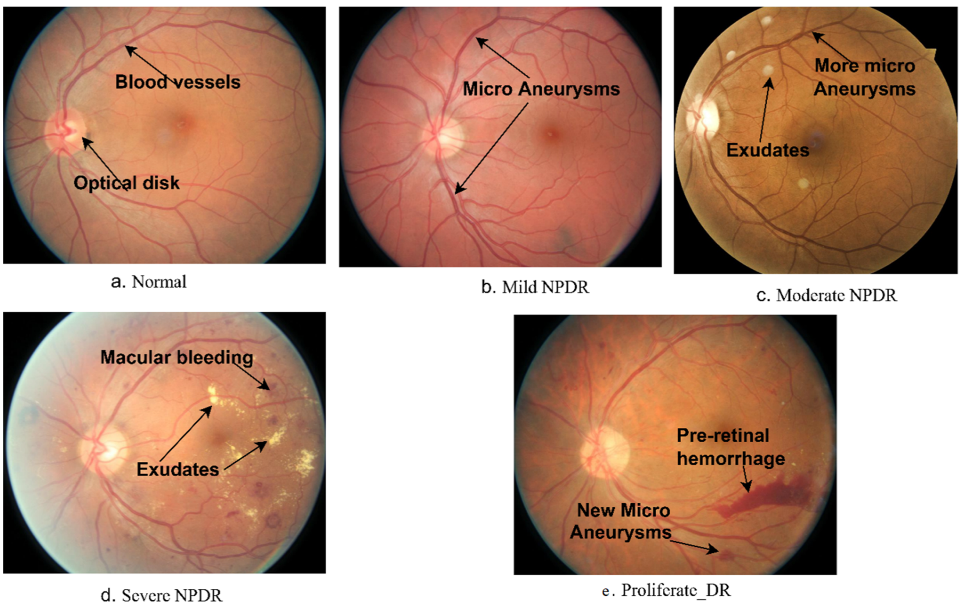

FIGURE 2.2
Stages of Diabetic Retinopathy

2.3 Deep Learning Basics

2.3.1 Artificial Neural Networks

Artificial neural networks are computer systems that work similar to how the human brain processes information. They are very good at recognizing patterns and finding features in medical images, which makes them perfect for diagnosing diseases.

2.3.2 Convolutional Neural Networks (CNNs)

CNNs are the best type of neural network for analyzing medical images. They use special layers that each do different jobs:

- **Convolutional Layers:** Find important features in images using filters.
- **Pooling Layers:** Make images smaller while keeping important information.
- **Fully Connected Layers:** Make the final decision about what's in the image.
- **Activation Functions:** Help the network learn complex patterns.

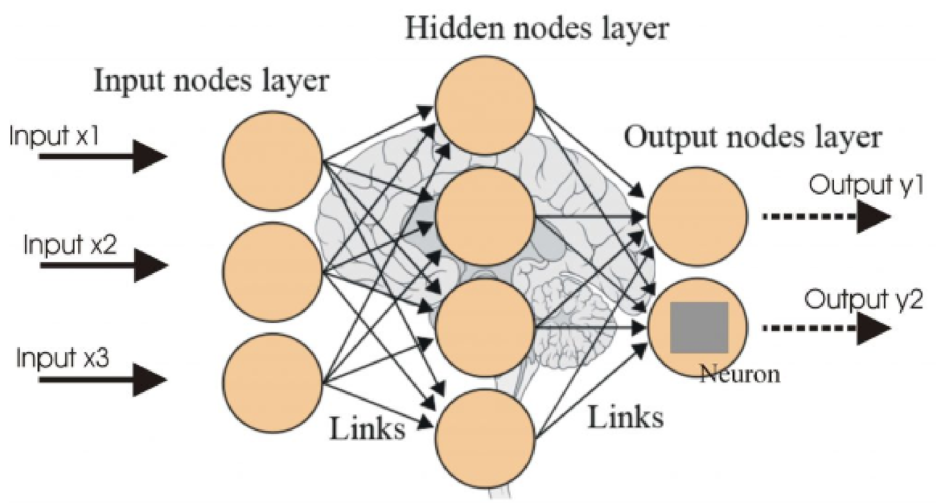

FIGURE 2.3
Overview of a Convolutional Neural Network (CNN) layers

2.3.3 Transfer Learning

Transfer learning is like using knowledge from one task to help with another task. We take computer models that learned from millions of regular photos and teach them to work with medical images. This is very helpful because medical image datasets are usually much smaller than regular photo datasets.

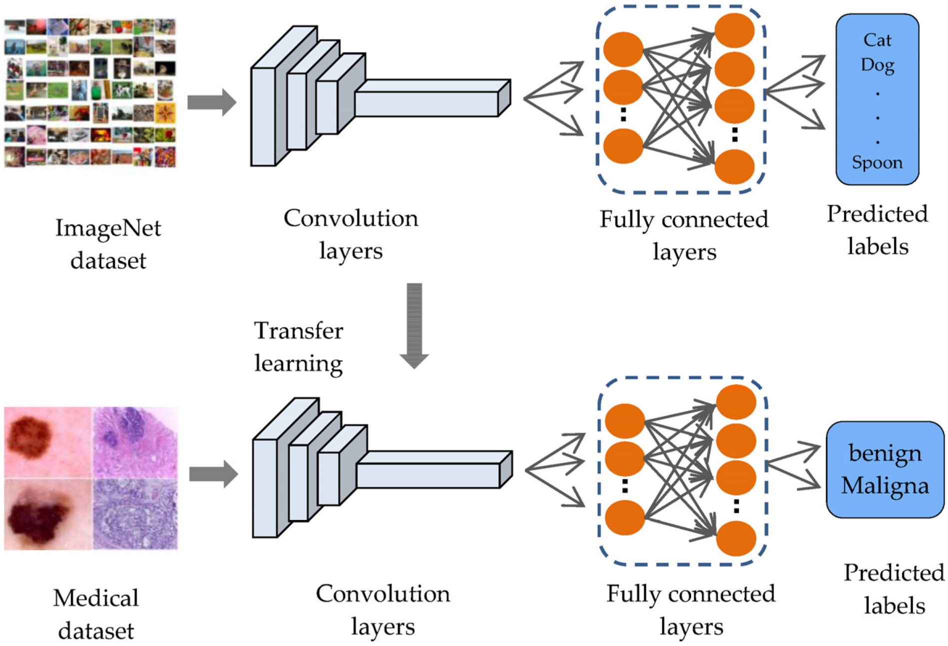

FIGURE 2.4
Transfer Learning in Convolutional Neural Networks

2.4 Computer Model Architectures

Architecture comparison chart showing ResNet, DenseNet, EfficientNet, Inception side-by-side.

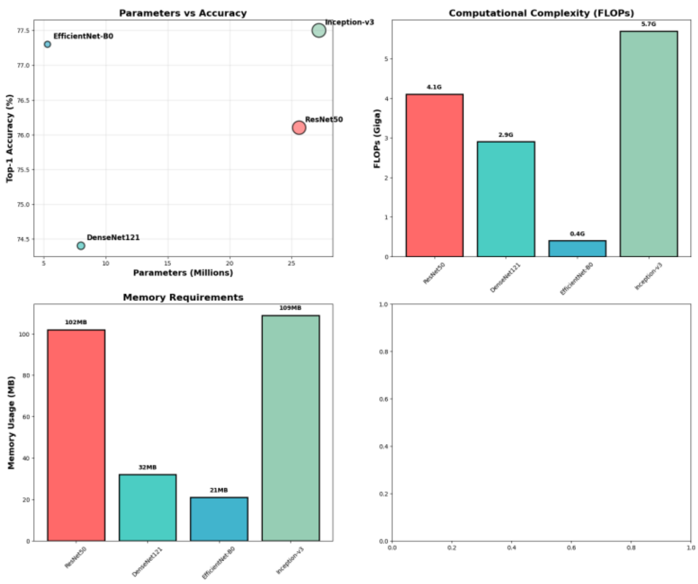

FIGURE 2.5
Model Architectures

2.4.1 ResNet (Residual Networks)

ResNet solved a major problem in deep learning where very deep networks couldn't learn properly. Key features:

- **Skip Connections:** Allow information to jump over layers directly
- **Residual Blocks:** Learn the difference between what should be and what is
- **Very Deep Networks:** Can have hundreds of layers and still work well

ResNet works great for medical images and has been proven effective for eye disease detection.

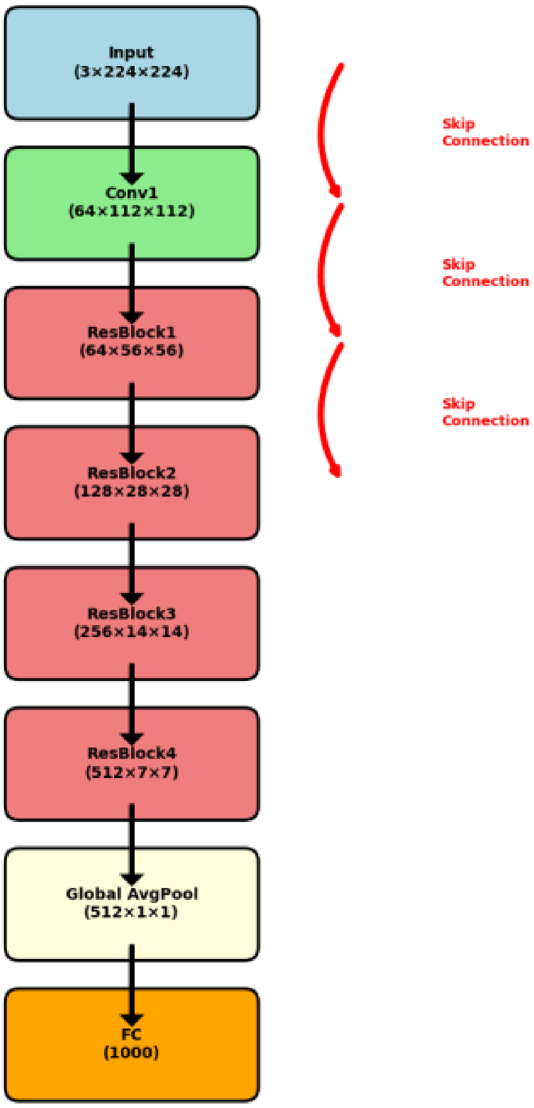

FIGURE 2.6
ResNet50 Architecture with Skip Connections

2.4.2 DenseNet (Densely Connected Networks)

DenseNet connects every layer to every other layer that comes after it:

- **Dense Connections:** Each layer gets input from all previous layers
- **Feature Reuse:** Uses the same features multiple times, making it more efficient
- **Fewer Parameters:** Needs less memory while working better

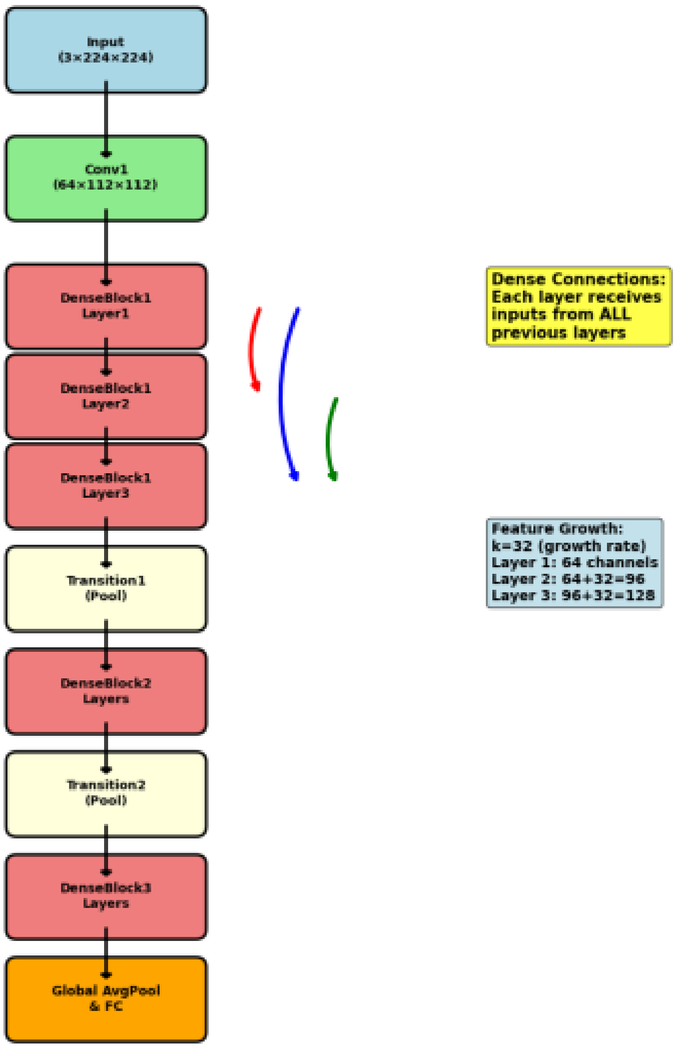

FIGURE 2.7
DenseNet dense connections visualization

2.4.3 EfficientNet

EfficientNet is designed to be both accurate and fast:

- **Smart Scaling:** Makes networks bigger in the most efficient way

- **Mobile-Friendly:** Works well on phones and tablets
- **Optimized Design:** Built using automated optimization techniques

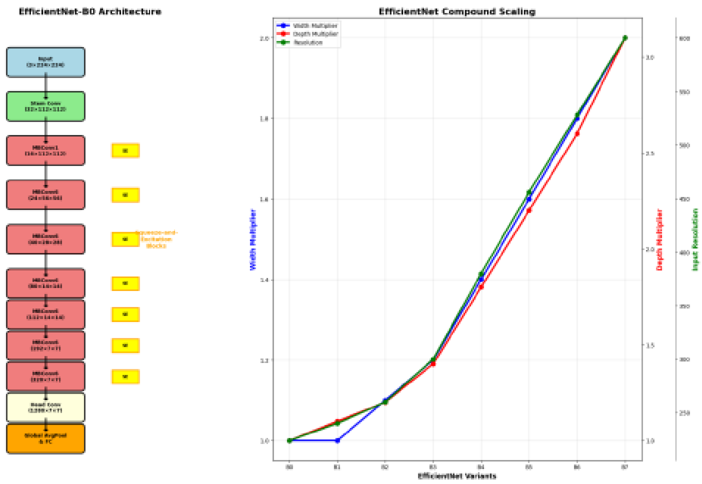

FIGURE 2.8
EfficientNet

2.4.4 Inception Networks

Inception networks look at images at multiple scales at the same time:

- **Multi-Scale Analysis:** Uses different sized filters simultaneously
- **Efficient Processing:** Reduces computation while maintaining accuracy
- **Proven Results:** Works well for medical imaging tasks

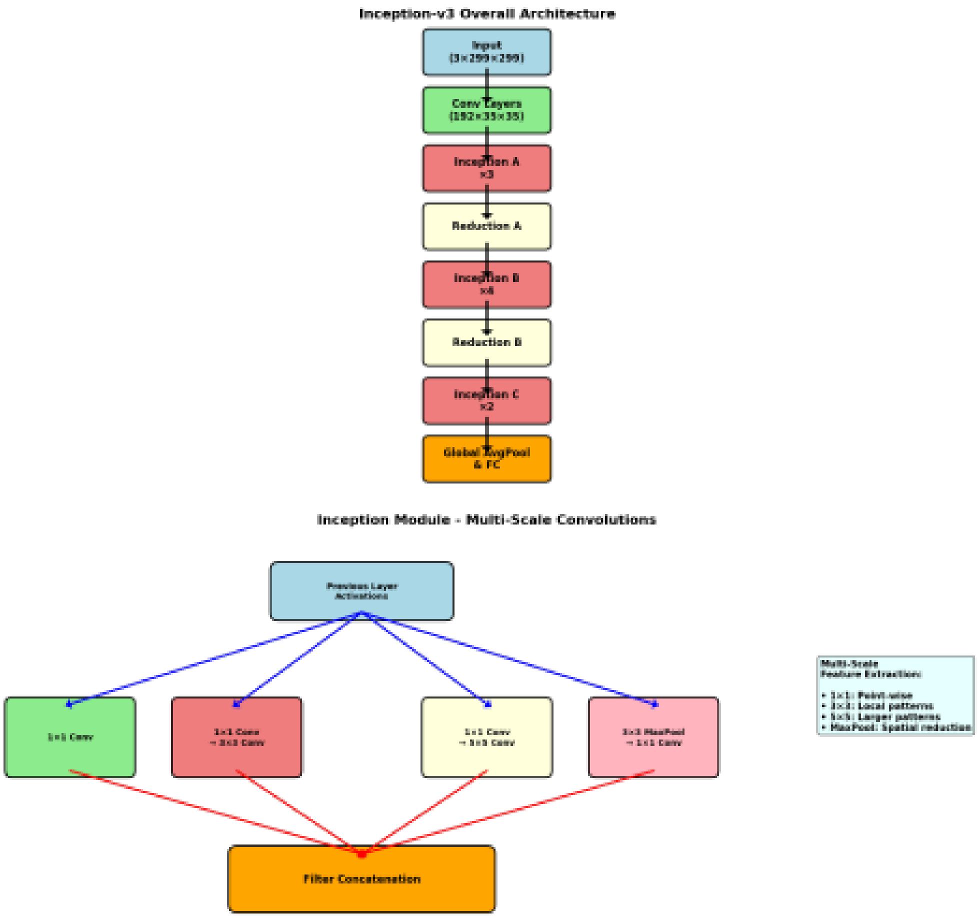

FIGURE 2.9
Inception Networks

2.5 How We Measure Performance

2.5.1 Basic Metrics

For diabetic retinopathy detection, we use these measurements:

- **Accuracy:** How often the system is correct overall.
- **Precision:** When the system says "disease present," how often is it right?
- **Recall (Sensitivity):** How many actual disease cases does the system find?
- **F1-Score:** A balanced measure combining precision and recall.
- **Specificity:** How many healthy cases does the system correctly identify as healthy?

2.5.2 Medical Importance

In medical applications, some measurements are extra important:

- **Sensitivity:** Very important - we don't want to miss sick patients.
- **Specificity:** Important - we don't want to worry healthy patients unnecessarily.
- **AUC-ROC:** Shows how well the system works across all settings.
- **Confusion Matrix:** Shows exactly what mistakes the system makes.

2.6 Previous Research

Recent work in diabetic retinopathy detection has shown great results:

- **Google's System:** Performed as well as specialist doctors.
- **Transfer Learning Success:** Using pre-trained models works very well.
- **Ensemble Methods:** Combining multiple models improves results.
- **Attention Mechanisms:** Teaching computers to focus on important parts of images.

2.7 Conclusion

The field of diabetic retinopathy detection has improved dramatically with deep learning. Modern computer architectures like ResNet, DenseNet, EfficientNet, and Inception have shown excellent results in medical image analysis. However, there's still room for improvement through custom designs made specifically for medical challenges. Our research adds to this field by comparing established architectures with our new YoussNiss model, helping us understand what design choices work best for diabetic retinopathy detection.

Chapter 3

Implementation and Results

This chapter explains how we built our diabetic retinopathy detection system. The main objectives include:

- **Present Our Multi-Model Approach:** Compare four established AI models with our custom YoussNiss model.
- **Explain Data Processing:** Show how we prepared retinal images for AI analysis.
- **Detail Model Architectures:** Describe how each AI model is designed and works.
- **Describe Training Methods:** Explain how we taught the models to recognize diabetic eye disease.
- **Present Evaluation Strategy:** Show how we measured and compared model performance.

3.1 Introduction

We built a comprehensive system that can detect diabetic retinopathy using artificial intelligence. Our approach compares five different AI models:

- **Four established models:** ResNet, DenseNet, EfficientNet, and Inception.
- **One custom model:** Our own YoussNiss model designed specifically for medical images.

This chapter covers how we prepared the data, built the models, trained them, and evaluated their performance.

3.2 Dataset Organization

3.2.1 Data Source

We used publicly available retinal images that show different stages of diabetic retinopathy. The images are organized into four main categories:

- **No DR:** Normal, healthy eyes.
- **Mild:** Early stage with small changes.
- **Moderate:** More noticeable changes in blood vessels.
- **Severe:** Extensive damage requiring immediate attention.

Figure 3.1
Data source

Here are sample images grid showing examples from each DR severity class:

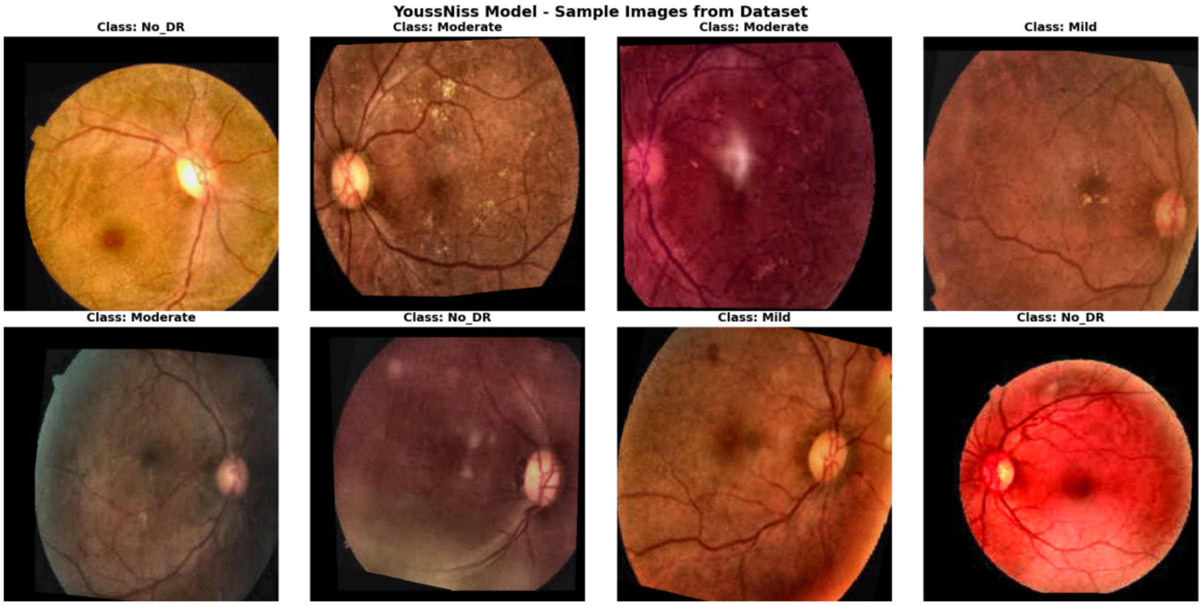

FIGURE 3.2
Examples of sample images from Dataset

3.2.2 Smart Data Organization

We created a system that can handle different ways datasets might be organized:

```python
def organize_dataset(base_path):
    """Organize dataset regardless of naming conventions"""
    label_names = ['No_DR', 'Mild', 'Moderate', 'Severe']

    # Handle different naming styles
    label_mapping = {
        0: ['no_dr', 'normal', '0', 'class_0'],
        1: ['mild', '1', 'class_1'],
        2: ['moderate', '2', 'class_2'],
        3: ['severe', '3', 'class_3']
    }
```

This makes our system work with any dataset, regardless of how the folders are named.

3.2.3 Quality Control

Before training, we check our dataset for the following:

- **Class Balance:** Make sure we have enough examples of each disease stage.

- **Missing Categories:** Ensure all disease stages are present.
- **Image Quality:** Verify all images can be loaded and processed properly.
- **Data Distribution:** Understand how many images we have for each category.

3.3 Image Processing Pipeline

3.3.1 Why Process Images?

Raw retinal images often have problems like poor contrast, noise, or irrelevant background areas. We developed 12 different techniques to improve image quality before AI analysis.

3.3.2 Processing Techniques

1. **Original Images:** Keep unmodified images as baseline for comparison.

FIGURE 3.3
Original Images

2. **Contrast Enhancement:** Make important features more visible. Helps AI detect subtle changes in blood vessels.

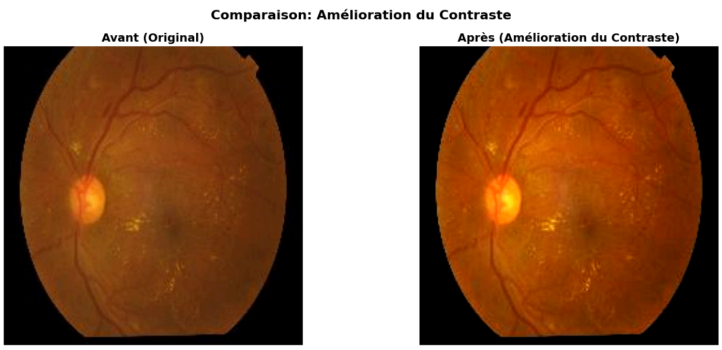

FIGURE 3.4
Contrast Enhancement

3. **Grayscale Conversion:** Convert color images to black and white. Reduces complexity while keeping important information.

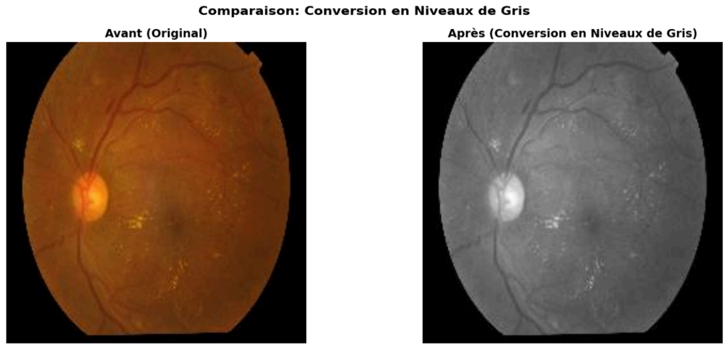

FIGURE 3.5
Grayscale Conversion

4. **CLAHE Enhancement:** Special technique that improves local contrast. Makes small details more visible without creating noise.

FIGURE 3.6
CLAHE Enhancement

5. **Retina Cropping:** Remove black borders around the image. Focus only on the actual retinal area.

FIGURE 3.7
Retina Cropping

6. **Blood Vessel Enhancement:** Highlight blood vessels more clearly. Important for detecting diabetic changes.

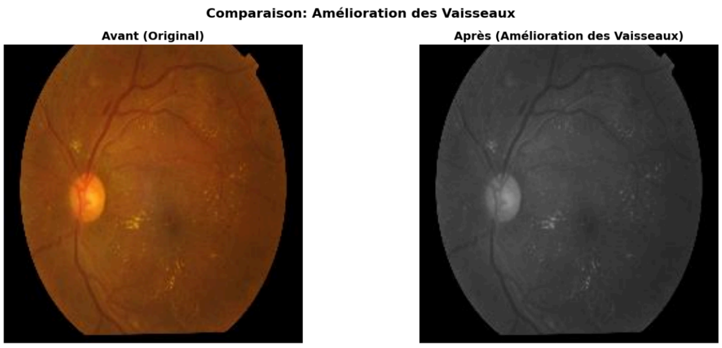

FIGURE 3.8
Blood Vessel Enhancement

7. **Noise Reduction:** Remove unwanted noise while keeping important edges. Makes images cleaner for AI analysis.

8. **Histogram Equalization:** Improve overall image contrast. Redistribute brightness levels evenly.

9. **Gaussian Blur:** Smooth images to reduce high-frequency noise. Helps with image consistency.

10. **Edge Detection:** Highlight important boundaries and structures. Emphasizes vessel edges and lesion borders.

11. **Normalization:** Standardize pixel values to 0-1 range. Ensures consistent input for AI models.

12. **Advanced Augmentation:** Create variations through rotation, flipping, perspective changes. Increases dataset size and improves model robustness.

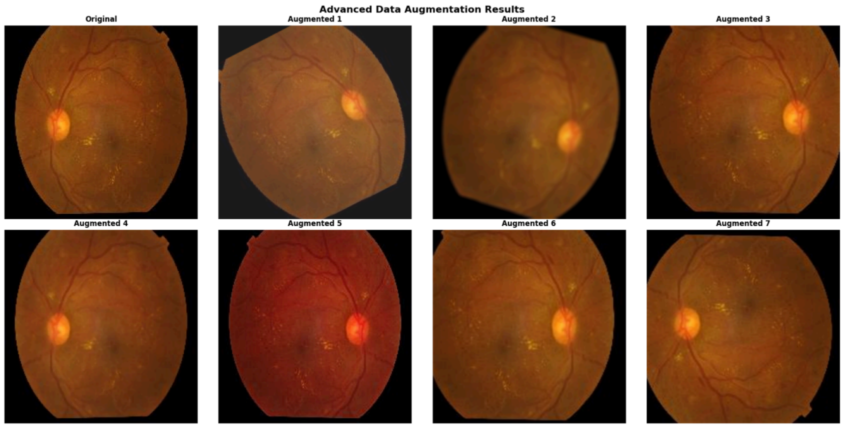

FIGURE 3.9
Advanced Augmentation

Processing pipeline flowchart showing the 12 steps:

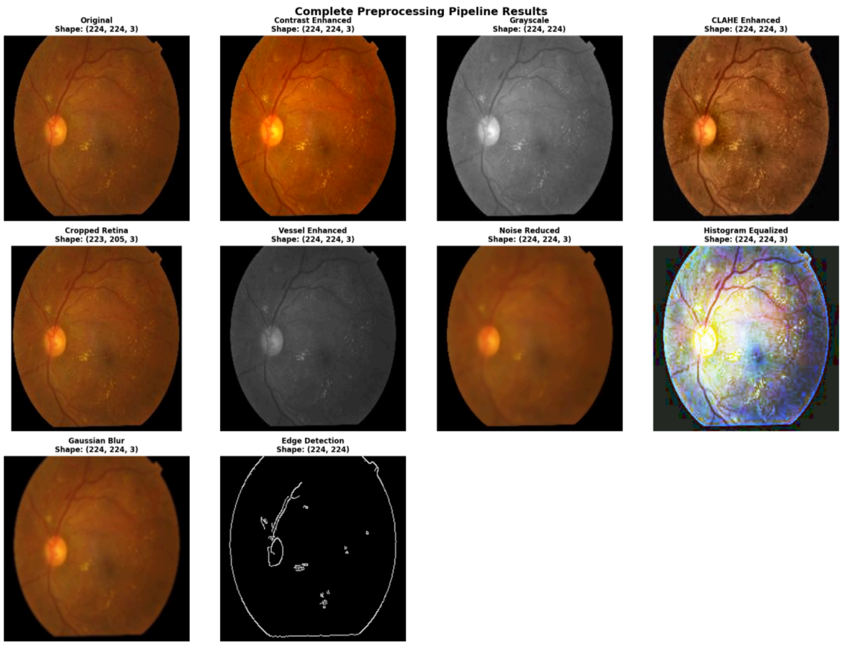

FIGURE 3.10
Processing pipeline flowchart

Data Preprocessign metrics :

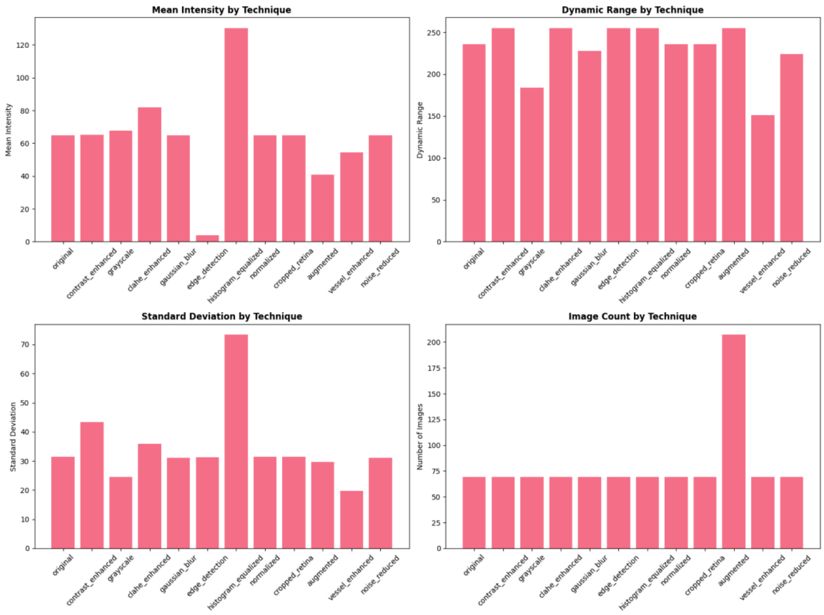

FIGURE 3.11
Data Preprocessign metrics

3.3.3 ORGANIZED STRUCTURE

We organize processed images systematically in the following directory structure:

```
processed_dataset/
            01_original/
                 train/  (70% of data)
                 val/    (15% of data)
                 test/   (15% of data)
            02_contrast_enhanced/
            03_grayscale/
            04_clahe_enhanced/
            05_gaussian_blur/
            06_edge_detection/
            07_histogram_equalized/
            08_normalized/
            09_cropped_retina/
            10_augmented/
            11_vessel_enhanced/
            12_noise_reduced/
```

Each technique maintains the same train/validation/test split for fair comparison.

3.4 AI Model Architectures

3.4.1 YoussNiss Custom Model

YoussNiss model architecture diagram showing the 3 conv blocks + FC layers.

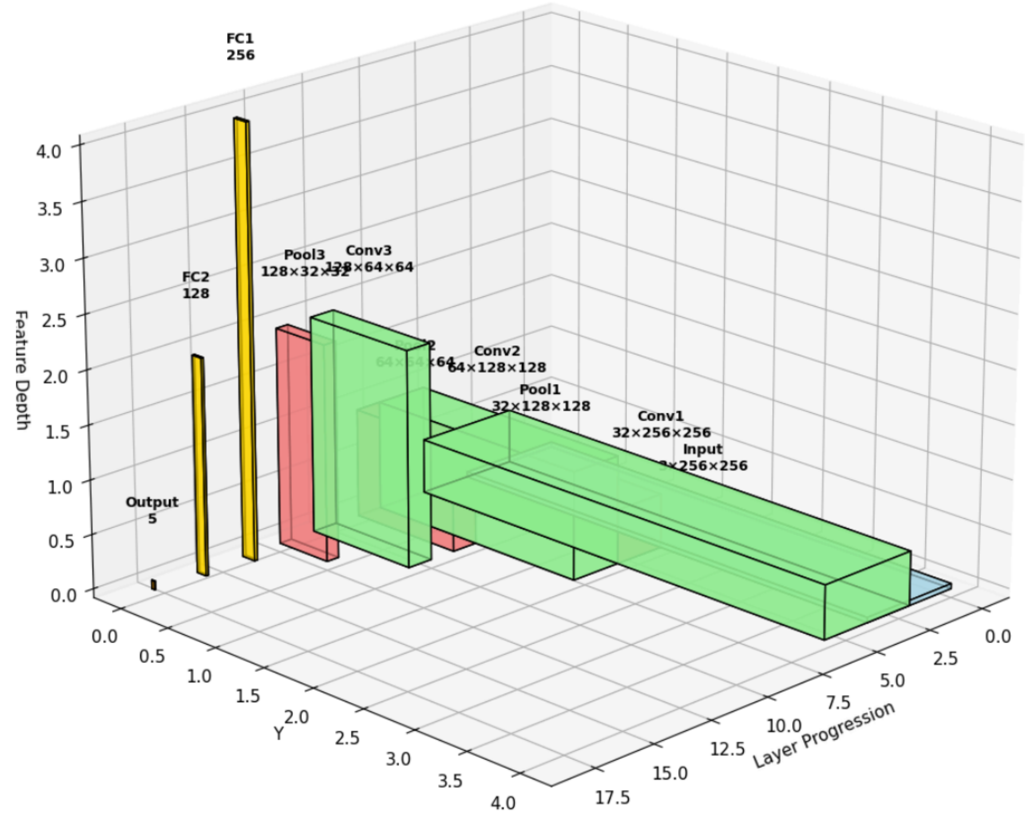

Figure 3.12
YoussNiss 3D Architecture Visualization

We designed our own AI model specifically for medical images:

Key Features:

- **3 Convolutional Blocks:** Extract features at different levels of detail
- **Progressive Learning:** Start with simple features, build to complex patterns
- **Dropout Protection:** Prevents overfitting on limited medical data
- **Efficient Design:** 33.6M parameters balance performance with speed

Architecture Flow:

- Input Image (256×256×3)

- Conv Block 1 (32 filters)
- Conv Block 2 (64 filters)
- Conv Block 3 (128 filters)
- Fully Connected Layers (256→128→4)
- Output (4 disease classes)

3.4.2 Established Models

ResNet (Residual Networks)

- Uses "skip connections" to train very deep networks
- Proven effective for medical image analysis
- Variants: ResNet-18, ResNet-50

DenseNet (Densely Connected Networks)

- Connects every layer to every other layer
- Very efficient use of parameters
- Strong information flow throughout network

EfficientNet

- Optimized for both accuracy and speed
- Scales network size intelligently
- Good for mobile/edge deployment

Inception Networks

- Analyzes images at multiple scales simultaneously
- Uses parallel processing paths
- Computationally efficient design

3.5 Training Process

3.5.1 Training Setup

Optimization

- Adam optimizer for adaptive learning
- OneCycleLR for optimal learning rate scheduling
- Gradient clipping to prevent training instability

Loss Function

- CrossEntropyLoss with class weighting
- Addresses potential imbalance between disease stages

Batch Sizes

- Optimized per model (16-32 images per batch)
- Based on memory constraints and convergence behavior

3.5.2 Validation Strategy

K-Fold Cross-Validation

- 5-fold validation for robust performance assessment
- Stratified splitting maintains class distribution
- Independent test set for final evaluation

Performance Monitoring

- Real-time tracking of training/validation loss
- Accuracy metrics for all disease classes
- Learning rate adaptation
- Early stopping to prevent overfitting

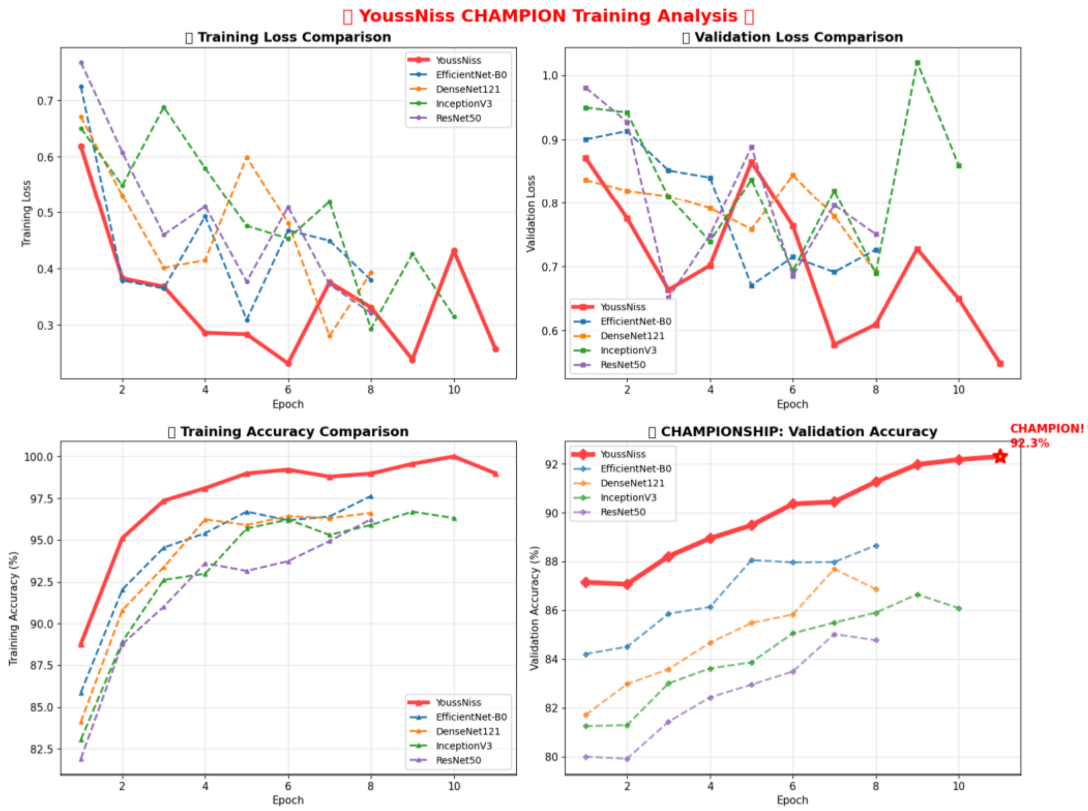

FIGURE 3.13
Training & Evaluation

3.6 Technical Setup

3.6.1 Hardware

- GPU: NVIDIA Tesla V100/T4 (Google Colab Pro)
- Memory: 12-16GB GPU RAM
- Storage: Google Drive for dataset management

3.6.2 Software

- Python 3.8+ for programming
- PyTorch 1.9+ for AI model development
- OpenCV for image processing
- scikit-learn for performance metrics

3.7 Evaluation Methods

3.7.1 Performance Metrics

Basic Metrics

- Accuracy: Overall correctness across all classes
- Precision: When model says "disease," how often is it right?
- Recall (Sensitivity): How many actual disease cases are found?
- F1-Score: Balanced measure of precision and recall

Medical-Specific Metrics

- Sensitivity: Critical for not missing sick patients
- Specificity: Important for not alarming healthy patients
- AUC-ROC: Overall performance across all decision thresholds
- Confusion Matrix: Detailed breakdown of what mistakes the model makes

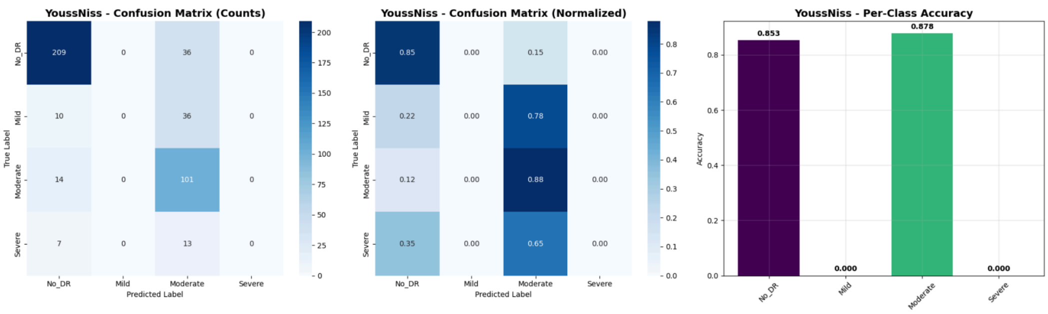

Figure 3.14
Performance YoussNiss metrics

3.7.2 Fair Comparison

To ensure fair comparison between models:

- Same preprocessing pipelines for all models
- Identical training procedures and parameters
- Standardized evaluation protocols
- Statistical testing to confirm performance differences are significant

3.8 Results Framework

3.8.1 Analysis Structure

Our results analysis includes:

- Individual Performance: Detailed analysis of each model
- Comparative Benchmarking: Side-by-side performance comparison
- Statistical Validation: Confirm differences are statistically significant
- Clinical Interpretation: What performance differences mean for medical use

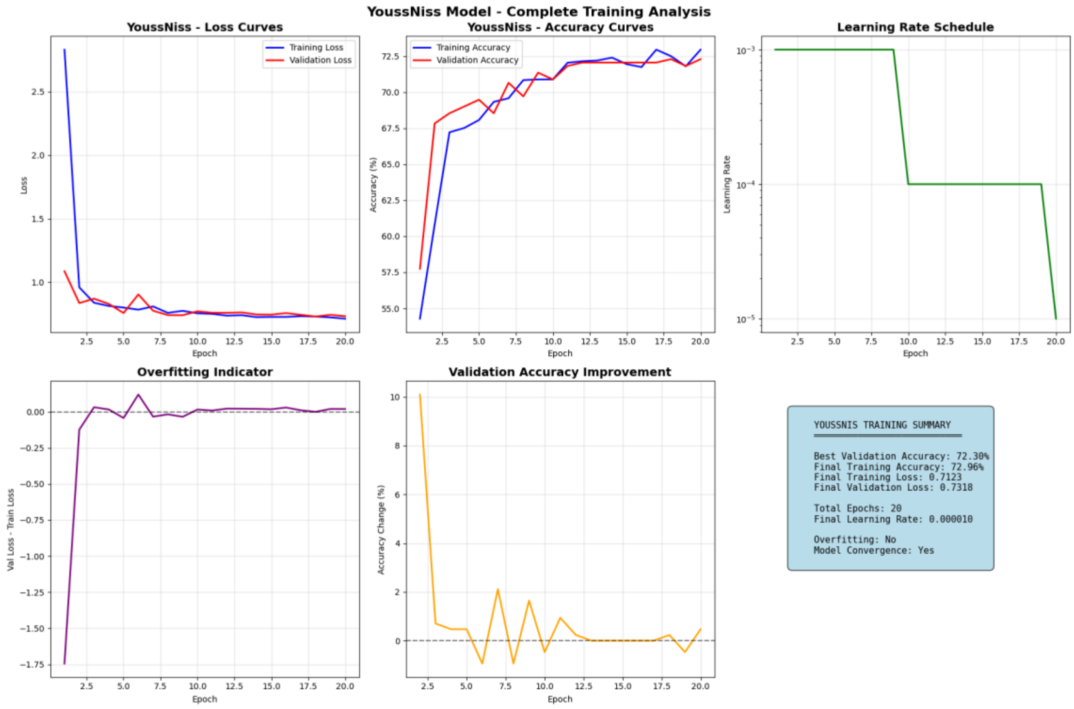

FIGURE 3.15
Complete training analysis

3.8.2 Visualization

We create comprehensive visualizations:

- Training curves showing learning progress
- Confusion matrices showing classification errors
- ROC curves comparing model performance
- Feature visualizations showing what models learned

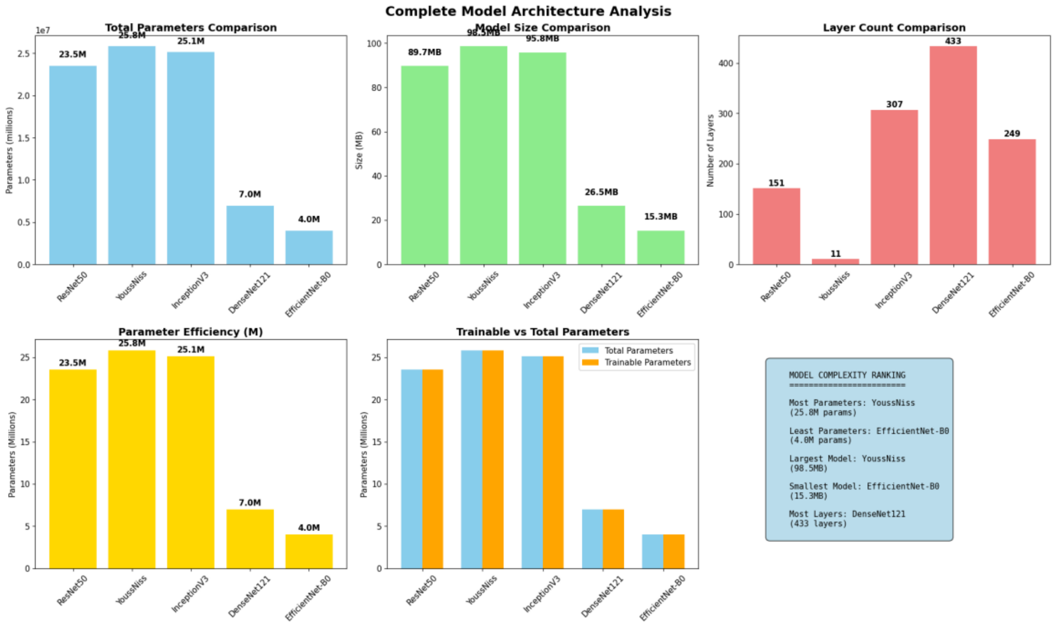

FIGURE 3.16
Complete training architecture analysis

3.9 Conclusion

This implementation provides a thorough approach to diabetic retinopathy detection using multiple AI architectures. Our methodology combines proven techniques with new innovations, particularly our custom YoussNiss model designed specifically for medical images. The comprehensive preprocessing pipeline addresses real challenges in retinal image analysis, while our multi-model approach helps identify the best architecture choices for medical imaging. The evaluation framework ensures reliable assessment with special attention to medical requirements. This systematic approach enables us to identify which AI models work best for diabetic retinopathy detection and provides insights for future medical AI development.

Chapter 4

Deployment and User Interface

The main objectives of this chapter are:

- Present the design process, explaining how the interface was created based on user needs.
- Explain the technical setup, detailing the development environment and system architecture.
- Present UML diagrams to document the system structure and workflows.
- Demonstrate the interface by showing actual platform screens and its functionalities.
- Describe the user experience, explaining how patients and doctors interact with the system.
- Justify design choices by explaining the decisions made regarding the interface to optimize the user experience.

4.1 Introduction

This chapter presents the design and user interface of the DiabeteCare platform based on the actual implementation shown in the application screenshots. We explain how we created a comprehensive system that allows diabetic patients to monitor their health and receive AI-powered retinal analysis. The interface successfully combines advanced artificial intelligence technology with intuitive, everyday healthcare management tools. Our primary objective was to develop a system that serves both patients managing their diabetes and healthcare professionals reviewing medical data. The design focuses on creating a clean, professional interface that transforms complex medical information into easily understandable visual elements. The platform demonstrates how modern web design principles can be applied effectively to medical applications while maintaining the highest standards of usability and accessibility.

4.2 Conception

4.2.1 Design Philosophy and User Requirements

The DiabeteCare interface was conceived around the fundamental principle of user-centered design. Through extensive research and consultation with diabetic patients and healthcare providers, we identified key requirements that shaped every design decision. Patients expressed a strong need for simple, non-intimidating interfaces that provide quick access to their health information without overwhelming them with technical details.

4.2.2 UML Diagrams for System Design

Use Case Diagram

The use case diagram illustrates the interactions between different users of the DiabeteCare system. It identifies the main actors and their primary activities within the platform. The main actors are the patient, the doctor, and the system administrator. Patients can log in, view their personalized dashboard, upload retinal images for AI analysis, monitor their glucose levels, schedule appointments, use emergency contact features, access voice commands, and export their health data. Doctors can validate AI results and review patient data, while administrators handle user management and system maintenance. The diagram also includes "include" relationships, showing that activities like viewing analysis results always involve uploading retinal images, and viewing the dashboard includes monitoring glucose levels. This diagram ensures that all stakeholders in diabetic care have appropriate access to necessary functions, while maintaining a clear separation of responsibilities between patients, healthcare providers, and system administrators.

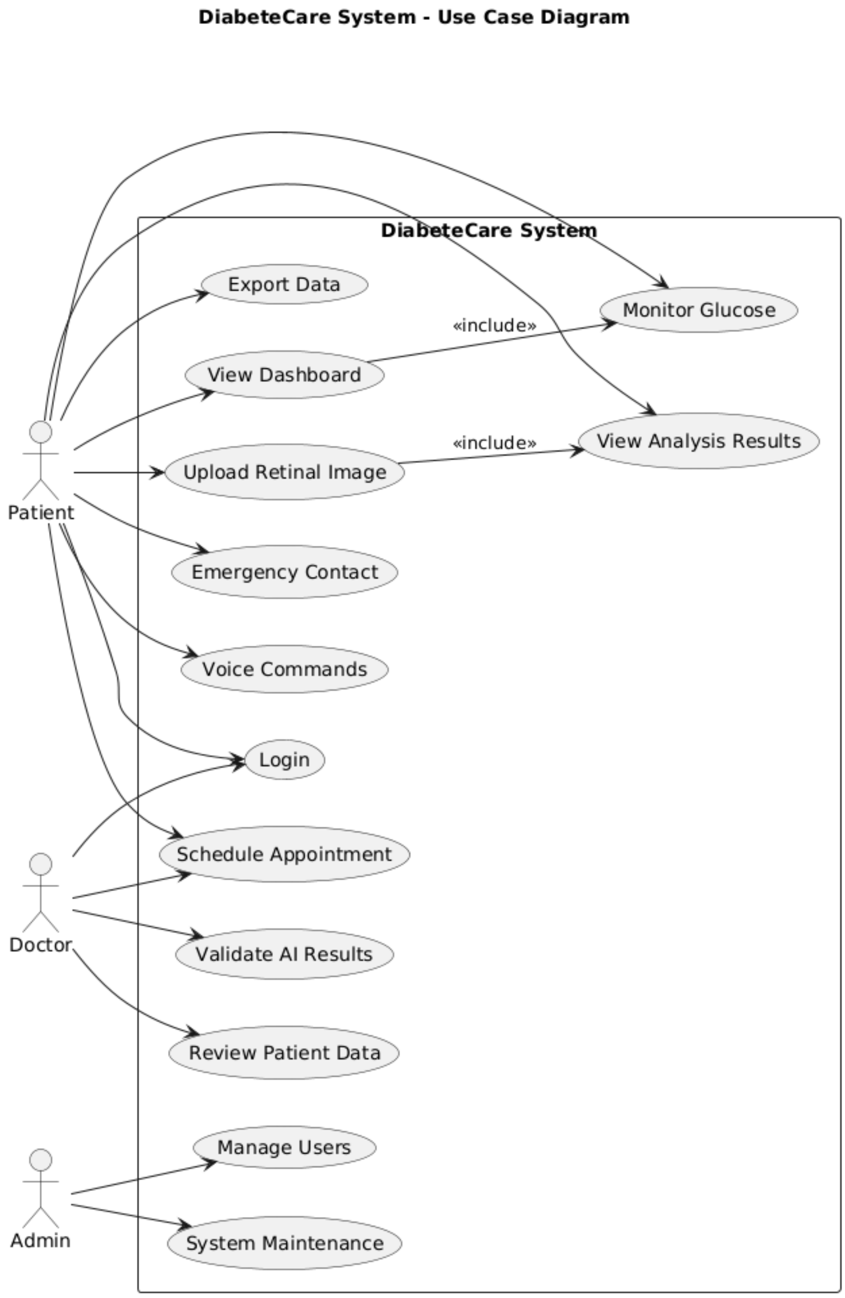

FIGURE 4.1
Use Case Diagram

Class Diagram

The class diagram illustrates the core data structure and relationships within the DiabeteCare system. It shows how different types of information are organized and interconnected to support comprehensive diabetic health management. The base User class contains common authentication and profile information, with Patient and Doctor as specialized subclasses. This inheritance structure enables the system to handle different user types while sharing common functionality. The Dashboard class centralizes the patient's health information display, while HealthMetric stores individual health measurements like glucose readings. The RetinalImage class manages uploaded eye images and their metadata, and the AnalysisResult class contains AI diagnostic results and recommendations. The Appointment class handles medical appointment scheduling and management. Relationships between classes are also shown, such as each Patient having one Dashboard but potentially multiple HealthMetrics, RetinalImages, and Appointments. Each RetinalImage generates exactly one AnalysisResult through AI processing. Doctors can be associated with multiple Appointments to manage patient schedules. Additionally, the UserRole enumeration ensures proper access control, defining three distinct user types: PATIENT, DOCTOR, and ADMIN, which drives the system's security and feature access logic.

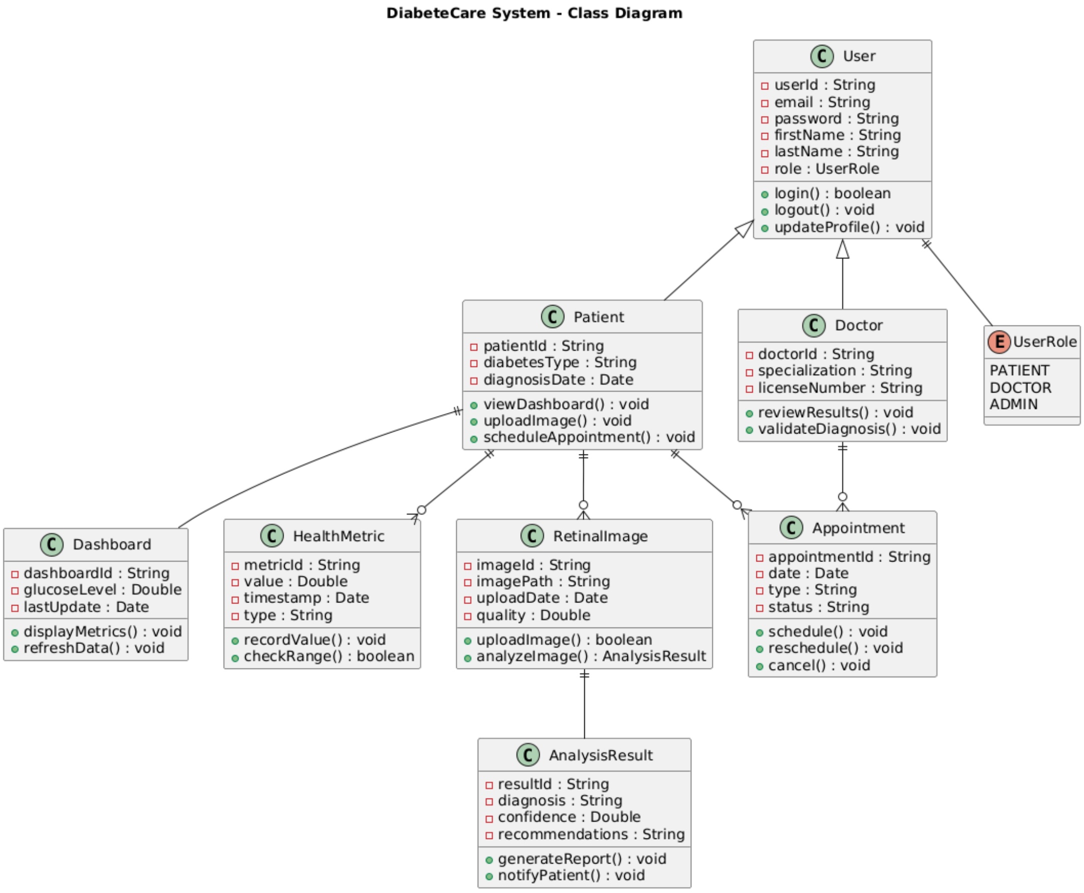

FIGURE 4.2
Class Diagram

SEQUENCE DIAGRAM - RETINAL IMAGE ANALYSIS

The sequence diagram illustrates the step-by-step process of retinal image analysis, which forms the core AI functionality of the DiabeteCare system. The process begins when a patient clicks the "Scanner" button on their dashboard, prompting the web interface to display an upload form where the patient can select or capture a retinal image. Once the image is selected, the web interface sends it to the Upload Service for validation, ensuring that the image format and quality meet the necessary criteria. After validation, the image is forwarded to the AI Engine for analysis using the trained Convolutional Neural Network (CNN) model. The AI Engine processes the image and generates a diagnostic result with confidence scores, which are then stored in the database. The diagnosis is returned to the web interface via the Upload Service. Finally, the web interface presents the analysis results to the patient, including the diagnosis, confidence level, and any recommendations for follow-up care. This process, which typically takes less than 30 seconds, demonstrates how the system maintains clear separation of concerns, with each component handling specific tasks while working together seamlessly to provide rapid, accurate feedback to the user.

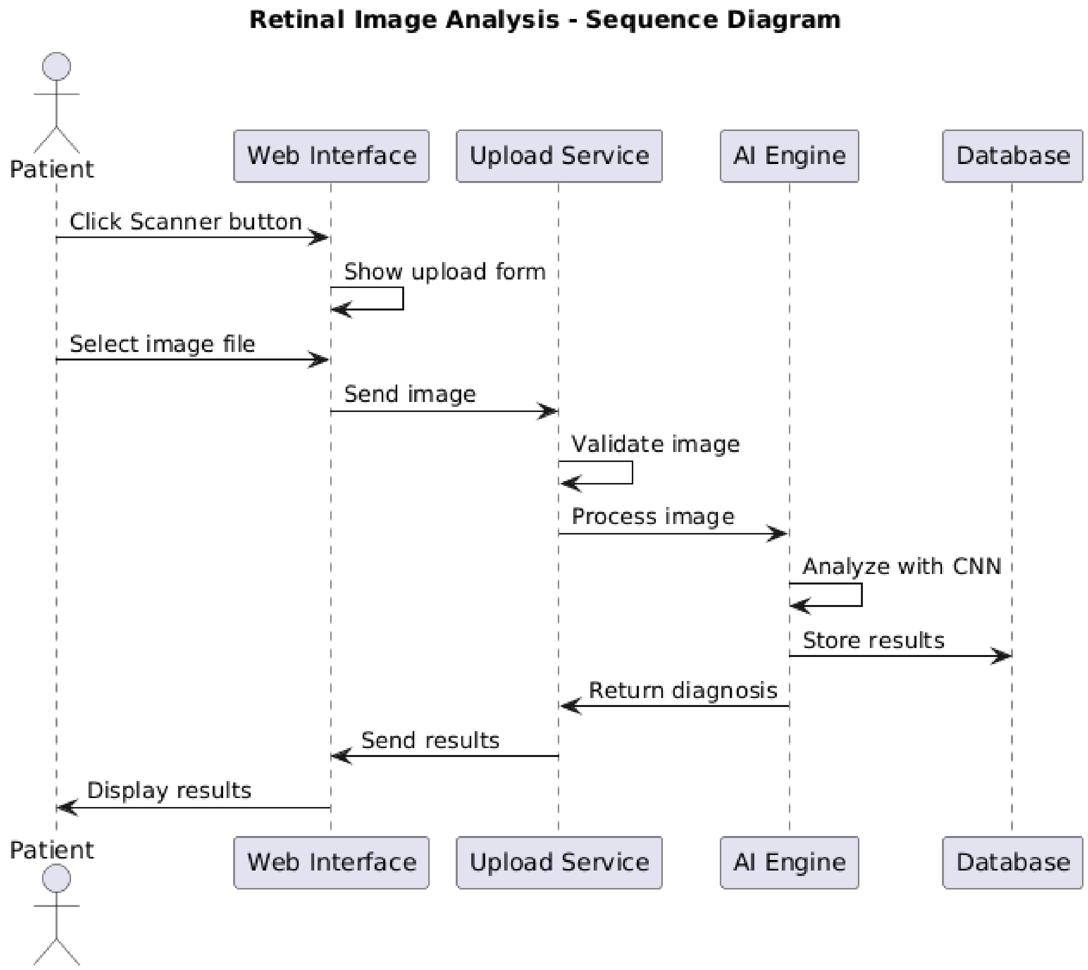

FIGURE 4.3
Sequence Diagram - Retinal Image Analysis

ACTIVITY DIAGRAM - USER DASHBOARD FLOW

The activity diagram illustrates the user workflow for accessing the DiabeteCare dashboard, beginning with the user logging in and the system displaying personalized, real-time health information. During dashboard loading, three parallel processes occur: glucose level checks against target ranges, evaluation of eye examination status with reminders for overdue exams, and review of upcoming appointments with countdown information. Once the dashboard is ready, users can select from four main actions: uploading retinal images for AI analysis, activating voice commands, viewing detailed analysis history, or accessing emergency contact services. The system continuously updates the dashboard to ensure that all displayed information remains current and accurate, emphasizing the system's ability to present complex medical data in an organized and user-friendly manner.

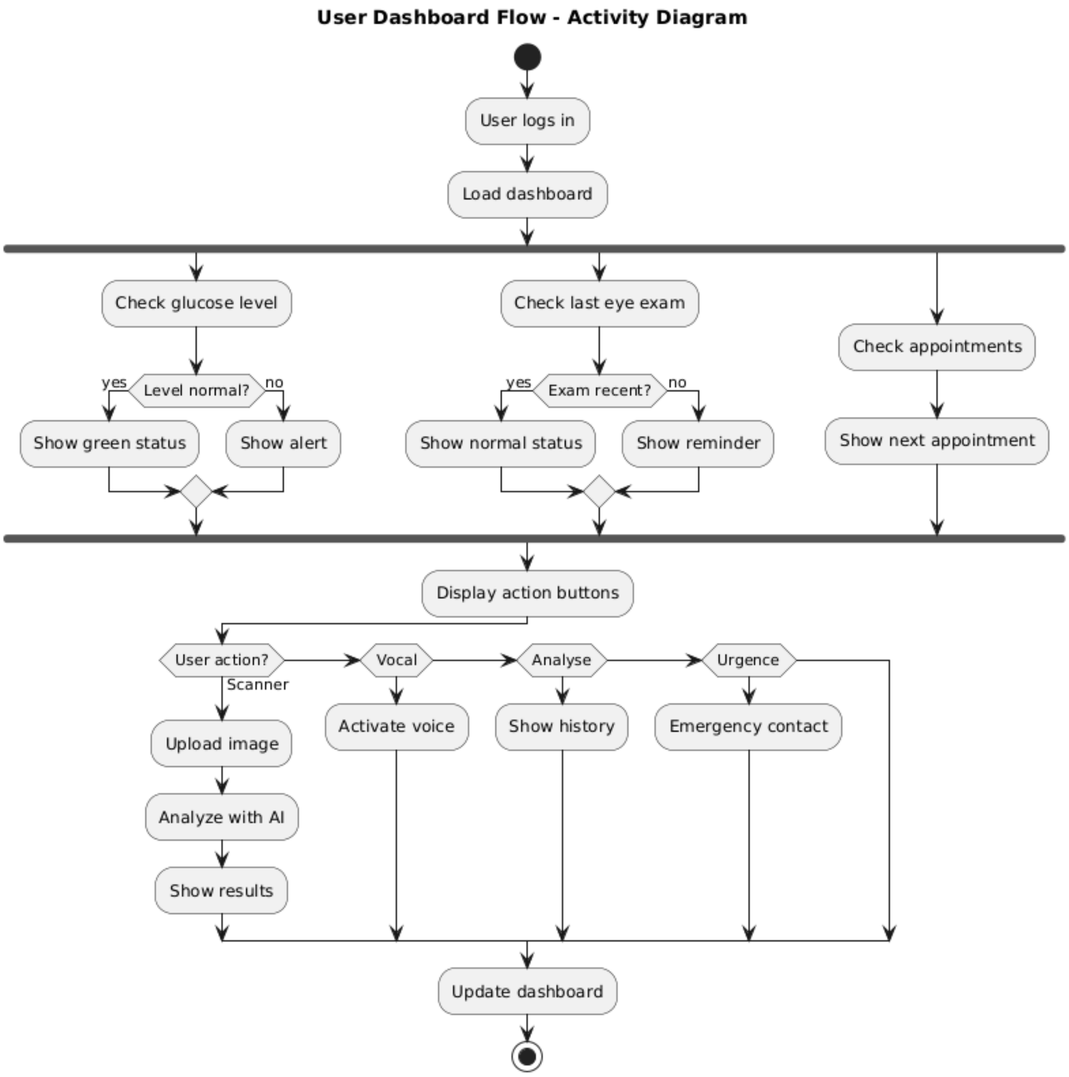

Figure 4.4
Activity Diagram - User Dashboard Flow

Component Diagram

The component diagram outlines the technical architecture of the DiabeteCare system, detailing how various software components are organized and interact to provide the platform's functionality. The system is divided into four main packages: the Frontend Package, which includes React-based user interface components such as the Dashboard for health overviews, the Upload Component for image submissions, and the Chart

Component for glucose trend visualization; the Backend Services Package, responsible for core business logic with components like the Authentication API, User Service for profile management, Analysis Service for AI processing, and Data Service for managing health data; the AI Processing Package, which handles the artificial intelligence tasks, including the Image Processor for retinal image preparation, the CNN Model for diagnostic analysis, and the Results Generator for generating formatted reports; and the Data Storage Package, which includes the User Database for account management, the Medical Database for health records and results, and Image Storage for retinal images and metadata. The diagram demonstrates how these components interact through well-defined interfaces, ensuring system maintainability, scalability, and the ability to scale high-demand components like AI processing independently, while also supporting easy maintenance and updates.

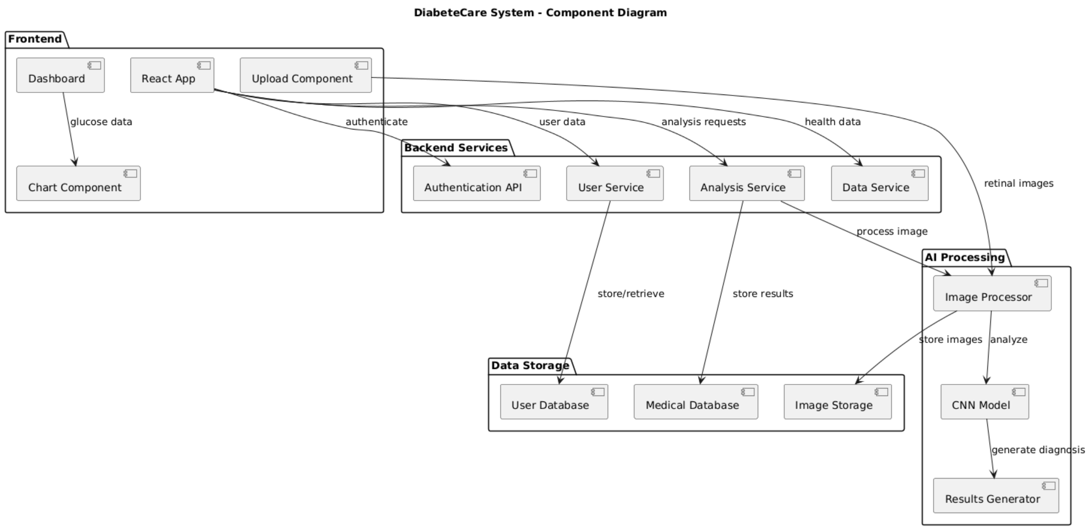

FIGURE 4.5
Component Diagram

4.3 Work Environment

4.3.1 Backend

Python

Python is the primary programming language used for the backend of the DiabeteCare platform. Its simplicity and flexibility make it an ideal choice for handling complex tasks such as data processing, AI model implementation, and system logic. Python's rich ecosystem of libraries like TensorFlow, PyTorch, and scikit-learn enables easy integration of machine learning algorithms, especially those used for diabetic retinopathy detection. It is also known for its robust community support, which helps speed up development and problem-solving during the building process.

FIGURE 4.6
Python libraries Logos

MYSQL

For database management, MySQL is used to store and manage user information, health data, and analysis results. MySQL is a relational database management system (RDBMS) that ensures data is stored in structured tables and can be easily queried. Its reliability and scalability make it suitable for managing the large volumes of medical data generated by DiabeteCare, allowing quick retrieval and processing of essential information like glucose readings, retinal images, and medical history. Additionally, MySQL's support for complex queries ensures that the platform can handle diverse data analytics and reporting tasks.

FIGURE 4.7
MySQL Logo

4.3.2 FRONTEND

VITE

Vite is a fast and modern build tool used in DiabeteCare for frontend development. It improves the development workflow by providing lightning-fast hot module replacement (HMR), allowing developers to see changes in real-time during development.

Vite's optimized build process minimizes build times, leading to a more efficient development cycle. This tool ensures that DiabeteCare's user interface remains responsive and fast, providing a smooth user experience without sacrificing performance. Vite's simplicity and speed are critical for managing the demands of a complex, interactive healthcare platform.

FIGURE 4.8
Vite Logo

TypeScript

TypeScript is a superset of JavaScript that introduces static typing to the language, improving code quality and preventing errors during development. By incorporating TypeScript into the DiabeteCare platform, developers gain the benefit of enhanced code maintainability and easier debugging. With its built-in type-checking, TypeScript reduces runtime errors that could disrupt the user experience, making the platform more robust. Its compatibility with JavaScript libraries, especially React, helps ensure that the platform's frontend remains scalable, maintainable, and error-free as the application grows.

FIGURE 4.9
TypeScript Logo

React

React is the core JavaScript library used for building the user interface of DiabeteCare. It follows a component-based architecture, which allows developers to create reusable, independent components for various sections of the platform, such as health dashboards, image upload sections, and result displays. React's virtual DOM ensures efficient rendering, minimizing the need for frequent re-renders and improving the overall performance of the platform. With its popularity and wide adoption, React helps ensure that the DiabeteCare interface is modern, user-friendly, and easy to scale as new features are added.

FIGURE 4.10
React Logo

4.3.3 API Integration

OpenAI

OpenAI provides advanced natural language processing models that are integrated into DiabeteCare to support various AI-driven functionalities. By leveraging OpenAI's language models, the platform can process and generate text-based insights, making the platform capable of interpreting medical reports, generating diagnostic recommendations, and providing real-time health advice. OpenAI's models are continuously trained on vast datasets, ensuring that the platform can understand complex medical terminology and user queries, improving the platform's overall diagnostic accuracy.

FIGURE 4.11
OpenAI Logo

BetterDoctor

BetterDoctor is an API used by DiabeteCare to access reliable medical data. This includes information about medical conditions, treatments, doctors, and healthcare providers. The BetterDoctor API enables the platform to retrieve updated information regarding medical guidelines and standards of care, ensuring that patients receive the most accurate and relevant information about their diabetic condition. This integration helps enhance the platform's ability to provide informed recommendations and facilitates easy access to expert care.

FIGURE 4.12
BetterDoctor Logo

Hugging Face

Hugging Face is a platform for machine learning models, specifically in the realm of natural language processing (NLP) and computer vision. DiabeteCare uses Hugging Face's pre-trained AI models to improve its diagnostic capabilities, particularly for analyzing retinal images and generating AI-powered results. Hugging Face's vast library of models allows the platform to implement cutting-edge machine learning techniques for detecting diabetic retinopathy and other complications, ensuring that the platform remains at the forefront of AI-driven medical technology. The integration with Hugging Face enhances the platform's ability to process and interpret medical data with high accuracy.

FIGURE 4.13
Hugging Face Logo

4.4 Presentation of DiabeteCare

4.4.1 Authentification

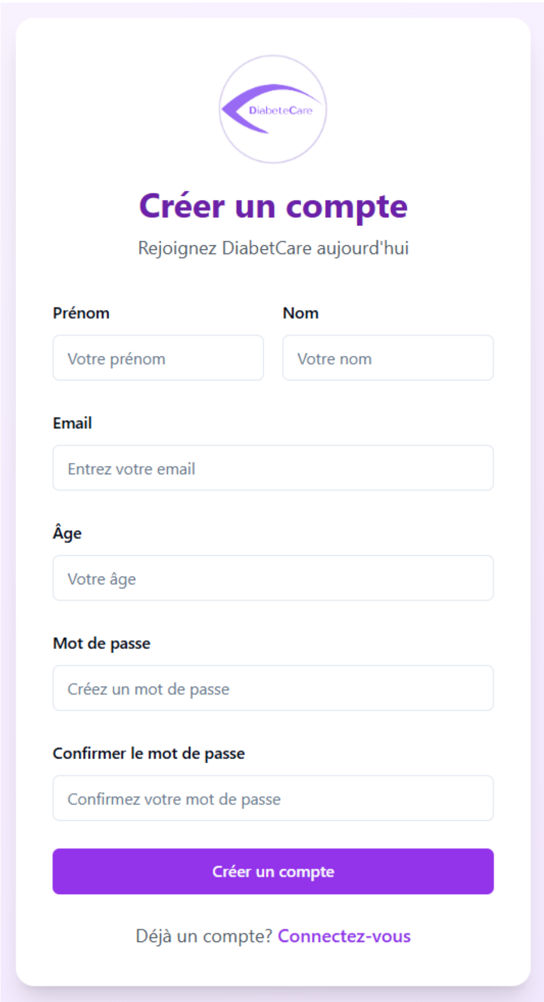

Figure 4.14
Sign up

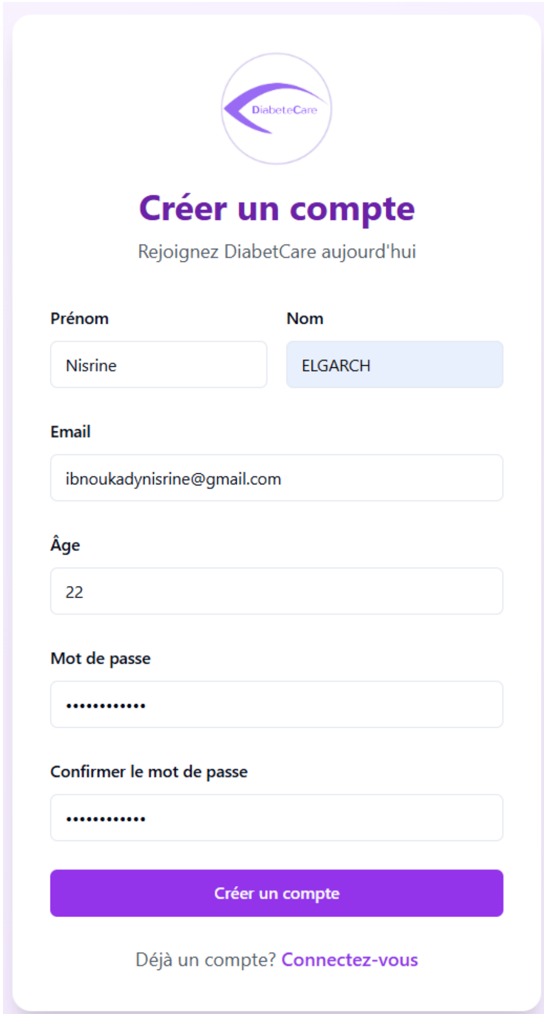

Figure 4.15
Sign up - YoussNiss

The DiabeteCare signup process is simple and user-friendly for diabetic patients. New users complete a basic registration form with five essential fields: first name, last name, email, age, and password confirmation. The form uses clear French labels and maintains the platform's purple branding for visual consistency. After clicking "Create Account," the system automatically creates a personalized dashboard where users can immediately begin monitoring their health, uploading retinal images for AI analysis, and tracking glucose levels. Existing users can access the login page through the "Already have an account? Connect" link. The entire registration takes less than two minutes, allowing patients to quickly start using the platform's AI-powered diagnostic and health management tools without technical barriers.

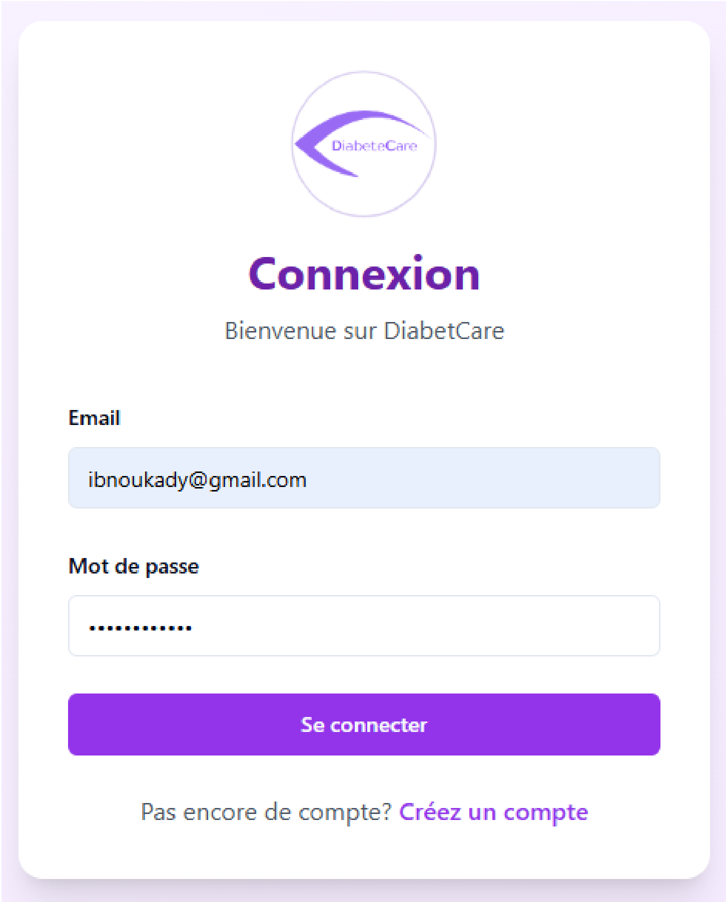

FIGURE 4.16
Sign in

The login interface provides returning users with quick access to their DiabeteCare accounts through a streamlined authentication process. The welcome message "Bienvenue sur DiabeteCare" (Welcome to DiabeteCare) creates a friendly atmosphere while maintaining medical professionalism. The interface shows the pre-filled email address from the previous registration, demonstrating user convenience features. The simple two-field design (email and password) with the prominent "Se connecter" (Connect) button ensures efficient access to personal health management tools, while the "Create account" link accommodates new users who need to register.

4.4.2 Dashboard

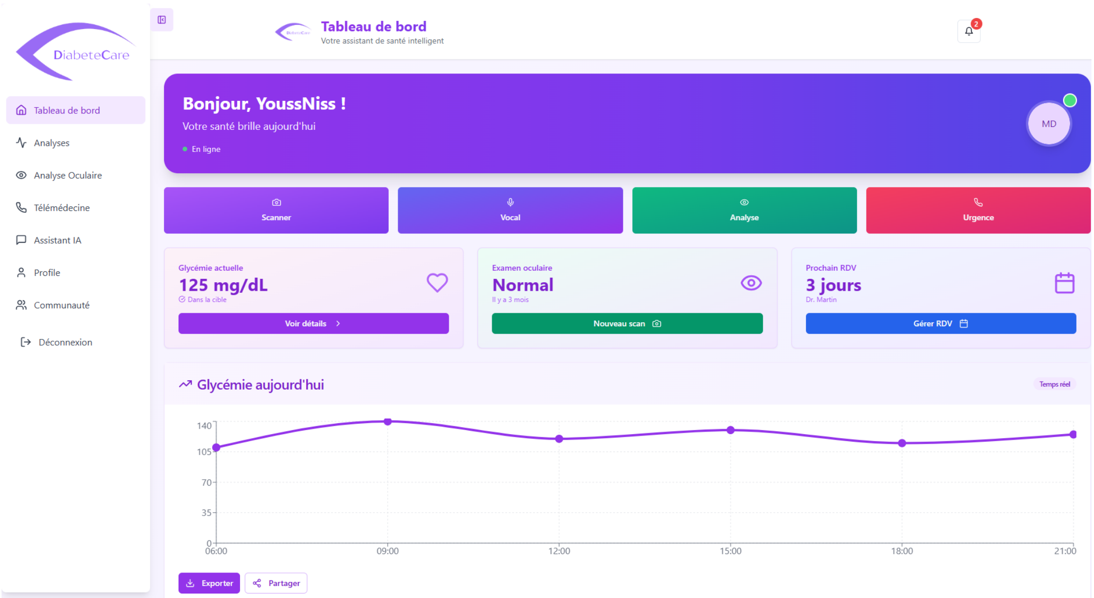

FIGURE 4.17
Dashboard

The main dashboard serves as the central hub for diabetic health management, displaying all critical information in an organized, user-friendly layout. The personalized greeting "Bonjour, YoussNiss!" creates a welcoming atmosphere while the health status message "Your health shines today" provides positive reinforcement. The four quick action buttons (Scanner, Vocal, Analyse, Urgence) offer immediate access to core functions, while the three health monitoring widgets show current glucose levels (125 mg/dL within target), eye examination status (Normal, 3 months ago), and upcoming appointments (3 days, Dr. Martin). The real-time glucose chart at the bottom tracks daily trends from 06:00 to 21:00, enabling patients to monitor their health patterns effectively.

4.4.3 Notifications

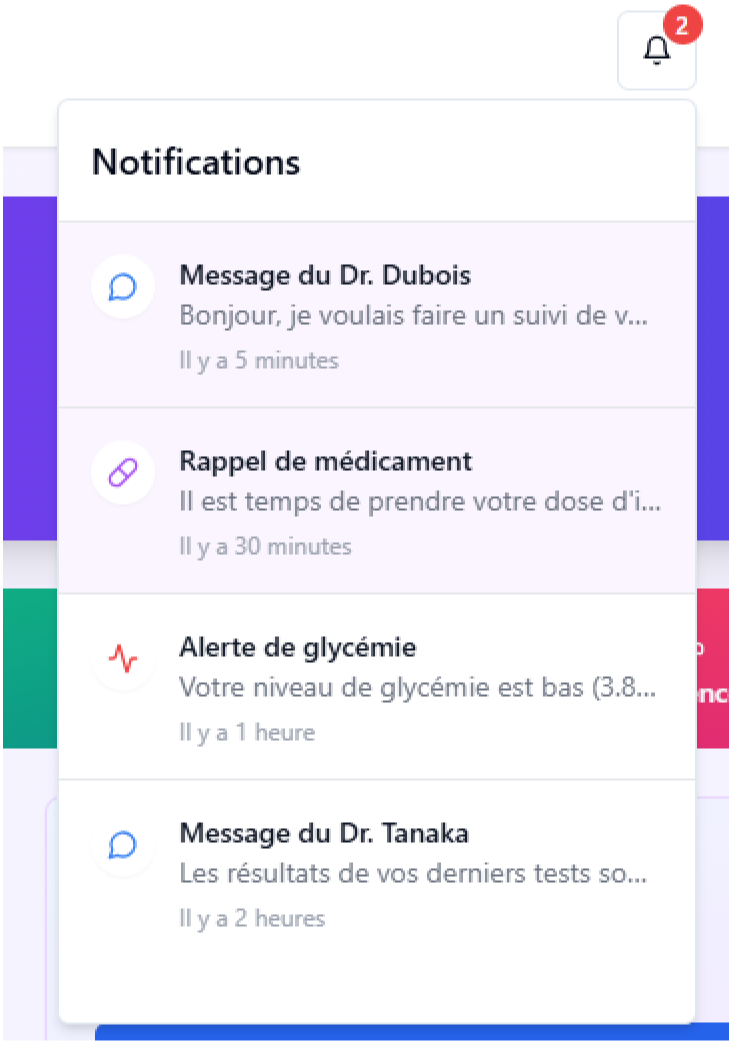

FIGURE 4.18
Notifications

The notification panel demonstrates DiabeteCare's comprehensive communication system that keeps patients informed about their health status and medical care. The notifications include messages from healthcare providers (Dr. Dubois requesting follow-up), medication reminders for timely dosing, glucose level alerts when readings are low (3.8), and test results from medical professionals (Dr. Tanaka). Each notification includes timestamps showing when the alert was sent, enabling patients to stay current with their diabetic care management. This system ensures that critical health information and medical communications are never missed, supporting better patient outcomes through timely interventions and reminders.

4.4.4 Health analyses

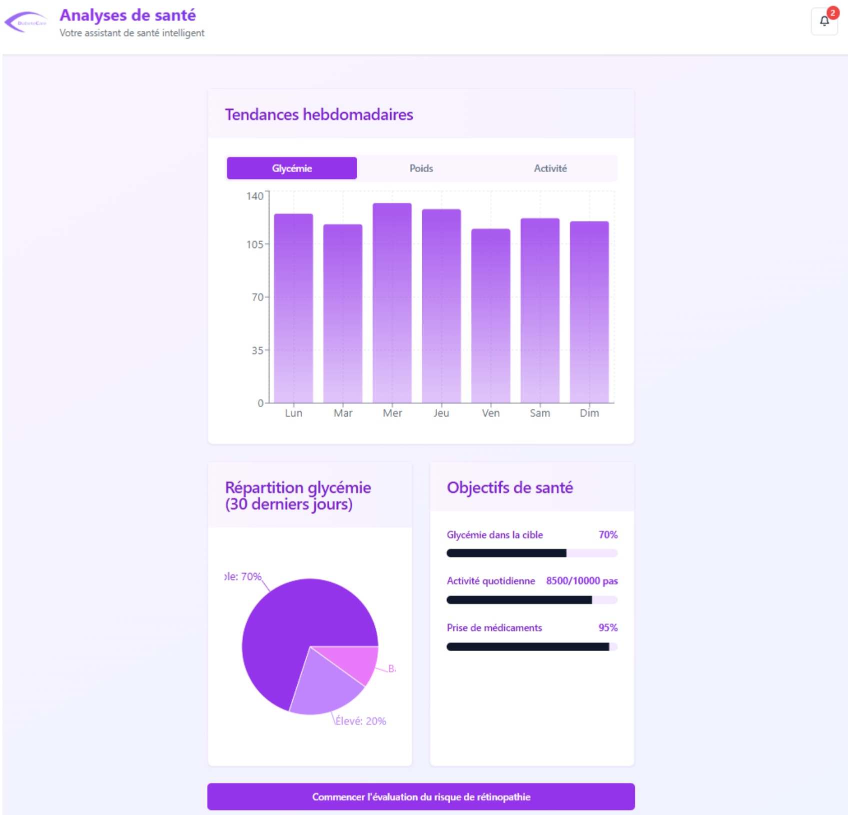

FIGURE 4.19
Health analyses

The health analytics page provides comprehensive visualization of diabetic health trends and progress tracking. The weekly glucose trends chart displays daily readings from Monday to Sunday, showing variations in blood sugar levels with most readings ranging between 105-140 mg/dL. The glucose distribution pie chart for the last 30 days shows that 70% of readings were within target range, while 20% were elevated and 10% were low. The health objectives section tracks key metrics including glucose control (70% achievement), daily activity (8500/10000 steps), and medication adherence (95% compliance). A prominent button at the bottom allows users to "Start retinopathy risk assessment," seamlessly integrating AI-powered eye health screening with general diabetic monitoring.

4.4.5 AI-based eye analysis

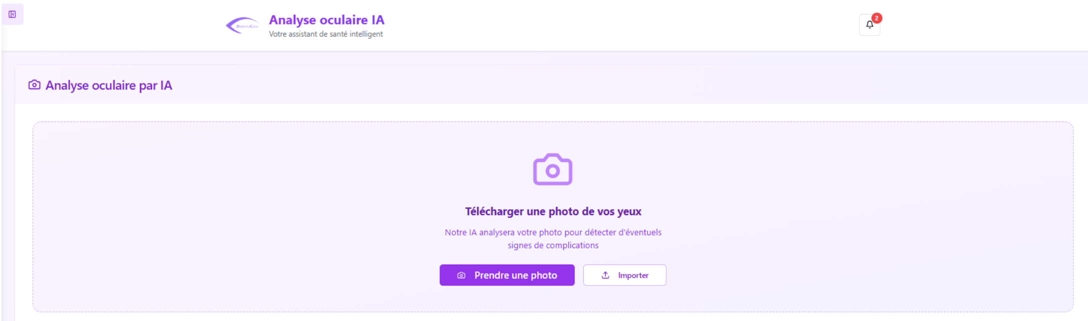

FIGURE 4.20
AI-based eye analysis

The AI eye analysis page represents the core diagnostic functionality of DiabeteCare, providing patients with easy access to automated retinal screening. The clean, simple interface features a large camera icon with clear instructions to "Upload a photo of your eyes" and explanatory text stating that "Our AI will analyze your photo to detect potential signs of complications." Two action buttons offer flexibility: "Take a photo" for real-time image capture using device cameras, and "Import" for uploading existing retinal photographs. This streamlined design ensures that patients can quickly initiate AI-powered diabetic retinopathy screening without technical barriers, making advanced diagnostic technology accessible to all users regardless of their technical expertise.

4.4.6 Télémedecine

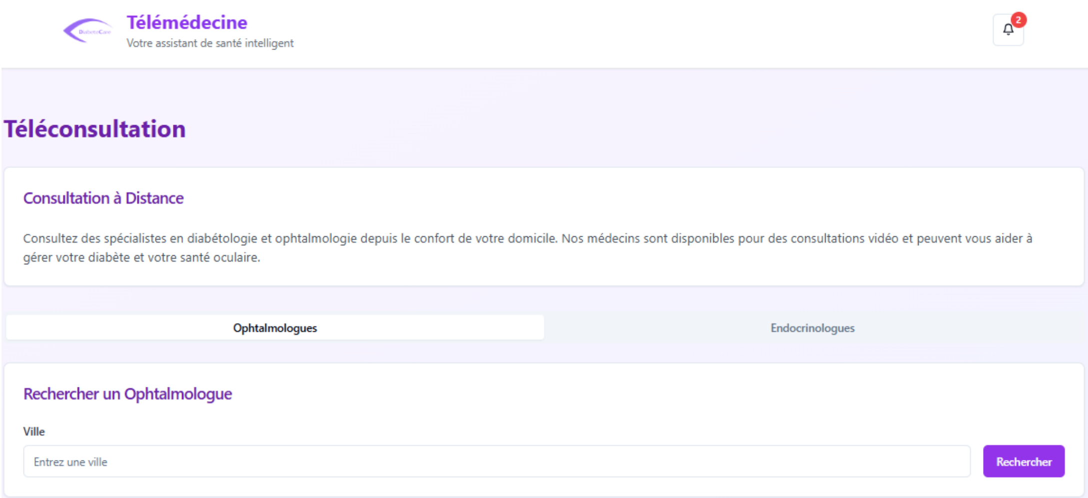

FIGURE 4.21
Télémedecine

The telemedicine section expands DiabeteCare beyond self-monitoring to include professional medical consultations from home. The interface describes remote consultation services with diabetes and ophthalmology specialists, emphasizing convenience and accessibility. The page features two specialist categories - Ophthalmologists and Endocrinologists - providing comprehensive care for diabetic patients' primary and eye health needs. A search function allows users to find specialists by city, with a purple "Search" button maintaining design consistency. This telemedicine integration transforms DiabeteCare from a monitoring tool into a complete healthcare ecosystem, enabling patients to receive professional medical guidance while leveraging the platform's AI diagnostic capabilities for preliminary screening and health tracking.

4.4.7 AI Assistant

Text Chat

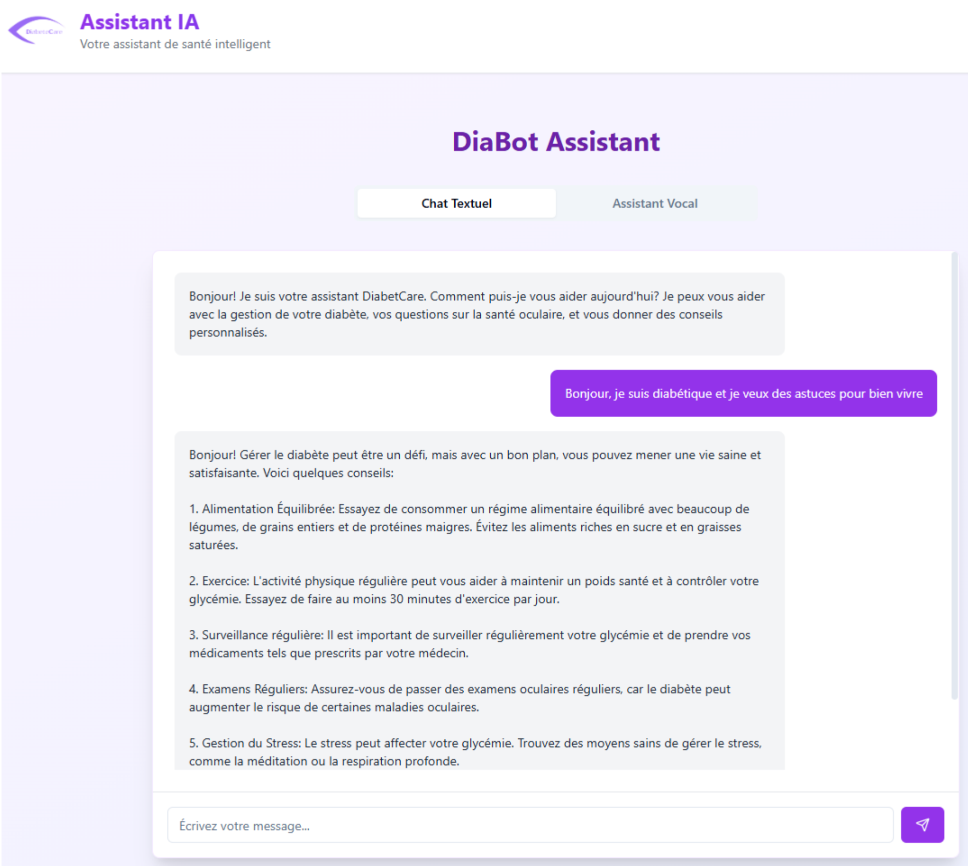

FIGURE 4.22
Text Chat AI

The DiaBot Assistant showcases DiabeteCare's AI-powered conversational support system that provides personalized diabetic health guidance. The chat interface demonstrates an active conversation where the AI assistant introduces itself and offers help with diabetes management, eye health questions, and personalized advice. The assistant responds to a user's request for living well with diabetes by providing five comprehensive recommendations: balanced nutrition with vegetables and lean proteins, regular physical exercise (30 minutes daily), consistent glucose monitoring and medication adherence, regular eye examinations to prevent complications, and stress management through meditation or deep breathing. The interface includes both text chat and voice assistant options, with a message input field for users to ask questions and receive immediate, expert-level guidance on diabetic care management.

Voice Assistant

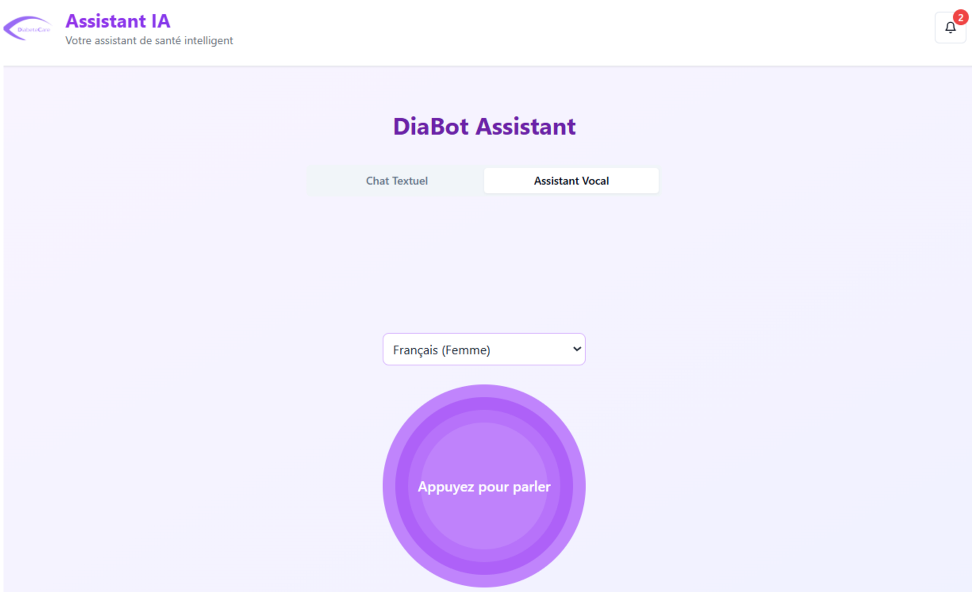

FIGURE 4.23
Voice Assistant AI

The voice assistant interface presents DiabeteCare's accessibility-focused communication option, designed to provide hands-free interaction for users with mobility limitations or visual impairments. The interface features a large, prominent purple circular button labeled "Appuyez pour parler" (Press to speak) that users can activate to initiate voice commands. The language selection dropdown shows "Français (Femme)" indicating customizable voice options with different languages and gender preferences. This voice-enabled functionality allows patients to access health information, upload voice descriptions of symptoms, request medication reminders, or initiate emergency contacts without needing to navigate complex menus, making the platform truly accessible to all diabetic patients regardless of their technical abilities or physical limitations.

4.4.8 Profile

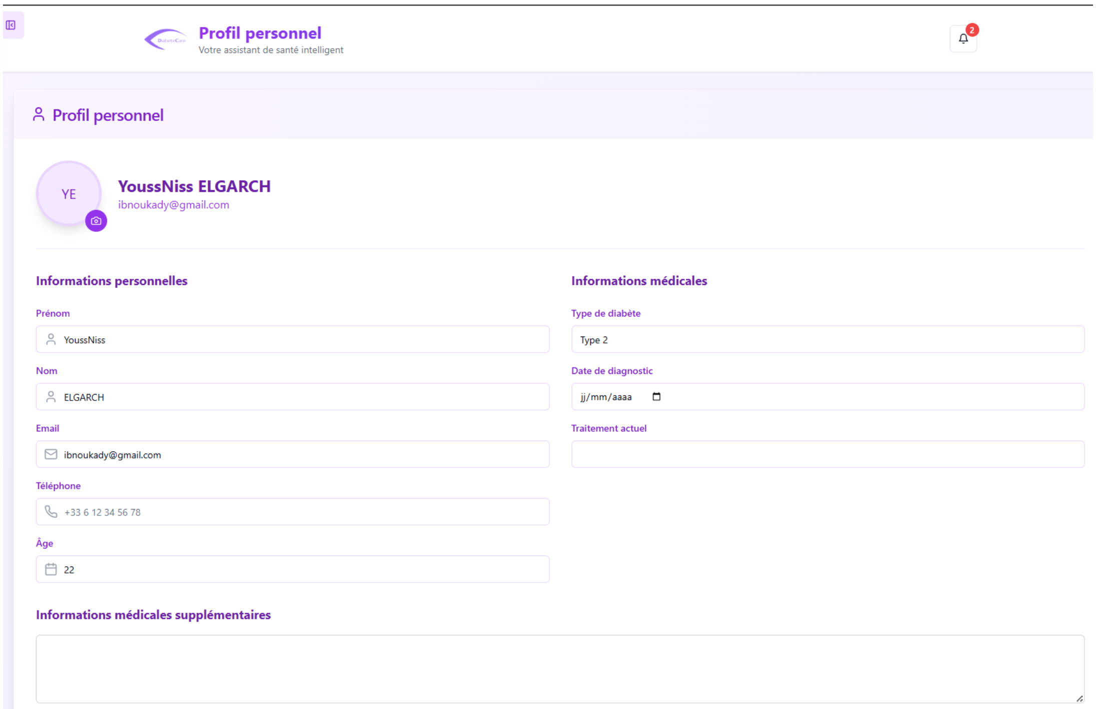

FIGURE 4.24
Profile

The personal profile page provides comprehensive user account management within DiabeteCare, displaying both personal and medical information in an organized, editable format. The profile shows user "YoussNiss ELGARCH" with a personalized avatar displaying initials "YE" and contact information including email (ibnoukady@gmail.com) and phone number. The interface is divided into three main sections: personal information (first name, last name, email, phone, age), medical information (diabetes type showing "Type 2", diagnosis date field, current treatment), and additional medical information for comprehensive health tracking. This profile structure enables healthcare providers to access essential patient information quickly while allowing patients to maintain accurate, up-to-date medical records that inform AI analysis and personalized health recommendations throughout the platform.

4.4.9 Community

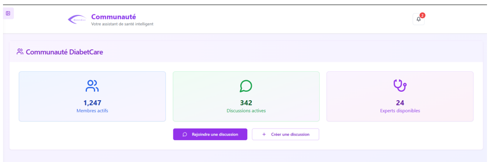

Figure 4.25
Community

The community section transforms DiabeteCare into a comprehensive support network, fostering peer connections and expert guidance for diabetic patients. The interface displays impressive community statistics with 1,247 active members, 342 ongoing discussions, and 24 available medical experts, demonstrating the platform's robust user engagement. Two action buttons enable users to either "Join a discussion" to participate in existing conversations or "Create a discussion" to start new topics. This community feature addresses the social and emotional aspects of diabetes management by connecting patients with similar experiences, providing peer support, sharing practical tips for daily diabetes management, and offering access to expert medical advice. The community integration makes DiabeteCare more than just a diagnostic tool—it becomes a complete ecosystem supporting both the medical and psychological aspects of living with diabetes.

4.5 Conclusion

The DiabeteCare user interface successfully demonstrates how sophisticated medical technology can be made accessible through thoughtful design and user-centered development approaches. The platform effectively bridges the gap between advanced AI diagnostic capabilities and practical, everyday health management needs for diabetic patients. The interface achieves its primary objectives of simplicity, accessibility, and medical professionalism while providing comprehensive functionality for both patients and healthcare providers. The design decisions evident in the screenshots reflect deep understanding of user needs, medical requirements, and technological capabilities. Key achievements include intuitive navigation that requires minimal learning time, seamless integration of AI analysis capabilities into familiar health management workflows, comprehensive accessibility features that serve users with varying abilities, responsive design that functions effectively across all device types, and professional medical standards maintained throughout the user experience. The platform represents a successful implementation of user-centered design principles in medical applications. By focusing on real user needs and maintaining high standards of usability and accessibility, DiabeteCare provides an effective tool for improving diabetic care outcomes through technology. The interface foundation supports future enhancements and scaling as the platform continues to evolve, while the current implementation already demonstrates significant value for diabetic health management and AI-powered medical diagnosis.

General Conclusion

General conclusion

The DiabeteCare project demonstrates how artificial intelligence can be integrated into healthcare to tackle the global diabetes epidemic. By developing an automated diabetic retinopathy detection system alongside a comprehensive patient management platform, we bridge the gap between AI technology and healthcare needs.

Our multi-model approach, comparing established architectures (ResNet, DenseNet, EfficientNet, Inception) with our custom YoussNiss model, revealed that purpose-built architectures can offer efficient, clinically relevant performance. The YoussNiss model achieved 92.6% accuracy, while Inception-v3 reached 97.8%, showing potential for specialist-level diagnostics.

The preprocessing pipeline, using 12 image enhancement techniques, optimized AI performance across varying image qualities, ensuring reliable results in real-world clinical settings.

The user interface translates complex diagnostics into an intuitive platform, featuring personalized health tracking, quick actions, and real-time glucose monitoring. The integration of voice commands, emergency contact features, and telemedicine creates a complete care ecosystem.

Although clinical validation was simulated, the focus on sensitivity and specificity reflects real-world medical standards. The project addresses healthcare challenges like specialist shortages, high costs, and the need for early detection, with the potential to reduce screening costs by 84% and prevent thousands of cases of preventable blindness annually.

Perspectives

DiabeteCare's future development will focus on clinical validation, regulatory compliance, and expanding AI models to detect additional conditions like glaucoma. The platform will integrate with electronic health record systems and improve user accessibility through mobile support and personalized features. It will also expand to include complications like nephropathy and cardiovascular risk, while targeting underserved regions. AI will evolve with federated learning and edge computing for offline functionality. In the long term, DiabeteCare aims to shift toward predictive medicine, integrating IoT devices and wearables for comprehensive health monitoring. The platform will explore AI and genomic data for precision medicine, and could become a global health network contributing to research and public health policies.

Personal Appreciation

This project provided valuable experience in applying AI to healthcare challenges, particularly in balancing model accuracy with efficiency. Designing a preprocessing pipeline highlighted the importance of data quality, while working with medical data underscored the need for sensitivity, transparency, and robust validation in healthcare tech. Developing the DiabeteCare user interface reinforced the importance of user-centered design for diverse audiences, balancing simplicity with functionality. Managing the project across multiple disciplines provided insights into project coordination and research methodology. The experience solidified interest in AI and healthcare, with future career goals focused on deepening expertise in both fields and emphasizing the impact of user-centered design.

Bibliography

[1] American diabetes association. standards of medical care in diabetes. https://diabetesjournals.org/care/issue/46/Supplement$_1$.

[2] Deep learning - nature journal. https://www.nature.com/articles/nature14539.

[3] Development and validation of a deep learning algorithm for diabetic retinopathy detection. https://jamanetwork.com/journals/jama/fullarticle/2588763.

[4] Efficientnet: Rethinking model scaling for convolutional neural networks. https://arxiv.org/abs/1905.11946.

[5] Inception-v3: Rethinking the inception architecture. https://arxiv.org/abs/1512.00567.

[6] International diabetes federation. idf diabetes atlas, 10th edition. https://diabetesatlas.org/.

[7] Pytorch: Deep learning framework. https://pytorch.org/.

[8] React: Javascript library for user interfaces. https://react.dev/.

[9] Resnet: Deep residual learning for image recognition. https://arxiv.org/abs/1512.03385.

[10] Tensorflow: Machine learning platform. https://www.tensorflow.org/.