AUTOMATED AI-AUGMENTED APPROACH TO PSORIASIS LESION DETECTION AND ACCURATE SEVERITY EVALUATION

24_25J_210

Final Report

Perera W.A.S.K – IT21261732

BSc (Hons) in Information Technology Specializing in Data Science

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

April 2025

AUTOMATED AI-AUGMENTED APPROACH TO PSORIASIS LESION DETECTION AND ACCURATE SEVERITY EVALUATION

Perera W.A.S.K

Bachelor of Science (Hons) in Information Technology Specializing in Data Science

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

April 2025

DECLARATION

I declare that this is my own work and this dissertation1 does not incorporate without acknowledgment any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:  Date: 4 / 10 / 2025

The above candidate has carried out research for the bachelor's degree Dissertation under my supervision.

Signature of the supervisor:  Date: 4 / 11 / 2025

(Mr. Samadhi Rathnayake)

Signature of the supervisor: _____ Date: _____

(Mr. Anjana Junius)

ABSTRACT

Psoriasis is a chronic, immune-mediated skin condition characterized by the rapid overproduction of skin cells, resulting in thickened, scaly plaques that commonly appear on the elbows, knees, and scalp, often accompanied by inflammation and discomfort. Traditional methods for diagnosing and assessing psoriasis rely heavily on manual clinical evaluations, which are time-consuming, subjective, and prone to inter-observer variability. This project proposes a comprehensive AI-powered diagnostic framework aimed at revolutionizing the identification and monitoring of psoriasis through advanced image analysis techniques. By leveraging the strengths of deep learning and explainable AI (XAI), the system automatically detects and segments psoriasis lesions from user-submitted images with high precision and efficiency. The solution not only enhances diagnostic accuracy but also supports real-time evaluation of lesion severity, enabling continuous tracking of disease progression. In addition to lesion detection, the system provides insights into the affected region's texture, scaling severity, and redness, facilitating data-driven decision-making for clinicians. A user-friendly interface allows healthcare professionals and patients to interact seamlessly with the system, encouraging broader accessibility and early intervention. The modular design enables integration with telemedicine platforms, supporting remote consultations and personalized treatment recommendations. This paper presents the architecture of the developed system, implementation methodologies, and potential real-world applications. The proposed framework significantly enhances the current diagnostic landscape of psoriasis by offering a reliable, scalable, and intelligent tool that aligns with the modern demands of dermatological care.

Key Words: AI Diagnosis, Deep Learning, Explainable AI, Ghost Module, Grad-CAM, Lesion Detection, Mix Convolution, MobileNetV2, Psoriasis, Severity Classification, YOLOv8.

ACKNOWLEDGMENT

I would like to express my heartfelt gratitude to everyone who supported me throughout the successful completion of my final-year research project.

First and foremost, I am deeply thankful to my research supervisor, Mr. Samadhi Rathnayake, for his unwavering guidance, continuous encouragement, and insightful feedback throughout the project. His support played a crucial role in shaping the quality of this work.

I am equally grateful to my co-supervisor, Dr. Junius Anjana, whose valuable expertise and suggestions significantly contributed to enhancing the depth and standard of the research. His involvement made a remarkable difference in the final outcome.

My sincere appreciation also goes to the CDAP panel for their ongoing support, coordination, and constructive input while our final-year project.

I would also like to extend my special thanks to our external supervisors, Dr. Saman Gunasekara and Dr. Kasun Nayanakantha, for sharing their domain knowledge and for their valuable advice, which helped refine the motivation and direction of the project.

A special note of thanks goes to our team leader, for his exceptional leadership, dedication, and ability to keep the team on track throughout this journey. I also want to thank my fellow group members for their collaboration and consistent support.

Lastly, I am sincerely grateful to my family and friends for their constant encouragement, belief in my potential, and emotional support. Their presence helped me stay motivated and focused, allowing me to reach this significant milestone.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	**Long Form**
CNN	Convolutional Neural Network
YOLO	You Only Look Once
PR Curve	Precision-Recall Curve
ROC Curve	Receiver Operating Characteristic Curve
API	Application Programming Interface
UI	User Interface
DB	Database
CVAT	Computer Vision Annotation Tool
ML	Machine Learning
XAI	Explainable Artificial Intelligence
RGB	Red, Green, Blue (Color Model)
JSON	JavaScript Object Notation
URL	Uniform Resource Locator
AI	Artificial Intelligence

1. INTRODUCTION

The growing capabilities of artificial intelligence (AI) and computer vision have paved the way for transformative advancements in the medical field, particularly in the domain of dermatology. Among various dermatological conditions, psoriasis stands out as a chronic, immune-mediated skin disease characterized by rapid skin cell proliferation, leading to the formation of red, thickened, and scaly patches on the skin surface [10]. The severity of psoriasis varies across individuals, making accurate assessment and classification essential for effective treatment and disease management.

Traditional methods of psoriasis evaluation often depend on expert opinion and manual scoring systems such as the Psoriasis Area and Severity Index (PASI), which involve subjective interpretation of symptoms like erythema, thickness, and scaling. These manual assessments are time-consuming, inconsistent, and susceptible to inter- and intra-observer variability, ultimately affecting the reliability of clinical decisions [2]. This creates a strong need for objective, consistent, and efficient approaches to assess disease severity more reliably.

With the emergence of deep learning models and convolutional neural networks (CNNs), researchers have developed AI-powered systems that can automate the analysis of medical images with high precision. This research proposes an automated, AI-driven platform that classifies psoriasis severity into three clinical levels - **mild, moderate, and severe** by analyzing skin images using deep learning techniques. The model is trained on dermatologist-labeled data and optimized to assess disease patterns beyond basic symptoms like scaling, thus ensuring broader and more clinically relevant evaluation [5].

Unlike previous approaches that focused solely on local scale severity assessment [3] or relied heavily on color-based segmentation techniques [1], the proposed method adopts a more comprehensive clinical perspective. By eliminating the dependence on a single

symptom and incorporating real-time processing capabilities, the system enables quicker, more accurate evaluations and supports timely decision-making in clinical environments.

The scientific contribution of this study lies in its novel use of deep neural networks and image-based analysis to improve diagnostic consistency in psoriasis care. Through extensive testing and evaluation, the system demonstrates the potential to bridge the gap between conventional, subjective assessments and modern, data-driven diagnosis. Ultimately, this research aims to provide an accessible, scalable, and effective tool that supports both clinicians and patients in managing psoriasis more efficiently and accurately.

1.1 Background Literature

Traditional clinical assessments of psoriasis have long relied on dermatologists' subjective evaluation using visual inspection and patient-reported symptoms. These assessments, such as the Psoriasis Area and Severity Index (PASI), often suffer from observer variability and inconsistent scoring due to differing interpretations across clinicians and clinical settings [9]. Psoriasis is a chronic autoimmune condition with various lesion morphologies (plaque, guttate, pustular, erythrodermic), which further complicates standardized evaluation [15].

To address these limitations, researchers have turned to computer vision and machine learning to develop objective, consistent, and scalable methods for psoriasis detection and severity classification. Arunkumar and Jayanna [1] presented one of the earliest methods for automated detection using RGB color histogram analysis. Their algorithm differentiated affected from healthy skin by analyzing color features (redness, greenness, and blueness), significantly reducing the dependency on subjective assessments. However, their approach lacked robustness as it did not incorporate deeper feature representations through advanced deep learning models.

George et al. [2] advanced the field by introducing a scale severity scoring framework using local descriptors and a bag-of-visual-words (BoVWs) model. Their method utilized superpixels and multiple color and texture descriptors, combined with classifiers like SVM and Random Forest, achieving a scoring accuracy of 80.81%. Despite its precision, the approach was limited to scaling severity and did not address other critical PASI parameters such as erythema and induration.

Vincent and Jayasingh [3] conducted a comprehensive review comparing various image processing techniques used for psoriasis detection, including preprocessing, segmentation, and feature extraction. Their review emphasized the need for integrated deep learning-based systems capable of combining multiple models for improved accuracy, highlighting the fragmented nature of current approaches.

Syu et al. [4] proposed a deep neural network (DNN)-based model for psoriasis detection. The architecture incorporated multiple convolutional layers for multi-level feature extraction, significantly outperforming traditional diagnostic techniques. However, their model focused solely on lesion detection and did not incorporate structured severity classification, exposing a gap in comprehensive psoriasis management systems.

Roslan et al. [5] explored convolutional neural networks (CNNs) for classifying different psoriasis types. Their model showed promising accuracy (up to 82.9% for plaque psoriasis), confirming CNNs' potential in clinical applications. Nevertheless, limitations regarding dataset diversity and real-time deployment remain critical research challenges.

The introduction of MixConv, a mixed depthwise convolution technique by Tan and Le [6], brought a major leap in efficiency and feature representation. By combining multiple kernel sizes within a single layer, MixConv allowed models like MobileNetV2

to capture both fine-grained local and contextual global features. As Tsang explains [11], this multi-scale capability is especially beneficial in dermatological contexts where lesion appearance varies significantly in size and texture.

Explainability remains a crucial aspect of AI in healthcare. Grad-CAM, developed by Selvaraju et al. [8], enables class-discriminative localization in CNNs by highlighting the most relevant regions in an image used for decision-making. This visualization helps in interpreting model predictions and building trust among clinicians. Saranya and Subhashini [7] further reviewed various Explainable AI models and their applications in clinical domains, emphasizing the need for interpretable solutions in dermatology.

Datasets also play a pivotal role in model generalization. Publicly available datasets such as those by Kumar [12], Goel [13], and Roboflow [14] provide a foundation for model training and benchmarking. However, many of these datasets lack diversity in skin tones and lesion types, limiting the generalizability of trained models in real-world clinical scenarios.

WebMD [15] and DermNet NZ [10] offer authoritative clinical overviews of psoriasis, describing its different forms and treatment challenges. These resources underscore the complex presentation of the disease and highlight the need for robust, explainable, and scalable AI solutions to aid clinicians in diagnosis and treatment planning.

In summary, while significant advancements have been made in automating psoriasis detection and severity scoring using machine learning and deep learning, existing models often focus on isolated aspects such as lesion detection or scaling severity. There is a pressing need for integrated, explainable systems that can handle diverse clinical scenarios and offer consistent, real-time evaluations. This research aims to address these gaps by developing an AI-based severity classification system that leverages deep

learning, explainable AI, and multi-scale feature extraction for clinically interpretable and scalable applications.

1.2 Research Gap

Traditional methods for psoriasis detection and severity classification have heavily relied on manual clinical assessments using visual inspection and subjective judgment by dermatologists. These methods are often inconsistent due to inter-observer variability and are time-consuming, limiting their reliability in large-scale or routine screenings. While prior AI-based approaches have introduced image analysis models using handcrafted features and deep learning architectures like CNNs, they often face challenges in generalizing across diverse skin tones, lighting conditions, and lesion morphologies.

For example, early work like that of Arunkumar and Jayanna [1] focused primarily on RGB color-based differentiation without addressing structural lesion features or generalizability. George et al. [2] introduced local descriptors for severity assessment, yet their system was primarily focused on scaling severity and lacked full integration into real-time diagnostic workflows. Deep neural models, such as those by Syu et al. [4] and Roslan et al. [5], provided promising classification performance, but did not incorporate explainability or modular enhancements for efficiency.

Furthermore, although CNNs are widely adopted, their traditional convolutional layers can be computationally intensive. Recent innovations like MixConv [6], which combines multiple kernel sizes within a layer for multi-scale feature capture, and Ghost Modules, which reduce redundancy and model size while preserving performance, have yet to be effectively applied in the context of psoriasis severity assessment. Additionally, most existing models focus either on detection or classification but rarely on a scalable, interpretable system capable of end-to-end severity evaluation in clinical contexts.

This study addresses these limitations by integrating:

- Mix Convolution Blocks for capturing multi-scale lesion features across various severity levels,
- Ghost Modules to enhance model efficiency and reduce computational complexity,
- and Explainable AI techniques like Grad-CAM to visualize and interpret model predictions, enhancing trust and usability.

The proposed system is designed to work robustly across diverse skin types and lesion types, ensuring clinical relevance, real-time assessment, and support for informed medical decision-making.

Table 1: Research Gap

Features	[1]	[2]	[3]	[4]	[5]	Proposed System
RGB-based Psoriasis Detection	✓	✗	✓	✗	✗	✓
Deep Learning for Lesion Classification	✗	✓	✓	✓	✓	✓
Robustness Across Diverse Skin Tones & Lesion Types	✗	✗	✗	✗	✓	✓
Real-Time, Clinically Scalable Psoriasis Severity Classification	✗	✗	✗	✗	✗	✓
Explainable AI Integration (e.g., Grad-CAM)	✗	✗	✗	✗	✗	✓

Mix Convolution for Multi-Scale Feature Extraction	✗	✗	✗	✗	✗	✓
Ghost Module for Lightweight and Efficient Architecture	✗	✗	✗	✗	✗	✓

1.3 Research Problem

Psoriasis, a chronic inflammatory skin disorder, affects over 125 million individuals globally and presents a significant diagnostic challenge due to its heterogeneous manifestation across different skin types, tones, and severities. Accurate severity classification of psoriasis is critical for effective clinical decision-making and personalized treatment planning. However, current clinical practices primarily rely on manual visual assessments by dermatologists using scoring systems like PASI, which are inherently subjective, time-consuming, and inconsistent due to inter- and intra-observer variability. The lack of a scalable, objective, and automated diagnostic solution limits the accessibility and accuracy of psoriasis management, especially in resource-constrained settings.

Although advancements in artificial intelligence and deep learning have demonstrated success in automating various dermatological tasks, there remain key challenges that hinder the development of a robust, clinically viable psoriasis severity classification system. Existing AI-based approaches often fall short in their ability to generalize across diverse lesion types and skin tones, leading to performance degradation in real-world applications. Moreover, many of these systems lack transparency, making it difficult for clinicians to trust or interpret model predictions, which in turn hampers adoption in clinical workflows.

The central research problem addressed in this project is the absence of a comprehensive, interpretable, and computationally efficient AI-based system for real-

time psoriasis severity detection and classification. This research targets the following four key challenges:

1. Robust and Scalable Lesion Detection Across Diverse Skin Conditions:

Developing an automated system capable of accurately detecting and classifying psoriasis lesions under varying clinical conditions, including differences in lighting, skin tones, and lesion morphology, is essential. Current methods often fail to perform reliably across these variations, limiting their generalizability.

2. Advanced Feature Representation for Severity Classification:

Traditional CNN architectures may overlook subtle features indicative of severity levels. There is a need to integrate advanced modules such as Mix Convolution Blocks for capturing multi-scale lesion patterns and Ghost Modules for reducing model redundancy while maintaining performance, enabling efficient severity classification with fewer parameters.

3. Real-Time Processing and Clinical Deployment:

Despite high accuracy in controlled settings, many AI models struggle with real-time inference, which is critical in clinical environments. Ensuring the proposed system is lightweight and capable of real-time image analysis will support seamless integration into healthcare settings, improving diagnostic efficiency and scalability.

4. Integration of Explainable AI for Clinical Interpretability:

Most deep learning models operate as “black boxes,” offering limited insight into their decision-making process. Incorporating Explainable AI techniques such as Grad-CAM is necessary to visualize critical features influencing predictions, thereby increasing clinician trust, and facilitating responsible AI deployment in dermatology.

By addressing these challenges, this research aims to develop an innovative, AI-powered psoriasis assessment tool that combines deep learning with interpretable and lightweight

architectures. The system will deliver fast, accurate, and explainable severity classification (mild, moderate, severe), enhancing clinical decision-making and ultimately improving patient outcomes. This solution strives to bridge the gap between technological innovation and practical, real-world applicability in dermatological healthcare.

1.4 Research Objectives

1.4.1 Main Objective

The primary objective of this research is to develop a clinically interpretable, efficient, and fully automated AI-driven system for the detection and severity classification of psoriasis, integrating advanced deep learning modules and explainable AI techniques. This system aims to provide real-time and reliable analysis of psoriasis lesions, enabling dermatologists to make informed, consistent, and timely clinical decisions across diverse patient populations.

The proposed system seeks to overcome the limitations of traditional manual assessment methods and existing automated models that lack generalizability and transparency. By integrating multi-scale convolutional processing, lightweight model optimization, and visual interpretability, the system is designed to capture nuanced variations in lesion morphology and skin tone. It aims to support healthcare professionals with an objective and consistent diagnostic tool, reducing variability in assessment and improving accessibility to high-quality dermatological care. Ultimately, the system aspires to enhance diagnostic accuracy, optimize treatment planning, and improve patient outcomes in both clinical and telemedicine settings.

1.4.2 Specific Objectives

Sub Objective 1: Development of an AI-Powered Lesion Detection and Segmentation Framework

To design and implement an advanced lesion detection and segmentation pipeline using state-of-the-art CNN-based architectures and object detection models such as YOLO.

The system will utilize modern convolutional neural networks backbones (e.g., EfficientNetB0, ResNet50) and object detection models like YOLOv5/YOLOv8 to accurately detect and localize psoriasis lesions from clinical images. These models will be trained on annotated datasets that include a variety of lesion shapes, sizes, and appearances. Emphasis will be placed on robustness across varying skin tones and lighting conditions. The segmented lesion regions produced by this module will serve as input for downstream severity analysis.

Sub Objective 2: Design of a Lightweight Architecture for Efficient Severity Assessment

To incorporate a lightweight and computationally efficient deep learning architecture using Ghost Modules and Mix Convolution Blocks to assess the severity of psoriasis lesions.

Once lesions are segmented, the system will analyze extracted regions to assess severity levels such as mild, moderate, or severe. This component will leverage MixConv for multi-scale feature extraction and Ghost Modules to maintain high performance with reduced computational load. These architectural enhancements ensure efficient and scalable severity classification suitable for real-time clinical settings and mobile health applications.

Sub Objective 3: Severity Classification Based on Visual and Structural Lesion Features

To develop a severity classification model that categorizes psoriasis lesions into mild, moderate, or severe, based on expert-labeled datasets and comprehensive visual features excluding scaling and PASI-based parameters.

This objective focuses on training a deep learning model using image datasets annotated by dermatology experts to reflect real-world clinical judgment of severity. The model will learn to assess severity levels based on a variety of non-scaling visual cues, such as lesion color variations, shape irregularities, lesion size, border definition, and distribution pattern on the skin.

Rather than using predefined scoring systems like PASI or relying on scaling characteristics, the model will derive patterns directly from expert assessments, allowing for a more flexible and realistic classification approach. The system will be fine-tuned to generalize across different lesion types and skin tones, making it robust for practical clinical deployment.

This sub-objective ensures that the severity assessment is clinically aligned, data-driven, and grounded in expert consensus, improving both accuracy and trustworthiness in AI-powered diagnosis.

Sub Objective 4: Integration of Explainable AI Techniques for Clinical Transparency

To integrate Explainable AI (XAI) techniques, such as Grad-CAM, for visualizing model decision regions and enhancing clinical interpretability.

To promote trust in AI-generated outcomes, the system will include Grad-CAM visualizations that highlight the specific regions influencing the model's severity classification. These visual insights will be made accessible through the user interface, allowing dermatologists to validate and interpret predictions. This enhances system transparency, encourages adoption, and aligns with ethical standards in clinical AI deployment.

Sub Objective 5: Development of a Web-Based Interface for Patient and Doctor Interaction

To design and develop a user-friendly web platform that allows both patients and dermatologists to interact with the AI-based psoriasis detection and severity classification system.

This objective aims to create a responsive and accessible web-based system where patients can upload images for preliminary assessment, and dermatologists can review AI-generated severity scores alongside explainable visual outputs. The platform will serve as a communication bridge, providing personalized insights to patients and aiding doctors in making data-driven clinical decisions. Features will include secure login, image upload, result visualization, Grad-CAM-based lesion explanations, and downloadable reports for clinical use.

2. METHODOLOGY

2.1 Methodology

2.1.1 Requirement Gathering

The foundation of the DermaScope AI system was established through a structured and multidisciplinary requirement gathering process, aimed at deeply understanding both the clinical and technical complexities of automated psoriasis detection and severity classification. This process combined insights from dermatology and data science, supported by comprehensive literature analysis, system evaluation, and user-centered design planning.

To ensure the system addresses real-world clinical needs, extensive consultations were held with dermatologists, AI researchers, and healthcare IT experts. Dermatologists provided essential input on lesion types, visual patterns, and severity indicators, while data scientists offered guidance on model selection, image preprocessing techniques, and performance tuning. This collaboration guaranteed that the design is grounded in both clinical relevance and technical feasibility.

A detailed review of existing AI-based dermatological tools revealed current system limitations, especially in terms of scalability, generalization to diverse skin tones, and lack of explainability. Based on these findings, the proposed system emphasizes real-time lesion analysis, efficient severity classification, and visual interpretability to support dermatologists in daily practice. Additionally, ethical, regulatory, and security aspects were considered from the early stages, ensuring compliance with healthcare standards and privacy regulations.

Key stages,

- **Literature Review and Benchmarking**.

  Identification and analysis of recent research papers and existing dermatology-based AI solutions to inform system features, performance expectations, and innovation potential.

- **Feasibility Study**.

  An overall feasibility assessment was carried out to evaluate the availability of psoriasis image datasets, annotation guidelines, and access to clinical expertise. Particular attention was paid to balancing model complexity with the goal of achieving real-time performance.

- **Schedule Feasibility**.

  A comprehensive project timeline was developed, outlining all key phases, deliverables, and checkpoints using a Gantt chart. This schedule includes major milestones such as dataset preparation, model development, system testing, and report generation. Continuous monitoring mechanisms were put in place to ensure timely progress and adaptability to challenges throughout the project lifecycle.

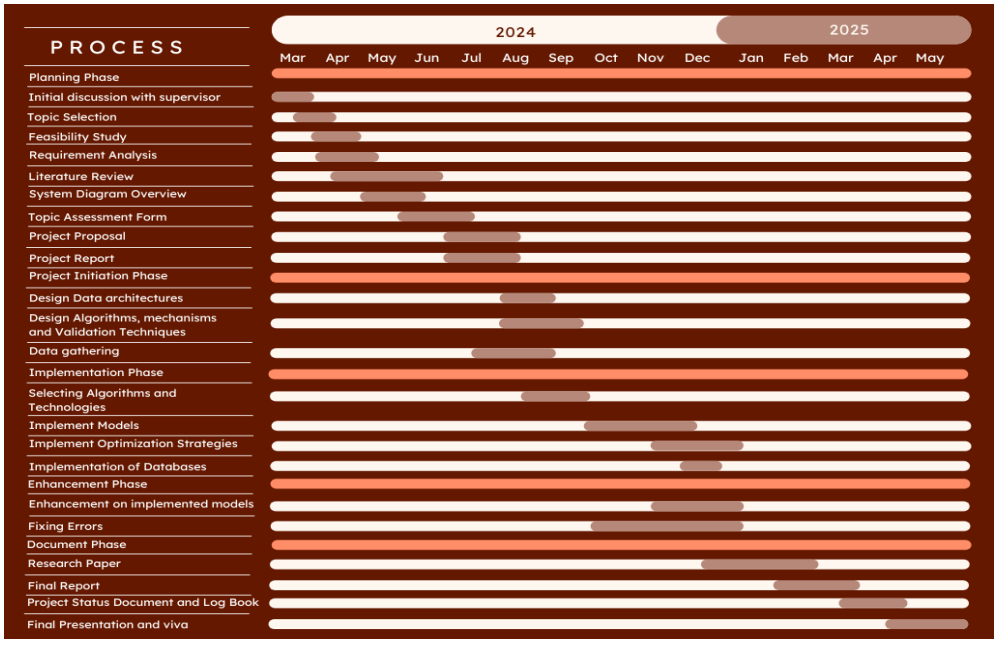


Figure 1: Gannt Chart

- **Expert Consultations**.

  Domain experts, including dermatologists and machine learning professionals, were interviewed to finalize clinical requirements and use-case scenarios. These insights influenced the choice of detection and classification models and the incorporation of explainable visual outputs.

- **Component Selection and Scope Finalization**.

  Key system components such as YOLO based lesion detection, EfficientNetB0/ ResNet50 segmentation models, MixConv and Ghost Modules for severity assessment, and Grad-CAM for explainability were identified and finalized. The system scope was refined in alignment with clinical needs and technical constraints.

- **Prototype Planning and User Flow Mapping**.

  Initial interface designs were drafted to illustrate how patients and dermatologists would interact with the system. This included modules for image upload, lesion visualization, severity scoring, and treatment history tracking—all aiming to enhance usability, transparency, and accessibility.

2.1.2. System Architecture

The system architecture of DermaScope AI illustrates a comprehensive, AI-powered dermatological analysis framework designed to detect and assess the severity of multiple skin conditions. The process begins with the collection of disease-specific data for acne, eczema, and psoriasis. These datasets undergo independent preprocessing and analysis tailored to each condition, ensuring the model can capture condition-specific patterns, lesion types, and variations. The outcomes of these analyses are then fed into a centralized Skin Disease Detection module, which identifies the specific dermatological condition present in an image.

Once the disease is identified, the pipeline progresses to the Severity and Progress Assessment stage, where advanced deep learning models evaluate the condition's

severity and monitor its progression over time. The final insights including disease type, severity level, and progression trends are presented through the DermaScope AI interface, designed for both patients and healthcare professionals. This unified platform provides interpretable results, visualizations, and clinical guidance, facilitating timely diagnosis and personalized treatment planning across a range of skin disorders.

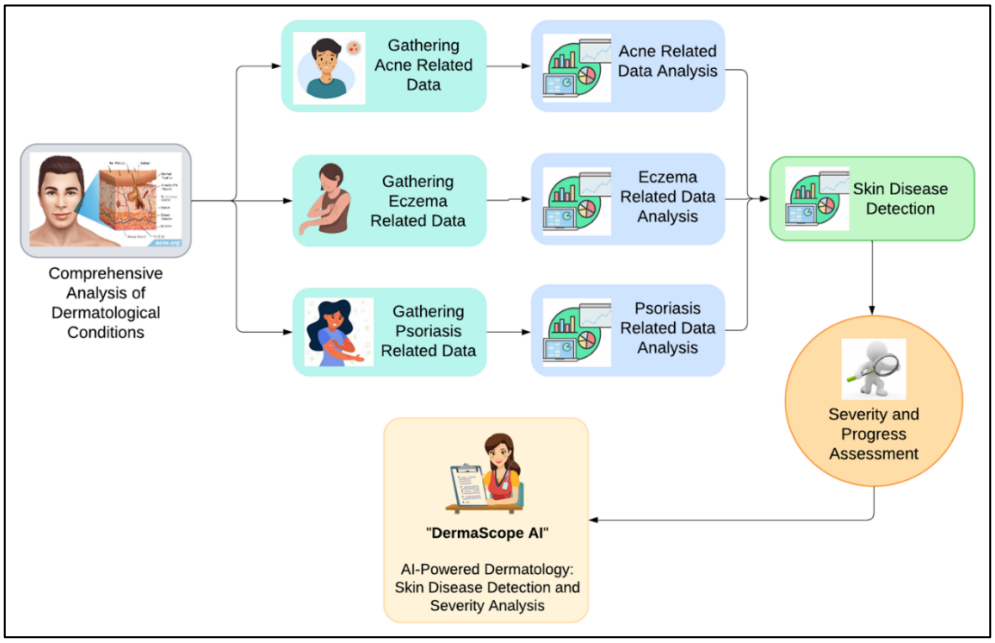


Figure 2:Overall System Architecture

2.1.4. Component Architecture

The DermaScope AI system architecture adopts a modular and cloud-integrated design, enabling seamless and accurate detection, segmentation, and severity assessment of psoriasis lesions. The process begins when the user uploads an image of the affected skin lesion through the DermaScope AI interface. This image is then sent to a web

server, which acts as the intermediary between the user interface and the backend processing modules.

The image data is routed to the web application layer, where the core modules are deployed. These include the Psoriasis Detection and Segmentation Module, responsible for localizing and isolating the lesion using deep learning-based image analysis techniques. Following detection, the Feature Extraction Module identifies key visual attributes from the segmented lesion such as texture, boundary irregularity, and color variations required for severity classification.

Next, the Psoriasis Severity Classification Module, enhanced with Mix Convolution and Ghost Modules, processes the extracted features to classify the lesion into categories such as mild, moderate, or severe. The results, along with additional metadata, are stored in a cloud-based database, ensuring scalability and accessibility for future retrieval and analysis.

The final output is rendered as a daily report and an interactive dashboard for dermatologists, presenting the analyzed data in a user-friendly and interpretable format. This report includes lesion classification results, image visualizations, and progression insights, empowering healthcare providers to make informed clinical decisions. The system’s architecture ensures efficiency, real-time processing, and high reliability, making DermaScope AI a practical solution for AI-assisted dermatological analysis.

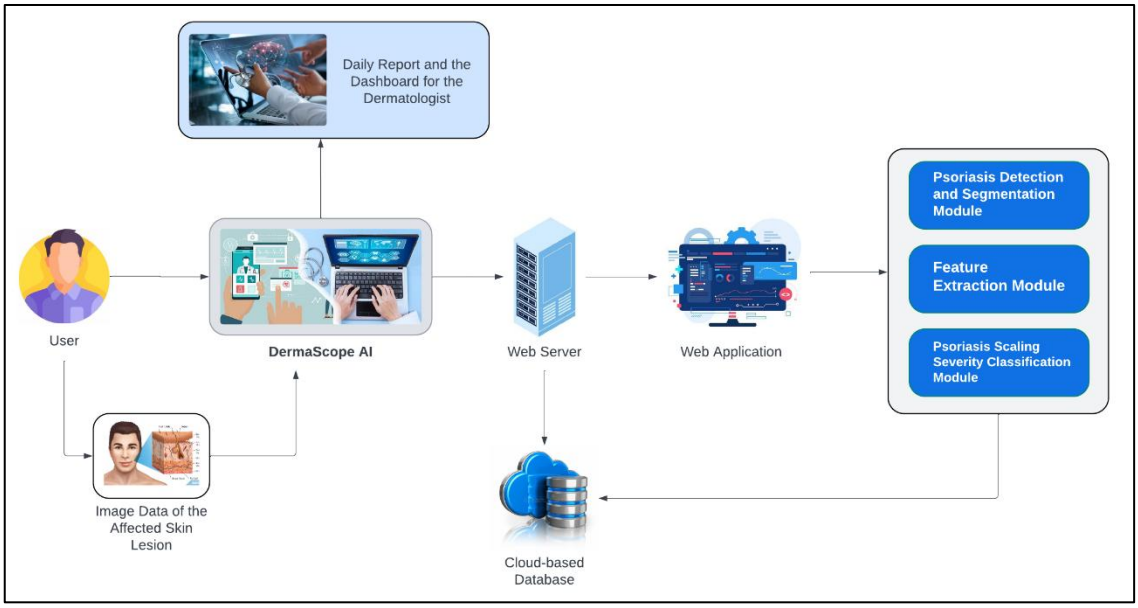


Figure 3: Component Architecture of Psoriasis Detection and Severity Assessment

2.1.2. Dataset Description and Data Preprocessing

Recognizing the lack of a unified, high-quality dataset for automated psoriasis detection and severity classification, a comprehensive dataset was curated by combining multiple trusted dermatological image repositories. Images were sourced from the Psoriasis Skin Dataset on Kaggle [12], the DermNet Dataset on Kaggle [13], and the Psoriasis Dataset from Roboflow Universe [14]. These sources ensured inclusion of a diverse range of skin tones, lesion locations, and severity levels. To enhance clinical accuracy, all images were validated and annotated by certified dermatologists, ensuring the dataset's reliability and alignment with real-world diagnostic standards.

The final dataset includes 270 images of psoriasis-affected skin and 270 images of normal skin, offering a balanced composition for binary classification tasks. For severity classification, the dataset comprises 100 expertly labeled images for each category: Mild, Moderate, and Severe psoriasis. Additionally, a YOLOv8-compatible lesion

detection subset was developed, featuring manually annotated bounding boxes around lesion regions to support precise object detection tasks.

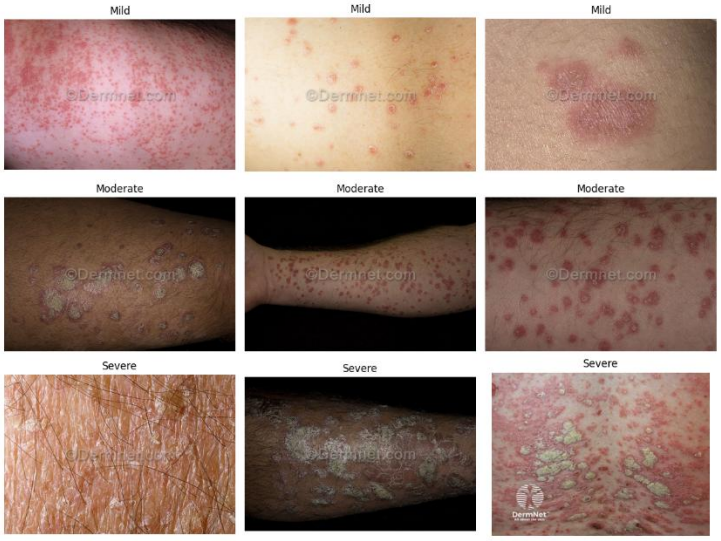


Figure 4: Images from data set

To support robust model training and improve generalization, a series of preprocessing steps were applied. Images were standardized to 640×640 pixels for lesion detection and 224×224 pixels for classification models. All images were converted to RGB format, and pixel values were normalized to the [0,1] range. To reduce overfitting and improve model resilience, various data augmentation techniques were employed, including horizontal and vertical flipping, ±30° rotation, brightness/contrast adjustment, zoom and cropping, and the addition of Gaussian noise. These enhancements helped the system learn meaningful features while maintaining adaptability across diverse clinical scenarios.

```python
# Data augmentation
datagen = ImageDataGenerator(
    rotation_range=30,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode="nearest"
)

train_gen = datagen.flow(X_train, y_train, batch_size=BATCH_SIZE)
```

Figure 5:Data Augmentation

```python
# Function to load data
def load_data(dataset_path):
    images = []
    labels = []
    categories = ['NORMAL_SKIN', 'PSORASIS_SKIN']  # Folder names for classes

    for category in categories:
        category_path = os.path.join(dataset_path, category)

        if not os.path.exists(category_path):
            raise FileNotFoundError(f"Directory not found: {category_path}. Please check the path and ensure the directory exists.")

        label = categories.index(category)  # Label 0 for NORMAL_SKIN, 1 for PSORIASIS_SKIN

        for img_name in os.listdir(category_path):
            img_path = os.path.join(category_path, img_name)
            try:
                img = cv2.imread(img_path)
                if img is None:
                    print(f"Could not read image {img_name}. Skipping.")
                    continue
                img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  # Convert to RGB
                img = cv2.resize(img, IMG_SIZE)  # Resize to target size
                images.append(img)
                labels.append(label)
            except Exception as e:
                print(f"Error loading image {img_name}: {e}")

    return np.array(images), np.array(labels)
```

Figure 6: Data Preprocessing

2.1.6. Psoriasis Detection and Classification

2.1.6.1. Model Building and Training

For the psoriasis lesion detection task, multiple deep learning models were explored and evaluated independently. CNN-based architectures such as EfficientNetB0, ResNet50, and InceptionResNetV2 were fine-tuned using transfer learning to detect and localize psoriasis-affected regions. These models, known for their effective feature extraction capabilities, were chosen due to their prior training on large-scale image datasets like

ImageNet. Transfer learning allowed these models to adapt efficiently to the psoriasis domain with limited training data, reducing both training time and computational requirements.

In addition to CNN-based approaches, the YOLOv8 object detection model was also implemented to achieve real-time lesion localization. To support YOLO training, a dedicated dataset was prepared using manual bounding box annotations created via the CVAT (Computer Vision Annotation Tool). The YOLO model was trained separately from the CNN models, providing a fast and lightweight alternative for object-level lesion detection. These experiments enabled a comparative understanding of each architecture's suitability for psoriasis detection tasks under different conditions.

Approach 1 – CNN

A custom Convolutional Neural Network (CNN) was developed to classify skin images as either psoriasis-affected or normal. The model architecture consisted of three convolutional layers with increasing filter sizes (32, 64, and 128), each followed by max pooling and dropout layers to reduce overfitting and capture hierarchical features. After flattening, a fully connected dense layer with ReLU activation and an additional dropout layer was added before the final softmax output layer, which predicted the binary classification outcome. This sequential CNN was designed to be lightweight yet effective for baseline performance on psoriasis detection tasks using image-based input.

```python
# Build CNN Model
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)),
    MaxPooling2D((2, 2)),
    Dropout(0.25),

    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Dropout(0.25),

    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Dropout(0.25),

    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(2, activation='softmax')  # Output layer for 2 classes
])
```

Figure 7: CNN Model

```python
[ ] # Compile the model
model.compile(optimizer=Adam(learning_rate=1e-4), loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

```python
# Train the model
history = model.fit(
    datagen.flow(X_train, y_train, batch_size=BATCH_SIZE),
    validation_data=(X_val, y_val),
    epochs=30,
    steps_per_epoch=len(X_train) // BATCH_SIZE,
    verbose=1
)
```

```
Epoch 1/30
13/13 ━━━━━━━━━━━━━━━━━━━━ 66s 5s/step - accuracy: 0.5337 - loss: 1.0674 - val_accuracy: 0.5463 - val_loss: 0.6675
Epoch 2/30
13/13 ━━━━━━━━━━━━━━━━━━━━ 7s 259ms/step - accuracy: 0.5938 - loss: 0.5716 - val_accuracy: 0.5463 - val_loss: 0.6677
Epoch 3/30
13/13 ━━━━━━━━━━━━━━━━━━━━ 61s 4s/step - accuracy: 0.5700 - loss: 0.6507 - val_accuracy: 0.6204 - val_loss: 0.6810
Epoch 4/30
13/13 ━━━━━━━━━━━━━━━━━━━━ 7s 260ms/step - accuracy: 0.5000 - loss: 0.6881 - val_accuracy: 0.6204 - val_loss: 0.6807
Epoch 5/30
```

Figure 8:CNN Model Training

```
[ ] model.summary()
```

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_6 (MaxPooling2D)	(None, 111, 111, 32)	0
dropout_8 (Dropout)	(None, 111, 111, 32)	0
conv2d_7 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_7 (MaxPooling2D)	(None, 54, 54, 64)	0
dropout_9 (Dropout)	(None, 54, 54, 64)	0
conv2d_8 (Conv2D)	(None, 52, 52, 128)	73,856
max_pooling2d_8 (MaxPooling2D)	(None, 26, 26, 128)	0
dropout_10 (Dropout)	(None, 26, 26, 128)	0
flatten_2 (Flatten)	(None, 86528)	0
dense_4 (Dense)	(None, 128)	11,075,712
dropout_11 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 2)	258

Total params: 33,507,656 (127.82 MB)
Trainable params: 11,169,218 (42.61 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 22,338,438 (85.21 MB)

Figure 9:CNN Model Summary

Approach 2 – EfficientNetB0

An advanced psoriasis detection model was implemented using the EfficientNetV2B0 architecture as the backbone. Leveraging transfer learning, the base model was fine-tuned by unfreezing all layers to adapt its high-level feature representations to the psoriasis classification task. The output from EfficientNetV2B0 was passed through a Flatten layer followed by two dense layers with ReLU activations and dropout for regularization. L2 regularizers were also applied to mitigate overfitting. The final softmax layer predicted two classes: normal skin and psoriasis-affected skin. This model structure was chosen for its balance of accuracy, speed, and efficiency, making it suitable for deployment in real-time clinical diagnostics.

```
[ ]  # Load EfficientNetV2B0
     base_model = EfficientNetV2B0(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

     Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/efficientnet_v2
     24274472/24274472 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step

     # Fine-tuning: Unfreeze more layers of the base model
     for layer in base_model.layers:
         layer.trainable = True

[ ]
     model = Sequential([
         base_model,
         Flatten(),  # Replace GlobalAveragePooling2D with Flatten
         Dense(256, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.01)),
         Dropout(0.5),
         Dense(128, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.01)),
         Dropout(0.5),
         Dense(2, activation='softmax')  # Two classes: Normal Skin and Psoriasis Skin
     ])
```

Figure 10:EfficientNetV2B0 Model

```
     # Compile the model
     model.compile(optimizer=Adam(learning_rate=1e-4), loss='sparse_categorical_crossentropy', metrics=['accuracy'])
     #model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

     # Train the model
     history = model.fit(
         train_gen,
         validation_data=(X_val, y_val),
         epochs=EPOCHS
     )

     Epoch 1/30
     14/14 ━━━━━━━━━━━━━━━━━━━━ 189s 9s/step - accuracy: 0.7858 - loss: 7.3729 - val_accuracy: 0.7593 - val_loss: 7.2450
     Epoch 2/30
     14/14 ━━━━━━━━━━━━━━━━━━━━ 123s 8s/step - accuracy: 0.9561 - loss: 6.7966 - val_accuracy: 0.8611 - val_loss: 7.0346
     Epoch 3/30
     14/14 ━━━━━━━━━━━━━━━━━━━━ 141s 8s/step - accuracy: 0.9607 - loss: 6.6518 - val_accuracy: 0.9074 - val_loss: 6.6358
     Epoch 4/30
     14/14 ━━━━━━━━━━━━━━━━━━━━ 117s 8s/step - accuracy: 0.9729 - loss: 6.4133 - val_accuracy: 0.9907 - val_loss: 6.2688
     Epoch 5/30
     14/14 ━━━━━━━━━━━━━━━━━━━━ 152s 9s/step - accuracy: 0.9854 - loss: 6.2727 - val_accuracy: 0.9907 - val_loss: 6.1094
     Epoch 6/30
     14/14 ━━━━━━━━━━━━━━━━━━━━ 138s 9s/step - accuracy: 0.9767 - loss: 6.0861 - val_accuracy: 0.9907 - val_loss: 5.9301
     Epoch 7/30
```

Figure 11::EfficientNetV2B0 Model Training

```
[ ] model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
efficientnetv2-b0 (Functional)	(None, 7, 7, 1280)	5,919,312
flatten (Flatten)	(None, 62720)	0
dense (Dense)	(None, 256)	16,056,576
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32,896
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 2)	258

Total params: 22,009,042 (83.96 MB)
Trainable params: 21,948,434 (83.73 MB)
Non-trainable params: 60,608 (236.75 KB)

Figure 12:EfficientNetV2B0 Model Summary

Approach 3 – ResNet50

For psoriasis detection, a model was constructed using the ResNet50 architecture as the feature extractor, leveraging its deep residual connections for enhanced image representation. The pre-trained ResNet50 was integrated without the top classification layers, followed by a Flatten layer, a dense layer with ReLU activation, and a final softmax layer for binary classification between normal and psoriasis-affected skin. Dropout regularization was applied to prevent overfitting. This model benefited from the rich hierarchical features of ResNet50, making it suitable for identifying complex dermatological patterns within the skin lesion images.

```python
# Split data into training and validation sets
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

# Load pre-trained ResNet50 model
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_t
94765736/94765736 ━━━━━━━━━━━━━━━━━━━━ 1s 0us/step
```

```python
# Add custom layers
model = Sequential([
    base_model,
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(2, activation='softmax')  # 2 categories: Normal Skin, Psoriasis Skin
])
```

Figure 13:ResNet50 Model

```python
# Compile the model
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Train the model
history = model.fit(X_train, y_train, validation_data=(X_val, y_val), epochs=50, batch_size=BATCH_SIZE)
```

```
Epoch 1/50
14/14 ━━━━━━━━━━━━━━━━━━━━ 68s 2s/step - accuracy: 0.9723 - loss: 0.1479 - val_accuracy: 0.5463 - val_loss: 52.2964
Epoch 2/50
14/14 ━━━━━━━━━━━━━━━━━━━━ 4s 286ms/step - accuracy: 0.9702 - loss: 0.1030 - val_accuracy: 0.5648 - val_loss: 24.4308
Epoch 3/50
14/14 ━━━━━━━━━━━━━━━━━━━━ 4s 289ms/step - accuracy: 0.9742 - loss: 0.1150 - val_accuracy: 0.5648 - val_loss: 6.3406
Epoch 4/50
14/14 ━━━━━━━━━━━━━━━━━━━━ 4s 289ms/step - accuracy: 0.9706 - loss: 0.1112 - val_accuracy: 0.5833 - val_loss: 27.8913
Epoch 5/50
14/14 ━━━━━━━━━━━━━━━━━━━━ 4s 290ms/step - accuracy: 0.9759 - loss: 0.0708 - val_accuracy: 0.5556 - val_loss: 18.6540
Epoch 6/50
14/14 ━━━━━━━━━━━━━━━━━━━━ 4s 294ms/step - accuracy: 0.9772 - loss: 0.4437 - val_accuracy: 0.5556 - val_loss: 30.8435
Epoch 7/50
14/14 ━━━━━━━━━━━━━━━━━━━━ 4s 294ms/step - accuracy: 0.9852 - loss: 0.0575 - val_accuracy: 0.5463 - val_loss: 57.8387
Epoch 8/50
14/14 ━━━━━━━━━━━━━━━━━━━━ 4s 298ms/step - accuracy: 0.9322 - loss: 0.3266 - val_accuracy: 0.5463 - val_loss: 62765.0977
Epoch 9/50
14/14 ━━━━━━━━━━━━━━━━━━━━ 4s 300ms/step - accuracy: 0.9783 - loss: 0.0851 - val_accuracy: 0.5463 - val_loss: 74872.5547
Epoch 10/50
14/14 ━━━━━━━━━━━━━━━━━━━━ 4s 299ms/step - accuracy: 0.9773 - loss: 0.0868 - val_accuracy: 0.5556 - val_loss: 68383.9922
Epoch 11/50
```

Figure 14:ResNet50 Model Training

```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 7, 7, 2048)	23,587,712
flatten (Flatten)	(None, 100352)	0
dense (Dense)	(None, 128)	12,845,184
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258

Total params: 109,193,224 (416.54 MB)
Trainable params: 36,380,034 (138.78 MB)
Non-trainable params: 53,120 (207.50 KB)
Optimizer params: 72,760,070 (277.56 MB)

Figure 15:ResNet50 Model Summary

Approach 4 – InceptionResNetV2

For the psoriasis detection task, the InceptionResNetV2 architecture was utilized as a base model, benefiting from its hybrid design that merges the strengths of both Inception modules and residual connections. The model was initialized with ImageNet weights and fine-tuned using a transfer learning approach, with added custom layers including global average pooling, dropout for regularization, and dense layers for final classification into normal and psoriasis-affected skin. This setup enables effective extraction of both global and local features while maintaining model efficiency, making it well-suited for dermatological image classification.

```python
# Load InceptionResNetV2 base model
base_model = InceptionResNetV2(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Add custom layers
model = Sequential([
    base_model,
    GlobalAveragePooling2D(),
    Dropout(0.5),
    Dense(128, activation='relu'),
    Dropout(0.3),
    Dense(2, activation='softmax')  # Output layer for 2 classes
])
```

Figure 16:InceptionResNetV2 Model

```python
# Compile the model
model.compile(optimizer=Adam(learning_rate=1e-4), loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Train the model
history = model.fit(
    datagen.flow(X_train, y_train, batch_size=BATCH_SIZE),
    validation_data=(X_val, y_val),
    epochs=20,
    steps_per_epoch=len(X_train) // BATCH_SIZE,
    verbose=1
)
```

```
/opt/conda/lib/python3.10/contextlib.py:153: UserWarning: Your input ran out of data; interrupting training. Make sure
t least `steps_per_epoch * epochs` batches. You may need to use the `.repeat()` function when building your dataset.
  self.gen.throw(typ, value, traceback)
13/13 ━━━━━━━━━━━━━━━━━━━━ 1s 54ms/step - accuracy: 0.9375 - loss: 0.2086 - val_accuracy: 0.9630 - val_loss: 0.1215
Epoch 3/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 10s 585ms/step - accuracy: 0.9749 - loss: 0.1154 - val_accuracy: 0.9722 - val_loss: 0.0491
Epoch 4/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 1s 51ms/step - accuracy: 0.9688 - loss: 0.1709 - val_accuracy: 0.9722 - val_loss: 0.0482
Epoch 5/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 10s 600ms/step - accuracy: 0.9920 - loss: 0.0290 - val_accuracy: 0.9815 - val_loss: 0.0342
```

Figure 17:InceptionResNetV2 Model Training

```
[5]: model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
inception_resnet_v2 (Functional)	(None, 5, 5, 1536)	54,336,736
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1536)	0
dropout (Dropout)	(None, 1536)	0
dense (Dense)	(None, 128)	196,736
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258

Total params: 163,480,104 (623.63 MB)
Trainable params: 54,473,186 (207.80 MB)
Non-trainable params: 60,544 (236.50 KB)
Optimizer params: 108,946,374 (415.60 MB)

Figure 18:InceptionResnetV2 Model Summary

Approach 5 – YOLOv8 (Best Model)

The YOLOv8 object detection model was utilized for psoriasis lesion detection and emerged as the final selected model due to its superior performance and real-time inference capabilities. The model was initialized using pre-trained weights from the COCO dataset and fine-tuned using a custom-annotated dataset, where psoriasis lesions were labeled using bounding boxes generated via the CVAT (Computer Vision Annotation Tool). The model configuration was defined through a YAML file, and the training was carried out over 100 epochs with an input resolution of 640×640 pixels and a batch size of 16.

YOLOv8's lightweight architecture allowed efficient training and deployment on GPU-based environments while maintaining high detection accuracy. The model architecture supported precise localization of affected skin regions and robust feature extraction even with limited training data. This made YOLOv8 an ideal solution for integrating into the DermaScope AI system for practical, real-time clinical use.

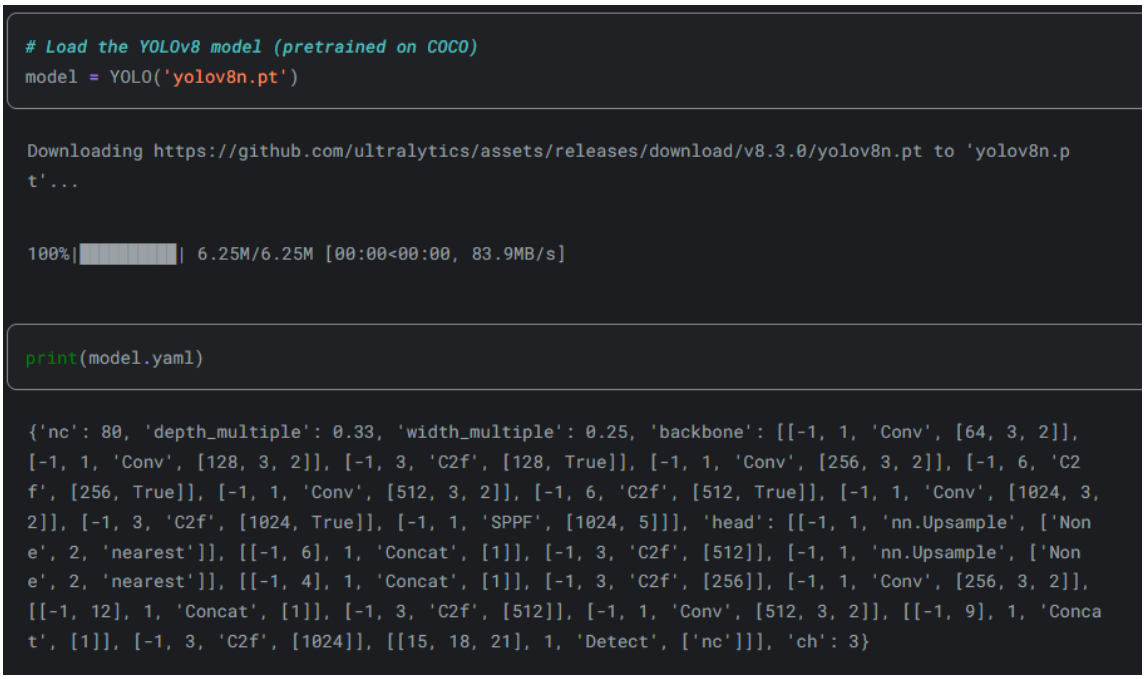


Figure 19: YOLO Model

```python
# Train the YOLOv8 model
train_results = model.train(
    data='data.yaml',         # Path to the dataset YAML file
    epochs=100,                # Number of epochs
    imgsz=640,                # Image size
    batch=16,                 # Batch size
    name='psoriasis_detection_final', # Name for the training session
    project='/kaggle/working/yolo_project',  # Path to save results
    save=True  # Ensure the best model is saved
)
```

```
ltralytics 8.3.75 🚀 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15095MiB)
gine/trainer: task=detect, mode=train, model=yolov8n.pt, data=data.yaml, epochs=100, time=None, patience=100,
wnloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
0%|██████████| 755k/755k [00:00<00:00, 18.0MB/s]
verriding model.yaml nc=80 with nc=2

                   from  n    params  module                                       arguments
 0                   -1  1       464  ultralytics.nn.modules.conv.Conv             [3, 16, 3, 2]
 1                   -1  1      4672  ultralytics.nn.modules.conv.Conv             [16, 32, 3, 2]
 2                   -1  1      7360  ultralytics.nn.modules.block.C2f             [32, 32, 1, True]
 3                   -1  1     18560  ultralytics.nn.modules.conv.Conv             [32, 64, 3, 2]
 4                   -1  2     49664  ultralytics.nn.modules.block.C2f             [64, 64, 2, True]
 5                   -1  1     73984  ultralytics.nn.modules.conv.Conv             [64, 128, 3, 2]
 6                   -1  2    197632  ultralytics.nn.modules.block.C2f             [128, 128, 2, True]
 7                   -1  1    295424  ultralytics.nn.modules.conv.Conv             [128, 256, 3, 2]
```

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      1/100       2.2G     0.6276      2.142      1.282         45        640: 100%|██████████| 27/27 [00:05<00:00,  4.85it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:01<00:00,  2.01it/s]
                   all        108        108      0.787       0.35      0.619       0.31

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      2/100      2.14G     0.5575      1.109      1.224         52        640: 100%|██████████| 27/27 [00:04<00:00,  6.06it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:00<00:00,  4.64it/s]
                   all        108        108      0.363       0.657      0.486      0.282

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      3/100      2.16G     0.5968      1.066      1.237         49        640: 100%|██████████| 27/27 [00:04<00:00,  6.51it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:00<00:00,  4.99it/s]
                   all        108        108      0.628       0.824      0.732      0.501

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      4/100      2.14G     0.6175      1.025      1.274         50        640: 100%|██████████| 27/27 [00:04<00:00,  6.55it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:00<00:00,  5.58it/s]
                   all        108        108       0.42       0.693       0.52      0.257
```

Figure 20:YOLO Model Training

2.1.7. Psoriasis Severity Classification

2.1.7.1. Model Building and Training

For the psoriasis severity classification task, multiple CNN-based models including a custom Convolutional Neural Network (CNN), EfficientNetB3, and MobileNetV2 were explored and fine-tuned using transfer learning techniques. These models were chosen for their effectiveness in medical imaging applications where accurate feature extraction and efficient computation are critical. The primary goal was to classify psoriasis lesions into three expert-labeled severity levels: mild, moderate, and severe.

The baseline CNN model provided a foundational understanding of performance benchmarks and was valuable for early experimentation. EfficientNetB3 was selected for its compound scaling strategy, which balances network depth, width, and resolution for optimal performance. MobileNetV2, known for its lightweight structure, was further enhanced using architectural modules to boost classification accuracy while maintaining real-time inference capability. All models were trained on a well-preprocessed dataset, using standardized input sizes and data augmentation techniques to ensure robustness across various skin tones and lesion presentations.

2.1.7.1.1. Ghost Module for Lightweight Feature Representation

To further optimize the classification model, the Ghost Module was integrated into the MobileNetV2 architecture. The Ghost Module improves efficiency by generating more feature maps using inexpensive linear operations rather than standard convolutions. This significantly reduces the number of parameters and computational cost while preserving representational power. By incorporating Ghost modules, the model achieved a smaller footprint, making it ideal for deployment in low-resource environments such as mobile health applications or embedded clinical systems, without sacrificing prediction quality.

```python
# Lightweight GhostModule
class GhostModule(Layer):
    def __init__(self, filters, ratio=2, kernel_size=1, **kwargs):
        super(GhostModule, self).__init__(**kwargs)
        self.filters = filters
        self.ratio = ratio
        self.kernel_size = kernel_size

    def build(self, input_shape):
        out_channels = self.filters // self.ratio
        self.primary_conv = Conv2D(out_channels, kernel_size=self.kernel_size, padding='same', activation='relu')
        self.ghost_conv = Conv2D(out_channels, kernel_size=3, padding='same', activation='relu')
        super(GhostModule, self).build(input_shape)

    def call(self, inputs):
        primary_out = self.primary_conv(inputs)
        ghost_out = self.ghost_conv(primary_out)
        return Concatenate()([primary_out, ghost_out])
```

Figure 21:Ghost Module Implementation

2.1.7.1.2. Mix Convolution Block for Multi-Scale Feature Extraction

To enhance the model's ability to capture diverse lesion patterns, Mix Convolution Blocks (MixConv) were also integrated into the classification architecture. MixConv combines multiple kernel sizes within a single convolutional layer, allowing the network to simultaneously extract both fine-grained and large-scale lesion features. This is particularly beneficial in psoriasis severity assessment, where lesions can vary significantly in size, shape, and texture. By enabling multi-scale feature learning, MixConv improved the model's robustness and accuracy in distinguishing between different severity levels, ultimately contributing to more reliable classification outcomes.

```python
# Simplified mixconv block
def mixconv_block(input_tensor, filters=128, kernel_sizes=[3, 5]):
    convs = [GhostModule(filters=filters)(input_tensor) for k in kernel_sizes]
    return Concatenate()(convs)
```

Figure 22:Mix Conv Block Implementation

Approach 1 – CNN

A baseline Convolutional Neural Network (CNN) model was implemented using a sequential architecture to establish a foundational approach for psoriasis severity classification. The model consisted of two convolutional layers with ReLU activation followed by max pooling, effectively reducing spatial dimensions while extracting key features. After flattening, the network included a fully connected dense layer and a dropout layer for regularization, concluding with a softmax output layer to classify the input into three severity categories. This simple yet effective architecture was used as a reference point for comparing the performance and complexity of more advanced models in the study.

```
[ ]  # Model definition
     model = Sequential([
         Conv2D(32, (3, 3), activation='relu', input_shape=(IMG_SIZE, IMG_SIZE, 3)),
         MaxPooling2D(pool_size=(2, 2)),
         Conv2D(64, (3, 3), activation='relu'),
         MaxPooling2D(pool_size=(2, 2)),
         Flatten(),
         Dense(128, activation='relu'),
         Dropout(0.5),
         Dense(len(os.listdir(DATA_DIR)), activation='softmax')  # Output layer with 3 classes
     ])

     /usr/local/lib/python3.10/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning:
       super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

Figure 23:CNN Model

```
[ ]
     # Compile the model
     model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

     # Train the model
     history = model.fit(train_generator, epochs=EPOCHS, validation_data=(X_test, y_test))

     Epoch 1/30
     8/8 ━━━━━━━━━━━━━━━━━━━━ 26s 3s/step - accuracy: 0.4157 - loss: 1.0426 - val_accuracy: 0.5167 - val_loss: 1.0133
     Epoch 2/30
     8/8 ━━━━━━━━━━━━━━━━━━━━ 25s 3s/step - accuracy: 0.5107 - loss: 1.0026 - val_accuracy: 0.5667 - val_loss: 0.9855
     Epoch 3/30
     8/8 ━━━━━━━━━━━━━━━━━━━━ 40s 3s/step - accuracy: 0.4918 - loss: 0.9893 - val_accuracy: 0.5667 - val_loss: 0.9472
     Epoch 4/30
     8/8 ━━━━━━━━━━━━━━━━━━━━ 41s 3s/step - accuracy: 0.4701 - loss: 0.9895 - val_accuracy: 0.6000 - val_loss: 0.9517
     Epoch 5/30
     8/8 ━━━━━━━━━━━━━━━━━━━━ 41s 3s/step - accuracy: 0.6043 - loss: 0.9106 - val_accuracy: 0.5167 - val_loss: 0.9912
```

Figure 24:CNN Model Training

```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
flatten (Flatten)	(None, 186624)	0
dense (Dense)	(None, 128)	23,888,000
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 3)	387

Total params: 23,907,779 (91.20 MB)
Trainable params: 23,907,779 (91.20 MB)
Non-trainable params: 0 (0.00 B)

Figure 25:CNN Model Summary

Approach 2 – EfficientNetB3

A customized severity classification model was developed using EfficientNetB3, integrated with a MixConv block and Ghost Modules to enhance feature extraction while maintaining efficiency. The model leverages pre-trained weights and fine-tunes the top layers to adapt to psoriasis-specific features. Ghost Modules generate lightweight feature maps, and MixConv enables multi-scale pattern detection. This architecture is designed for high-resolution image inputs and optimized for scalable, real-world clinical deployment.

```python
    #     return model
    def build_efficientnetb3_model(input_shape=(300, 300, 3), num_classes=3):
        base_model = EfficientNetB3(weights="imagenet", include_top=False, input_shape=input_shape)

        # Fine-tune: unfreeze last 60 layers
        base_model.trainable = True
        for layer in base_model.layers[:-60]:
            layer.trainable = False

        inputs = Input(shape=input_shape)
        x = base_model(inputs, training=True)

        # ✅ Apply MixConv block with GhostModule
        x = mixconv_block(x, filters=64)  # You can test with 64 or 128

        x = GlobalAveragePooling2D()(x)
        x = Dense(512, activation="relu")(x)
        x = BatchNormalization()(x)
        x = Dropout(0.5)(x)
        outputs = Dense(num_classes, activation="softmax")(x)

        model = Model(inputs, outputs)
        return model
```

Figure 26:EfficientNetB3 Model

```python
# Build and compile
model = build_efficientnetb3_model()
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)

# Train
history = model.fit(
    datagen.flow(X_train, y_train, batch_size=16),
    validation_data=(X_val, y_val),
    epochs=30,
    #callbacks=[lr_schedule]
)
```

```
Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb3_notop.h5
43941136/43941136 ━━━━━━━━━━━━━━━━━━━━ 2s 0us/step
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: You
  self._warn_if_super_not_called()
Epoch 1/30
15/15 ━━━━━━━━━━━━━━━━━━━━ 164s 8s/step - accuracy: 0.4544 - loss: 1.6050 - val_accuracy: 0.3667 - val_loss: 1.1120
Epoch 2/30
15/15 ━━━━━━━━━━━━━━━━━━━━ 134s 8s/step - accuracy: 0.4581 - loss: 1.6242 - val_accuracy: 0.3667 - val_loss: 1.1013
Epoch 3/30
15/15 ━━━━━━━━━━━━━━━━━━━━ 140s 7s/step - accuracy: 0.3888 - loss: 1.5238 - val_accuracy: 0.3667 - val_loss: 1.0931
```

Figure 27:EfficientnetB3 Model Training

```
model.summary()
```

Model: "functional"

Layer (type)	Output Shape	Param #	Connected to
input_layer_1 (InputLayer)	(None, 300, 300, 3)	0	-
efficientnetb3 (Functional)	(None, 10, 10, 1536)	10,783,535	input_layer_1[0][0]
ghost_module (GhostModule)	(None, 10, 10, 64)	58,432	efficientnetb3[0][0]
ghost_module_1 (GhostModule)	(None, 10, 10, 64)	58,432	efficientnetb3[0][0]
ghost_module_2 (GhostModule)	(None, 10, 10, 64)	58,432	efficientnetb3[0][0]
batch_normalization (BatchNormalization)	(None, 10, 10, 64)	256	ghost_module[0][0]
batch_normalization_1 (BatchNormalization)	(None, 10, 10, 64)	256	ghost_module_1[0][0]
batch_normalization_2 (BatchNormalization)	(None, 10, 10, 64)	256	ghost_module_2[0][0]
dropout (Dropout)	(None, 10, 10, 64)	0	batch_normalization[0…
dropout_1 (Dropout)	(None, 10, 10, 64)	0	batch_normalization_1…
dropout_2 (Dropout)	(None, 10, 10, 64)	0	batch_normalization_2…
concatenate_3 (Concatenate)	(None, 10, 10, 192)	0	dropout[0][0], dropout_1[0][0], dropout_2[0][0]
global_average_pooling2d (GlobalAveragePooling2D)	(None, 192)	0	concatenate_3[0][0]
dense (Dense)	(None, 512)	98,816	global_average_poolin…
batch_normalization_3 (BatchNormalization)	(None, 512)	2,048	dense[0][0]
dropout_3 (Dropout)	(None, 512)	0	batch_normalization_3…
dense_1 (Dense)	(None, 3)	1,539	dropout_3[0][0]

Total params: 22,123,030 (84.39 MB)
Trainable params: 5,530,513 (21.10 MB)
Non-trainable params: 5,531,489 (21.10 MB)
Optimizer params: 11,061,028 (42.19 MB)

Figure 28:EfficientNetB3 Model Summary

Approach 3 – MobileNetV2 (Best Model)

For the psoriasis severity classification task, the selected model was a customized version of MobileNetV2, enhanced with Ghost Modules and a MixConv block. This architecture was designed to achieve a balance between computational efficiency and robust feature representation. The base MobileNetV2 model, pre-trained on the ImageNet dataset, was partially frozen to retain its learned low-level features, while the top layers were fine-tuned to adapt to the specific patterns of psoriasis severity. A MixConv block was integrated after the base model to enable multi-scale feature extraction, allowing the model to effectively capture variations in lesion texture, shape, and structure.

Following the MixConv block, two Ghost Modules were used in parallel to efficiently generate additional feature maps using lightweight linear operations. These modules reduce redundancy in feature representations and significantly minimize computational overhead without compromising representational strength. The outputs of the Ghost Modules were concatenated, followed by a Global Average Pooling layer, a Dense layer with ReLU activation, Batch Normalization, Dropout, and a final softmax layer for multi-class classification. The model was constructed to classify input images into three expert-labeled severity categories: mild, moderate, and severe.

This architecture was ultimately selected for its lightweight design, scalability, and suitability for real-time deployment, particularly in clinical environments where both performance and efficiency are critical.

```python
def build_mobilenetv2_model(input_shape=(224, 224, 3), num_classes=3):
    base_model = MobileNetV2(weights="imagenet", include_top=False, input_shape=input_shape)

    # Fine-tune top layers only
    base_model.trainable = False
    for layer in base_model.layers[:-20]:
        layer.trainable = False

    inputs = Input(shape=input_shape)
    x = base_model(inputs, training=True)
    x = mixconv_block(x, filters=128)  # reduced filters

    x = GlobalAveragePooling2D()(x)
    x = Dense(256, activation="relu")(x)
    x = BatchNormalization()(x)
    #x = Dropout(0.5)(x)
    x = Dropout(0.5)(x)
    outputs = Dense(num_classes, activation="softmax")(x)

    model = Model(inputs, outputs)
    return model
```

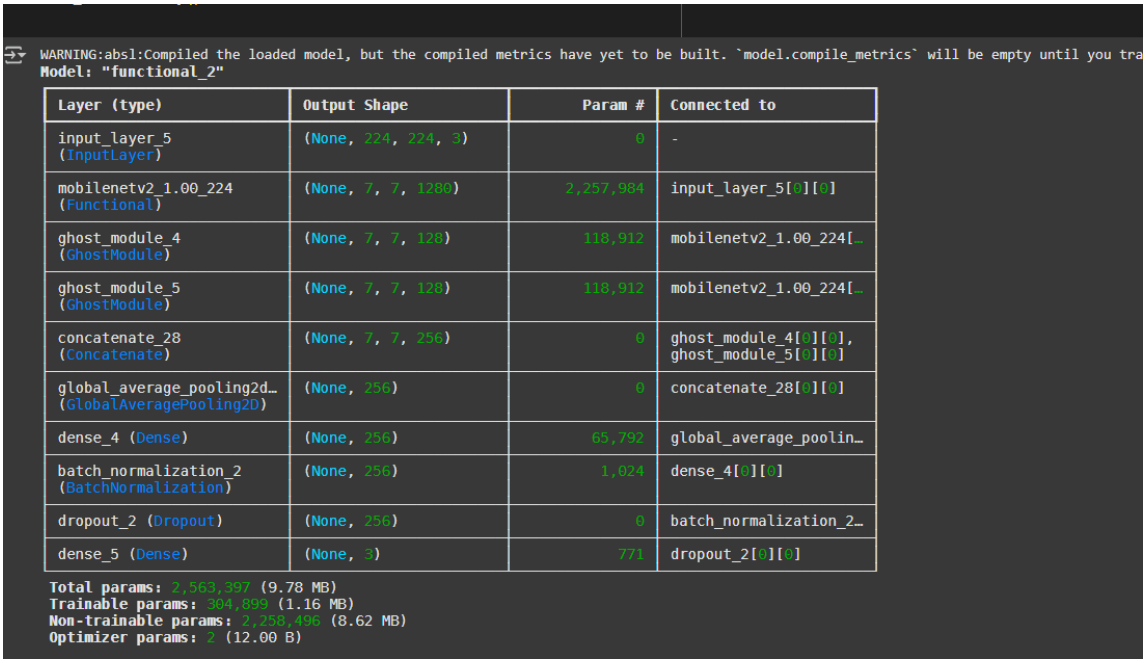
Figure 29: MobileNetV2 Model

```python
# Build and compile model
model = build_mobilenetv2_model()
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
    loss="sparse_categorical_crossentropy",
    metrics=["accuracy"]
)
```

```python
# Train the model
history = model.fit(
    datagen.flow(X_train, y_train, batch_size=16),
    validation_data=(X_val, y_val),
    epochs=30,
    class_weight=class_weight_dict
)
```

```
Epoch 1/30
15/15 ━━━━━━━━━━━━━━━━━━━━ 36s 2s/step - accuracy: 0.3303 - loss: 1.6913 - val_accuracy: 0.5167 - val_loss: 1.0651
Epoch 2/30
15/15 ━━━━━━━━━━━━━━━━━━━━ 18s 1s/step - accuracy: 0.4322 - loss: 1.3911 - val_accuracy: 0.5500 - val_loss: 0.9775
Epoch 3/30
15/15 ━━━━━━━━━━━━━━━━━━━━ 16s 1s/step - accuracy: 0.5854 - loss: 0.9979 - val_accuracy: 0.6167 - val_loss: 0.9283
Epoch 4/30
15/15 ━━━━━━━━━━━━━━━━━━━━ 16s 1s/step - accuracy: 0.5000 - loss: 1.1963 - val_accuracy: 0.5667 - val_loss: 0.9041
Epoch 5/30
15/15 ━━━━━━━━━━━━━━━━━━━━ 24s 1s/step - accuracy: 0.5549 - loss: 1.0514 - val_accuracy: 0.6167 - val_loss: 0.8532
Epoch 6/30
```

Figure 30:MobileNetV2 Model training



```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you tra
Model: "functional_2"
```

Layer (type)	Output Shape	Param #	Connected to
input_layer_5 (InputLayer)	(None, 224, 224, 3)	0	-
mobilenetv2_1.00_224 (Functional)	(None, 7, 7, 1280)	2,257,984	input_layer_5[0][0]
ghost_module_4 (GhostModule)	(None, 7, 7, 128)	118,912	mobilenetv2_1.00_224[…
ghost_module_5 (GhostModule)	(None, 7, 7, 128)	118,912	mobilenetv2_1.00_224[…
concatenate_28 (Concatenate)	(None, 7, 7, 256)	0	ghost_module_4[0][0], ghost_module_5[0][0]
global_average_pooling2d… (GlobalAveragePooling2D)	(None, 256)	0	concatenate_28[0][0]
dense_4 (Dense)	(None, 256)	65,792	global_average_poolin…
batch_normalization_2 (BatchNormalization)	(None, 256)	1,024	dense_4[0][0]
dropout_2 (Dropout)	(None, 256)	0	batch_normalization_2…
dense_5 (Dense)	(None, 3)	771	dropout_2[0][0]

```
Total params: 2,563,397 (9.78 MB)
Trainable params: 304,899 (1.16 MB)
Non-trainable params: 2,258,496 (8.62 MB)
Optimizer params: 2 (12.00 B)
```

Figure 31:MobileNetV2 Model Summary

2.2. Commercialization Aspects of the Product

2.2.1. Target Market

The primary target market for DermaScope AI includes dermatology clinics, hospitals, telemedicine platforms, and individual dermatologists seeking reliable and automated tools for skin disease diagnosis. Additionally, the product is tailored to assist medical researchers, students, and health-tech companies involved in dermatological AI research. In the long term, the platform will also be adapted for use by general practitioners and patients, especially in rural or underserved areas where access to specialized dermatologists is limited. The system's ability to provide accurate detection and severity classification of psoriasis makes it an asset across both clinical and educational healthcare environments.

2.2.2. Revenue Streams

DermaScope AI will adopt a multi-faceted revenue model to support sustainable growth. This includes a subscription-based licensing model for hospitals, clinics, and telemedicine providers, with tiered pricing depending on usage volume and features. A pay-per-scan model will be introduced for individual practitioners and smaller clinics, enabling flexible use. A freemium model will also be offered, allowing users access to basic detection tools while advanced analytics, severity tracking, and expert-reviewed reports will be included in premium plans. Additional revenue will be generated through API integrations with Electronic Health Record (EHR) systems and collaborations with pharmaceutical companies for clinical trial assistance or targeted treatment suggestions.

2.2.3. Marketing Approach

Phase 1: Product Validation and Clinical Pilot Testing

Initiate partnerships with a small network of dermatologists and clinics to pilot and validate the product. Gather feedback from healthcare professionals on usability, accuracy, and clinical applicability to refine the system for real-world deployment.

Phase 2: Professional Outreach and Freemium Launch

Launch a freemium version targeting individual dermatologists and small clinics. Offer essential detection and classification features, while keeping advanced visual analytics and explainable AI outputs available under premium plans. Attend medical tech expos, conferences, and dermatology forums to showcase the tool.

Phase 3: Digital and B2B Marketing Campaigns

Run targeted digital campaigns through LinkedIn, medical journals, AI-healthcare newsletters, and Google Ads to attract healthcare decision-makers. Publish white papers and clinical validation studies to build credibility and visibility in the AI-healthcare domain.

Phase 4: Community Building and Medical Advocacy

Create an online community platform for dermatologists, researchers, and health AI professionals to collaborate, share feedback, and advocate for the adoption of AI in dermatology. Host expert webinars and panel discussions to foster engagement and thought leadership in the field.

Phase 5: Strategic Collaborations and Global Expansion

Establish partnerships with hospital networks, medical device manufacturers, and global health NGOs to expand usage in both urban and rural settings. Work with regulatory bodies and localization experts to adapt the platform for different languages, cultural contexts, and regional compliance standards to enter international markets.

2.3 Testing and Implementation

2.3.1. Functional Requirements

Psoriasis Lesion Detection:

- The system should allow users (patients or healthcare providers) to upload skin lesion images through a user-friendly interface for AI-based analysis.
- During the testing phase, multiple models were explored to detect and localize psoriasis-affected skin, including CNN-based models such as EfficientNetB0, ResNet50, and InceptionResNetV2, as well as the YOLOv8 object detection model.
- After comparative evaluation, YOLOv8 was selected as the final detection model due to its high speed, accuracy, and real-time performance.
- The final system should process the input image using YOLOv8, predict whether the skin is affected by psoriasis or not.
- The model should also return an accuracy or confidence score with each prediction to inform users of the detection reliability.

- Visual results, including detected regions and prediction outcomes, should be displayed on the interface for verification and review.

Severity Classification:

- The system should classify psoriasis lesions into one of three severity levels: mild, moderate, or severe, based on expert-labeled training data.
- The classification model (e.g., MobileNetV2 with Ghost Module and MixConv) should process segmented lesion regions and return a severity label with confidence scores.
- The severity results should be stored and displayed in a report format.

Explainable AI Integration:

- The system should generate Grad-CAM heatmaps to visualize which parts of the lesion contributed most to the classification decision.
- These visual explanations should be shown alongside classification results for dermatologist validation.

User Dashboard:

- Healthcare professionals should be able to access a web-based dashboard to view patient records, diagnostic results, and historical progression.

2.3.2. Non-Functional Requirements

Performance:

- The system should provide fast inference for lesion detection and severity classification (within a few seconds per image).
- All model components should be optimized for GPU to ensure real-time response capability.

Scalability:

- Cloud-based storage and processing should be employed to ensure support for growing datasets and users.

Security:

- All uploaded patient images and reports must be encrypted during transmission and at rest.
- Secure authentication mechanisms should be implemented for access to the dashboard and user data.

Usability:

- Upload instructions, tooltips, and result interpretations should be clearly presented.
- The user interface should be simple, intuitive, and easy to navigate for dermatologists and patients alike.

Accessibility:

- The system should support accessibility features such as keyboard navigation, screen reader compatibility, and contrast modes for visually impaired users.

Extensibility:

- The architecture should be modular to allow future integration of additional skin disease models (e.g., for acne, eczema).

Robustness:

- All model outputs should include confidence thresholds and fallback logic.

Maintainability:

- The codebase should follow best practices with modular structure, version control, and proper documentation.

2.3.3. Backend Implementation

The backend for the AI-powered dermatology system was developed using Flask, with a strong focus on modularity, scalability, and real-time image analysis. The application supports user authentication, disease detection, severity assessment, and dynamic image handling.

Import Statements

The system begins with importing essential libraries such as:

- Flask, request, jsonify, CORS – for handling API endpoints, requests, and enabling cross-origin communication.
- TensorFlow, PyTorch (torch), PIL, NumPy – to load and preprocess medical images for deep learning predictions.
- psycopg2.pool – to manage connections to the PostgreSQL database.
- dotenv and os – for secure access to environment variables.
- CloudinaryImageHandler (custom) – for uploading and retrieving skin images from the cloud.
- Keras custom_object_scope – used to register and load models containing the custom GhostModule.

GhostModule Definition

A custom convolutional block named GhostModule is defined, which mimics the behavior of lightweight CNN layers by generating more feature maps from fewer computations. It consists of a primary and ghost convolution that are concatenated, significantly reducing model complexity while maintaining performance. This module is registered globally using get_custom_objects().

Model Loading

Deep learning models are loaded at startup for different detection and classification tasks:

- YOLOv8 is used for real-time object-level psoriasis lesion detection.

```
psoriasis_model = YOLO('./models/psoriasis/Yolo_best_detection.pt')
psoriasis_model.eval()
acne_detection_model = tf.keras.models.load_model('./models/acne/acne_detection_model.h5', compile=False)
```

Figure 32:Detection Model Load

- CNN-based models (including MobileNetV2 + MixConv + GhostModule) are loaded for severity classification.

```
#psoriasis_severity_model = tf.keras.models.load_model('./models/psoriasis/psoriasis_sever
with custom_object_scope({'GhostModule': GhostModule}):
    psoriasis_severity_model = tf.keras.models.load_model(
        './models/psoriasis/psoriasis_severity_model_mobilevnet_mix_with_gost_80_.h5',
        compile=False
    )
```

Figure 33:Severity Assessment Model Load

Flask App and CORS Configuration

The Flask app is initialized with CORS settings allowing requests from both development and deployment frontend URLs. Environment variables are loaded using dotenv to configure database credentials securely.

```
load_dotenv()

app = Flask(__name__)

CORS(app, resources={
    r"/api/*": {
        "origins": ["http://localhost:5173", "http://localhost:4173"],
        "methods": ["GET", "POST", "PUT", "DELETE", "OPTIONS"],
        "allow_headers": ["Content-Type", "Authorization"]
    }
})
```

Figure 34:Flask App and CORS Configuration

Detection Endpoints

Each detection route handles image uploads, preprocessing, model inference, and result storage:

- /api/detect/psoriasis take image files, preprocess them (resize, normalize), and generate predictions. Results include detection status and confidence scores.
- YOLOv8 is used in the psoriasis detection route, returning bounding boxes and class-wise confidence scores.

```python
@app.route('/api/detect/psoriasis', methods=['POST'])
def detect_psoriasis():
    try:
        if 'image' not in request.files:
            return jsonify({'error': 'No image file provided'}), 400

        img_file = request.files['image']
        user_id = request.form.get('userId')
        body_part = request.form.get('bodyPart')

        if not user_id or not body_part:
            return jsonify({'error': 'User ID and body part are required'}), 400

        temp_path = f'temp_{user_id}.jpg'
        img = Image.open(img_file.stream)
        img.save(temp_path)

        CONFIDENCE_THRESHOLD = 0.7  # Adjust threshold to reduce false positives

        results = psoriasis_model.predict(temp_path)
        result = results[0]

       # Extract detections
        detections = result.boxes.data if result.boxes else []

        has_psoriasis = False
        max_confidence = 0.0  # Default confidence value
        #has_psoriasis = len(result.boxes) > 0
        #confidence = float(max(result.boxes.conf)) if has_psoriasis else 0.0

        if len(detections) > 0:
            for det in detections:
                class_id = int(det[5].item())  # Get class ID
                confidence = float(det[4].item())  # Get confidence score
                class_name = psoriasis_model.names[class_id]  # Get class name

                print(f"Detected Class: {class_name}, Confidence: {confidence}")
```

Figure 35:Detection Endpoint

Severity Assessment

Each detection route handles image uploads, preprocessing, model inference, and result storage:

- /api/assess/psoriasis_severity assess the severity of the disease using respective CNN models.
- These routes accept an image and the corresponding resultId, perform inference, and update the database record with the severity level and detailed class probabilities.

```python
@app.route('/api/assess/psoriasis_severity', methods=['POST'])
def assess_psoriasis_severity():
    try:
        if 'image' not in request.files:
            return jsonify({'error': 'No image file provided'}), 400

        img_file = request.files['image']
        result_id = request.form.get('resultId')

        if not result_id:
            return jsonify({'error': 'Result ID is required'}), 400

        img = Image.open(img_file.stream)
        img = img.convert('RGB')
        img = img.resize((224, 224))
        img_array = np.array(img) / 255.0
        img_array = np.expand_dims(img_array, axis=0)

        severity_prediction = psoriasis_severity_model.predict(img_array)
        severity_scores = severity_prediction[0]
        severity_labels = ['Mild', 'Moderate', 'Severe']

        severity_result = {
            'severity': {
                'level': severity_labels[np.argmax(severity_scores)],
                'confidence': round(float(np.max(severity_scores)) * 100, 2)
            },
            'severity_scores': {
                label: round(float(score) * 100, 2)
                for label, score in zip(severity_labels, severity_scores)
            }
        }
```

Figure 36:Severity Endpoint

Database Integration

The backend communicates with a PostgreSQL database using connection pooling. Each patient's test history is recorded, and corresponding skin analysis results are stored in the skin_test_results table. Queries are optimized for fetching both individual and aggregated patient data.

```
        conn = db_pool.getconn()
        try:
            cursor = conn.cursor()
            cursor.execute("""
                UPDATE skin_test_results
                SET test_results = test_results || %s::jsonb
                WHERE id = %s
                """,
                (json.dumps(severity_result), result_id)
            )
            conn.commit()
        except Exception as e:
            print(f"Database error: {str(e)}")
            raise
        finally:
            db_pool.putconn(conn)

        return jsonify(severity_result), 200

    except Exception as e:
        print(f"Error in assess_psoriasis_severity: {str(e)}")
        return jsonify({'error': str(e)}), 500
```

Figure 37:Database Integration

Backend Testing

Backend testing was conducted using Postman, a widely adopted API testing tool, to verify the functionality and reliability of the system's endpoints. Various POST and GET requests were sent to test critical functionalities such as image upload, prediction generation, user authentication, and data retrieval. Each request was checked for expected inputs and outputs to ensure that the server responded correctly and consistently. Special attention was given to validating JSON responses, status codes, and error handling. This process ensured that the backend APIs performed as intended, supported smooth integration with the frontend, and maintained secure and efficient communication with the database and machine learning models.

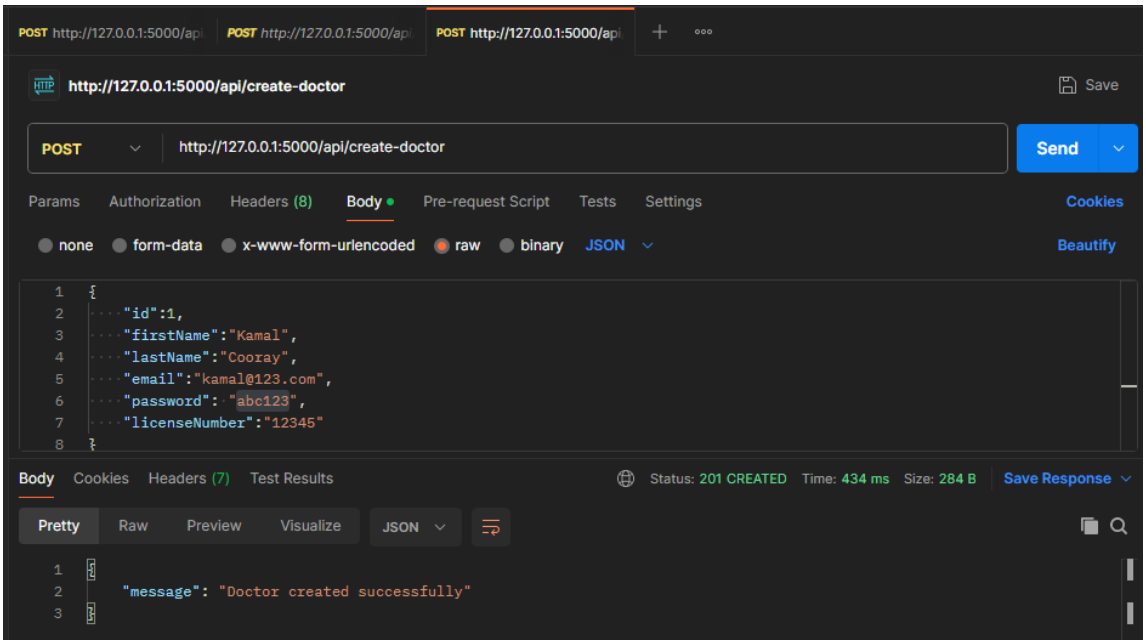


Figure 38:Backend Testing

2.3.4. Frontend Implementation

The frontend of the psoriasis detection and severity assessment system was developed using React, ensuring a responsive and user-friendly interface. Key features include image upload, body part selection, real-time result display, and dynamic visual feedback. The application integrates with backend APIs to provide accurate detection and severity classification while maintaining a smooth user experience.

Detection Function

The handleDetection() function:

- Sends a POST request to http://localhost:5000/api/detect/psoriasis
- Includes form data: image, user ID, and selected body part
- Parses and displays the result, updating the prediction state

This integrates with the YOLOv8 model on the backend for detecting psoriasis lesions

```
const handleDetection = async () => {
  if (!image || !user?.userId) {
    setError('Please upload an image and ensure you are logged in.');
    return;
  }

  if (!selectedBodyPart) {
    setError('Please select a body part.');
    return;
  }

  try {
    setIsScanning(true);
    setScanType('detection');
    const formData = new FormData();
    formData.append('image', image);
    formData.append('userId', user.userId.toString());
    formData.append('bodyPart', selectedBodyPart);

    const response = await fetch('http://localhost:5000/api/detect/psoriasis', {
      method: 'POST',
      body: formData,
    });

    if (!response.ok) {
      throw new Error('Failed to process image');
    }
```

Figure 39:Detection Frontend

Severity Assessment Function

The handleSeverityAssessment() function:

- Sends a POST request to http://localhost:5000/api/assess/psoriasis_severity
- Requires the resultId from the detection phase
- Receives severity level and scores and updates the UI accordingly

This works with a CNN-based severity classification model in the backend.

```
const handleSeverityAssessment = async () => {
  if (!image || !user?.userId || !prediction?.detection?.resultId) {
    setError('Missing required information for severity assessment.');
    return;
  }

  try {
    setIsScanning(true);
    setScanType('severity');
    const formData = new FormData();
    formData.append('image', image);
    formData.append('resultId', prediction.detection.resultId.toString());

    const response = await fetch('http://localhost:5000/api/assess/psoriasis_severity', {
      method: 'POST',
      body: formData,
    });

    if (!response.ok) {
      throw new Error('Failed to assess severity');
    }

    const result: SeverityResponse = await response.json();
    console.log("Severity result:", result);

    setPrediction(prev => ({
      ...prev,
      severity: {
        level: result.severity.level,
        confidence: result.severity.confidence,
        scores: result.severity_scores
      }
    }));
```

Figure 40:Severity Assessment Frontend

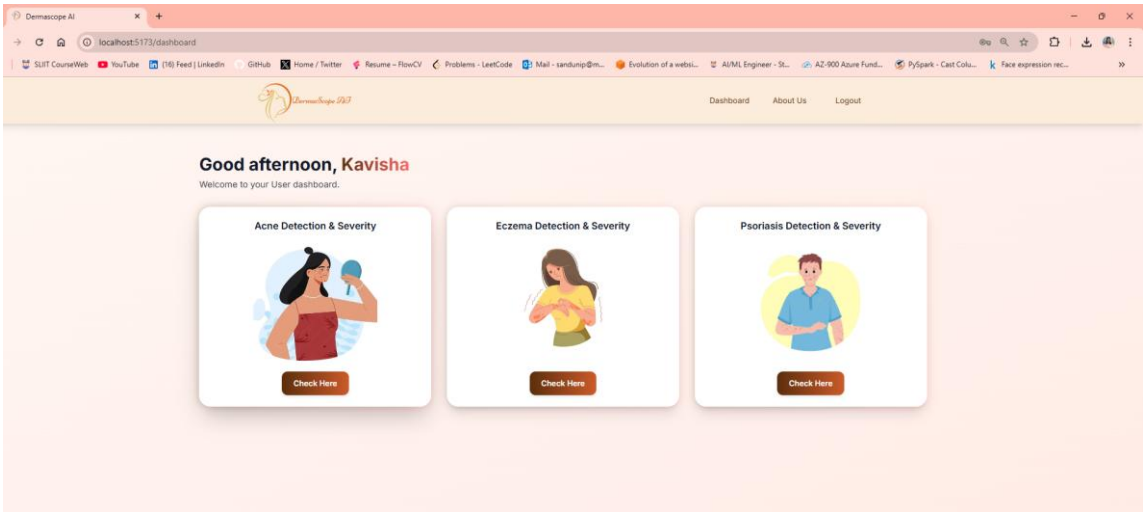


Figure 41:Dashboard

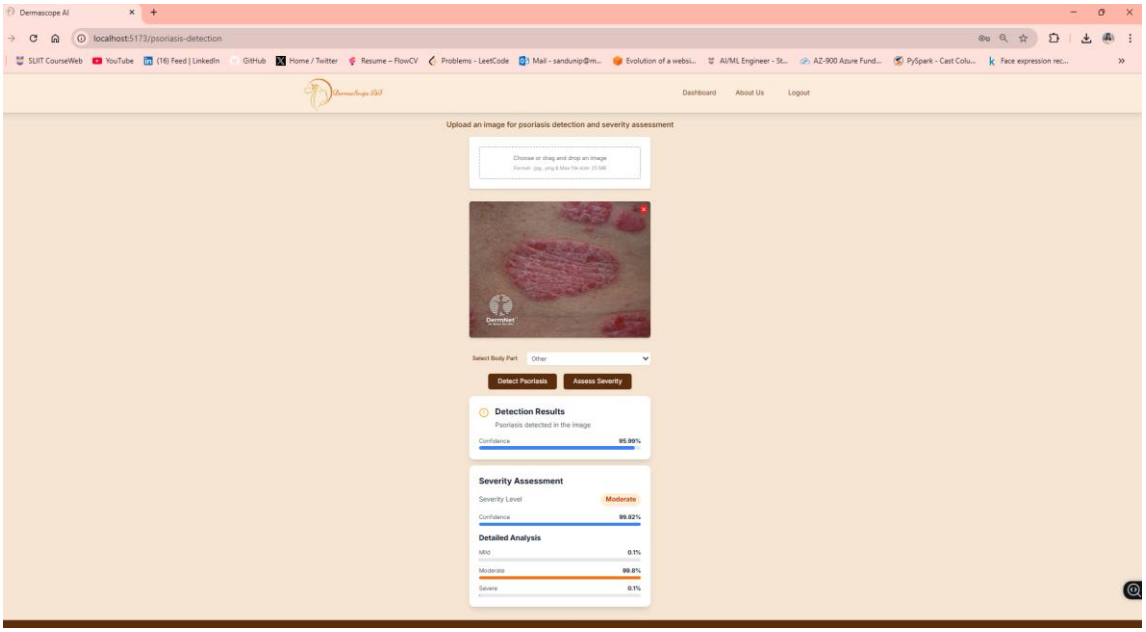


Figure 42:Detection and Severity Assessment

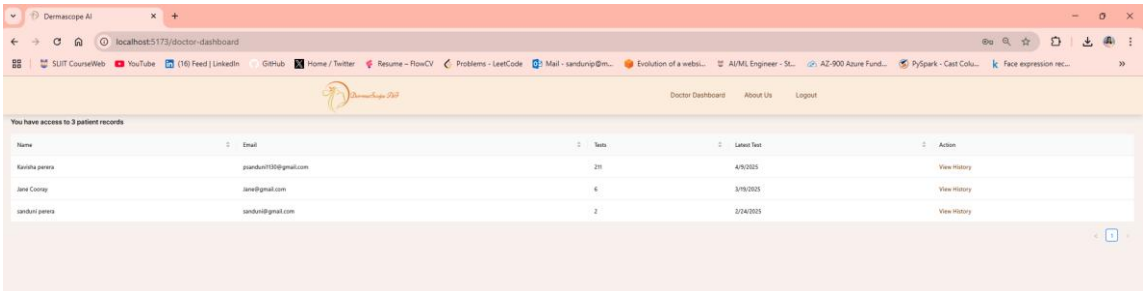


Figure 43:Doctor's Dashboard

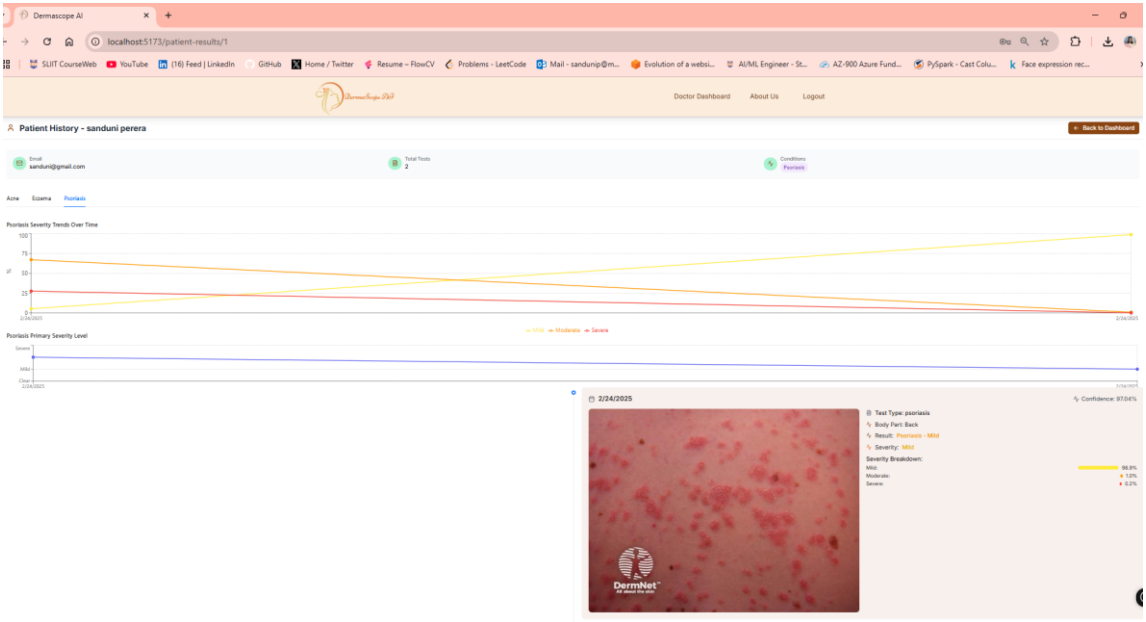


Figure 44:Patient History Page

3. RESULTS AND DISCUSSION

3.1. Model Results

3.1.1. Detection Model

Approach 1 – CNN

In the initial approach, a basic Convolutional Neural Network (CNN) model was used to classify skin images as either normal or psoriasis affected. The model achieved a validation accuracy of 77.78%, with a precision of 0.67, recall of 1.0, and an F1-score of 0.80 for psoriasis class. While the model effectively identified all psoriasis cases (as seen in the confusion matrix with zero false negatives), it struggled with correctly identifying normal skin, misclassifying 24 out of 59 images.

```
Validation Accuracy: 0.7777777910232544
Precision: 0.6712328767123288, Recall: 1.0, F1 Score: 0.8032786885245902
Classification Report:
                precision    recall  f1-score   support

   Normal Skin       1.00      0.59      0.74        59
Psoriasis Skin       0.67      1.00      0.80        49

      accuracy                           0.78       108
     macro avg       0.84      0.80      0.77       108
  weighted avg       0.85      0.78      0.77       108
```

Figure 45:CNN Classification Report

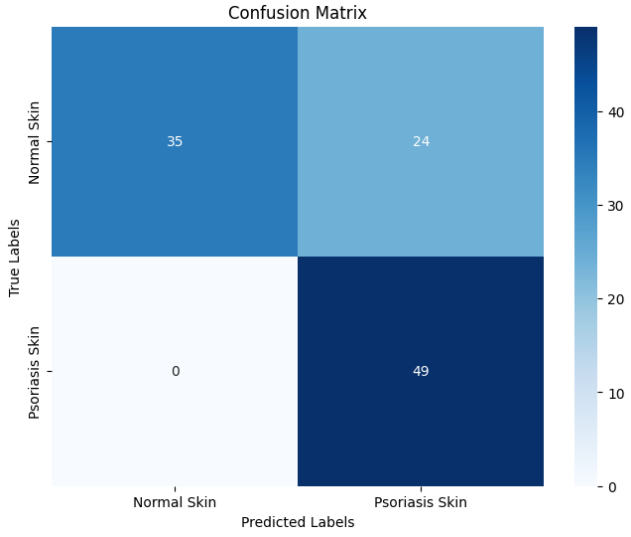


Figure 46::CNN Confusion Matrix

The training and validation accuracy curves showed significant fluctuation, indicating instability and possible overfitting. Although the CNN model showed potential, its limited generalization capacity and imbalance in class performance highlighted the need for a more optimized architecture.

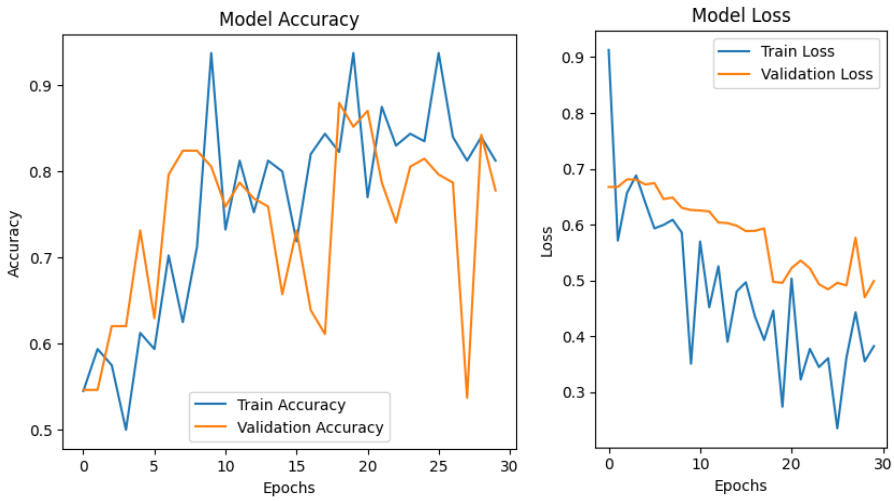


Figure 47:CNN Training and Validation Accuracy/Loss Curves

Approach 2 – EfficientNetB0

The EfficientNetB0 model demonstrated perfect classification performance on the validation dataset. The classification report reveals a precision, recall, and F1-score of 1.00 for both classes—Normal Skin and Psoriasis Skin—indicating no false positives or false negatives. The confusion matrix further supports this by showing that all 54 instances of each class were correctly classified, resulting in a total of 108 accurate predictions without any misclassification.

```
Validation Loss: 2.840066909790039
Validation Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1 Score: 1.0
Classification Report:
                precision    recall  f1-score   support

   Normal Skin       1.00      1.00      1.00        54
Psoriasis Skin       1.00      1.00      1.00        54

      accuracy                           1.00       108
     macro avg       1.00      1.00      1.00       108
  weighted avg       1.00      1.00      1.00       108
```

Figure 48:EfficientNetB0 Classification Report

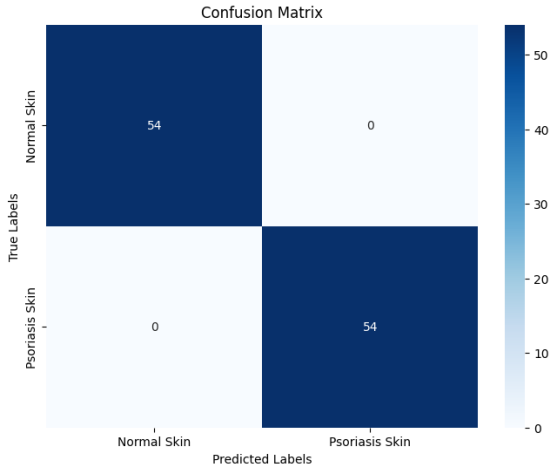


Figure 49:EfficientNetB0 Confusion Matrix

The training and validation curves for accuracy and loss depict a steady and consistent learning process. Training accuracy quickly approached 1.0 within a few epochs and remained stable, while validation accuracy followed a similar trend, indicating excellent generalization. Similarly, the training and validation loss gradually decreased across epochs, confirming that the model learned effectively without overfitting. These visualizations validate the robustness and reliability of the EfficientNetB0-based psoriasis classification model.

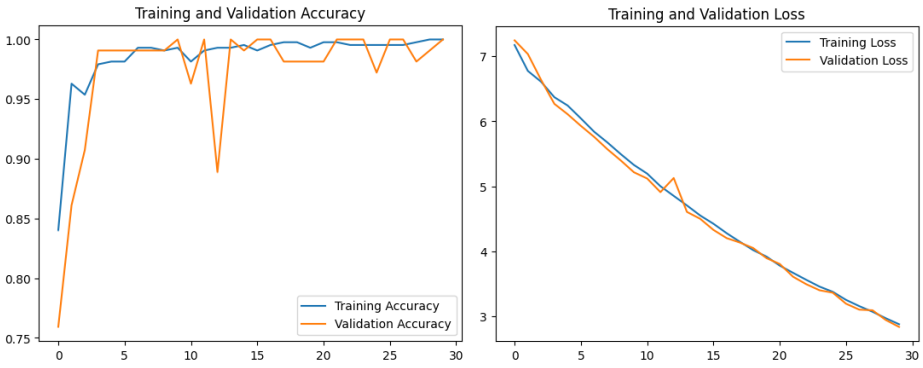


Figure 50:EfficienNetB0 Training and Validation Accuracy/Loss Curves

Approach 3 – ResNet50

The ResNet50 model achieved a 93% accuracy in classifying psoriasis and normal skin. It showed high precision and recall, especially for normal skin. The confusion matrix indicates only a few misclassifications, with 1 false positive and 7 false negatives, highlighting strong performance with slight room for improvement.

Classification Report:

	precision	recall	f1-score	support
Normal Skin	0.89	0.98	0.94	59
Psoriasis Skin	0.98	0.86	0.91	49
accuracy			0.93	108
macro avg	0.93	0.92	0.92	108
weighted avg	0.93	0.93	0.93	108

Figure 51:ResNet50 Classification Report

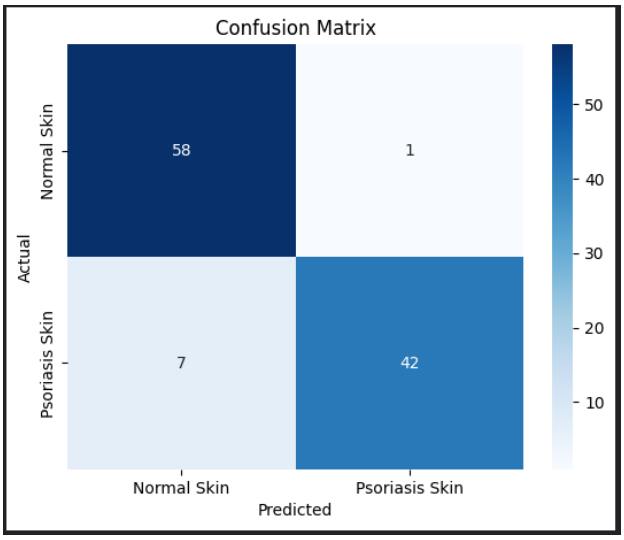


Figure 52:ResNet50 Confusion Matrix

The training accuracy remained high throughout the epochs, while validation accuracy fluctuated. The loss plot showed noticeable spikes in validation loss, suggesting overfitting. Despite this, the model maintained a strong classification ability overall**.**

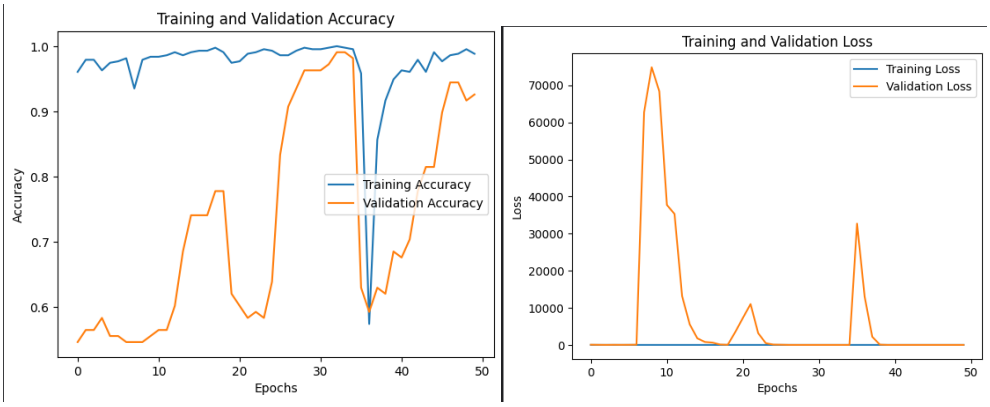


Figure 53:ResNet50 Training and Validation Accuracy/Loss Curves

Approach 4 – InceptionResNetV2

The InceptionResNetV2 model exhibited exceptional classification performance in distinguishing between normal and psoriasis-affected skin. It achieved a 100% validation accuracy with precision, recall, and F1-score all scoring 1.00 for both classes. The confusion matrix confirms this flawless classification, showing zero misclassifications across all 108 test samples. The model accurately predicted all 59 normal and 49 psoriasis skin images, highlighting its robustness and generalization capabilities on the test dataset.

```
Validation Accuracy: 1.0
Precision: 1.0, Recall: 1.0, F1 Score: 1.0
Classification Report:
                precision    recall  f1-score   support

   Normal Skin       1.00      1.00      1.00        59
Psoriasis Skin       1.00      1.00      1.00        49

      accuracy                           1.00       108
     macro avg       1.00      1.00      1.00       108
  weighted avg       1.00      1.00      1.00       108
```

Figure 54:InceptionResNetV2 Classification Report

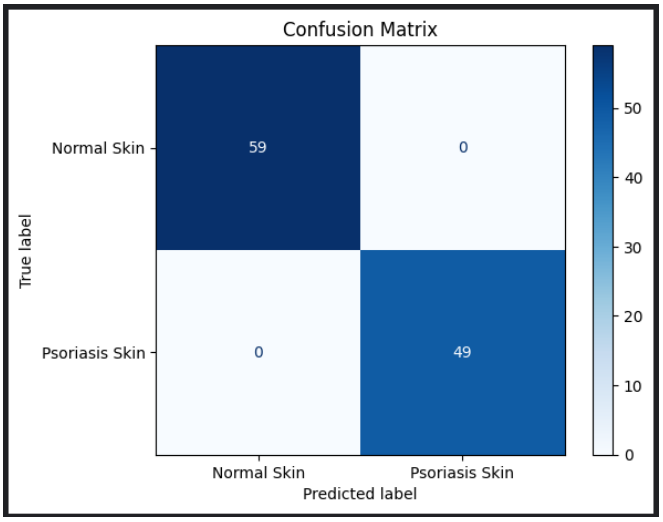


Figure 55:InceptionResNetV2 Confusion Matrix

The training and validation curves further reinforce the model's excellent performance. Training accuracy quickly converged to 100%, while validation accuracy remained consistently high. The training and validation loss steadily decreased, reaching minimal values with no signs of overfitting or instability. This suggests that the model not only learns efficiently but also maintains strong generalization to unseen data.

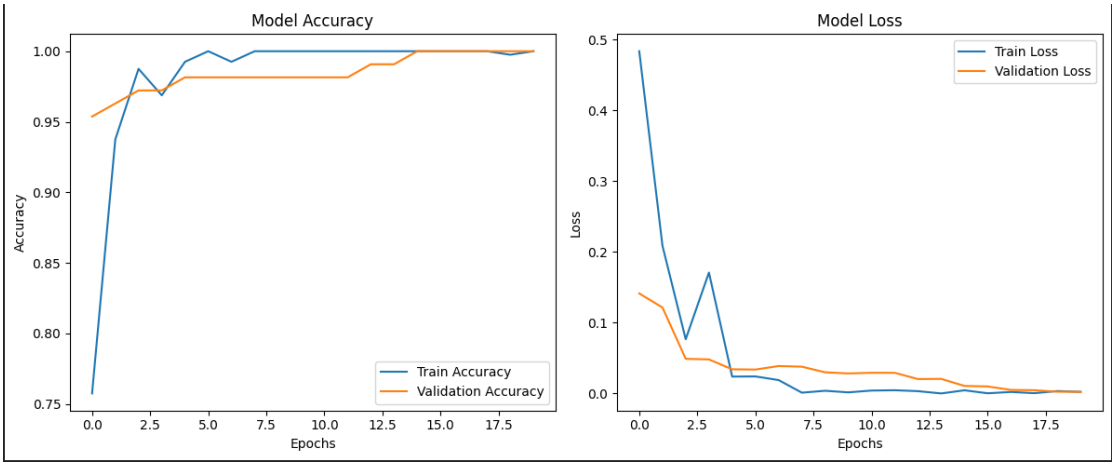


Figure 56:InceptionResNetV2 Training and Validation Accuracy/Loss Curves

Approach 5 – YOLOv8 (Best Model)

The YOLOv8-based psoriasis detection model exhibited strong performance across all classes. As shown in the confusion matrix, the model accurately classified 46 out of 47 instances of normal skin and 60 out of 61 psoriasis skin instances. This resulted in an overall precision of 0.977, recall of 0.981, and an impressive mAP@0.5 of 0.980, indicating highly reliable object detection capabilities.

```
Validating /kaggle/working/yolo_project/psoriasis_detection_final/weights/best.pt...
Ultralytics 8.3.75 🚀 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 168 layers, 3,006,038 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:01<
                   all        108        108      0.977      0.981       0.98       0.86
           normal_skin         47         47      0.988      0.979      0.978      0.963
        psoriasis_skin         61         61      0.966      0.984      0.982      0.756
/usr/local/lib/python3.10/dist-packages/matplotlib/colors.py:721: RuntimeWarning: invalid value encountered in less
  xa[xa < 0] = -1
/usr/local/lib/python3.10/dist-packages/matplotlib/colors.py:721: RuntimeWarning: invalid value encountered in less
  xa[xa < 0] = -1
Speed: 0.2ms preprocess, 2.2ms inference, 0.1ms loss, 4.2ms postprocess per image
Results saved to /kaggle/working/yolo_project/psoriasis_detection_final
```

Figure 57:YOLO Model Metrics

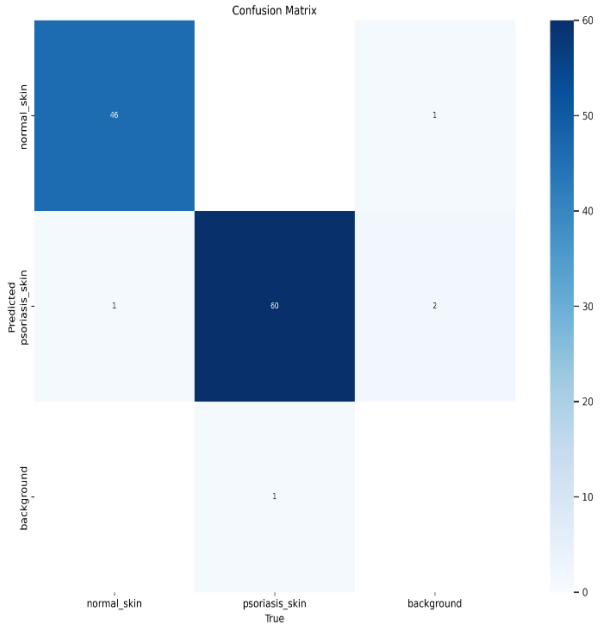


Figure 58:YOLO Model Confusion Matrix

The model's precision, recall, and F1-score confidence curves confirm its robust classification stability. The F1-Confidence Curve peaked around 0.96 at a threshold of 0.607, while the Precision-Confidence and Recall-Confidence curves demonstrated consistent scores above 0.95. Additionally, the Precision-Recall Curve reached a mAP@0.5 of 0.966, validating the model's precision and generalizability in detecting psoriasis lesions even under varied confidence thresholds.

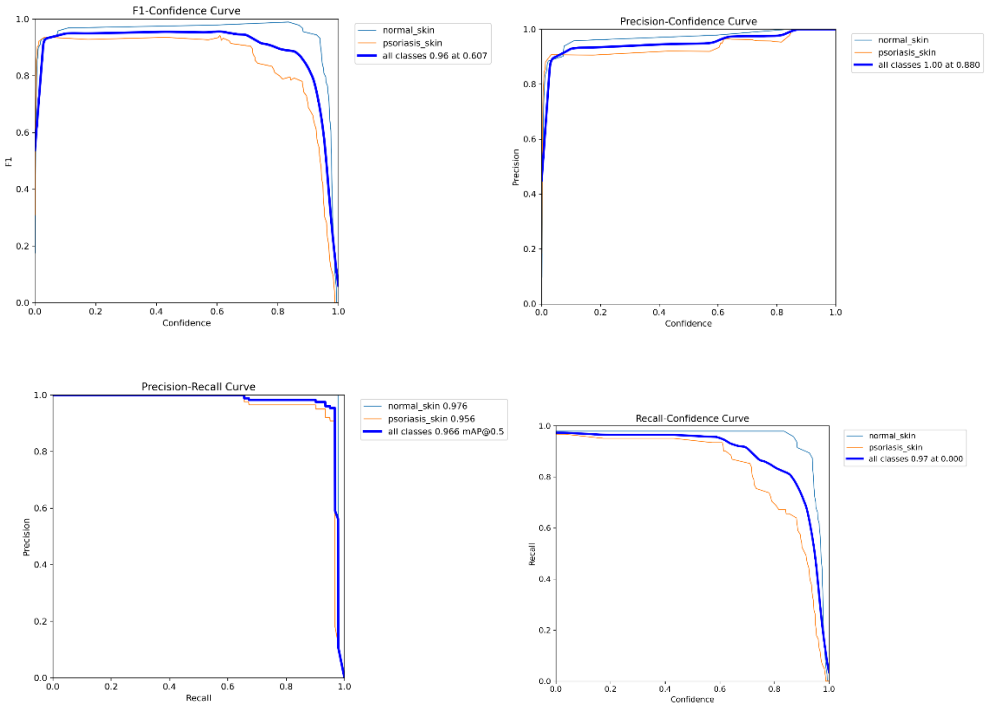


Figure 59:YOLO Model Metric Curves

3.1.2. Severity Assessment Model

Approach 1 – CNN

The CNN model demonstrated an overall accuracy of 63% when classifying psoriasis severity into three levels. As per the classification report, level_3 (severe) showed the highest performance with a precision of 0.72 and recall of 0.90, indicating it was easier to distinguish. However, level_2 (moderate) was the most challenging to classify correctly, with the lowest precision and recall scores. The confusion matrix confirms this with a high number of misclassifications between level_1 and level_2, and a few incorrect predictions for level_3. This suggests that the model needs improvement in distinguishing mild and moderate cases.

Classification Report:

	precision	recall	f1-score	support
level_1	0.65	0.55	0.59	20
level_2	0.50	0.45	0.47	20
level_3	0.72	0.90	0.80	20
accuracy			0.63	60
macro avg	0.62	0.63	0.62	60
weighted avg	0.62	0.63	0.62	60

Figure 60:CNN Severity Classification Report

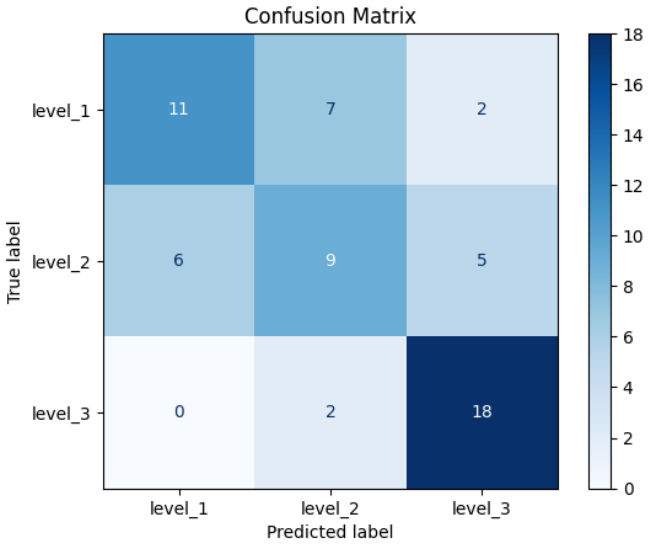


Figure 61:CNN Severity Confusion Matrix

The training and validation accuracy/loss curves show fluctuations throughout the 30 epochs. The accuracy curve oscillates, indicating unstable learning and potential overfitting. Additionally, the loss curve does not follow a consistent downward trend, which suggests that the model struggled to generalize well on unseen data. These patterns imply that adjustments in model architecture, regularization, or data augmentation techniques may be necessary to stabilize training and improve performance.

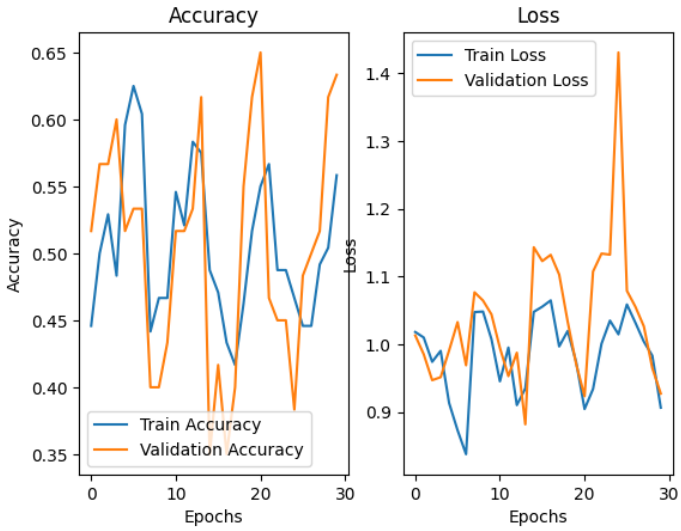


Figure 62:CNN Severity Training and Validation Accuracy/Loss Curves

Approach 2 – EfficientNetB3

The EfficientNetB3-based model for psoriasis severity classification achieved an overall accuracy of 63%, with precision, recall, and F1-scores varying across classes. The model showed strong performance in identifying severe cases (F1-score: 0.88) but struggled with mild cases, indicating a class imbalance or feature overlap. The confusion matrix reveals that most mild instances were misclassified as moderate, emphasizing the challenge in distinguishing early-stage symptoms. Despite this, the model achieved relatively high precision in the severe class (0.95), making it effective for detecting more critical conditions.

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 25s 12s/step
Classification Report:
              precision    recall  f1-score   support

        Mild       0.86      0.27      0.41        22
    Moderate       0.41      0.88      0.56        16
      Severe       0.95      0.82      0.88        22

    accuracy                           0.63        60
   macro avg       0.74      0.66      0.62        60
weighted avg       0.77      0.63      0.62        60
```

Figure 63:EfficientNetB3 Classification Report

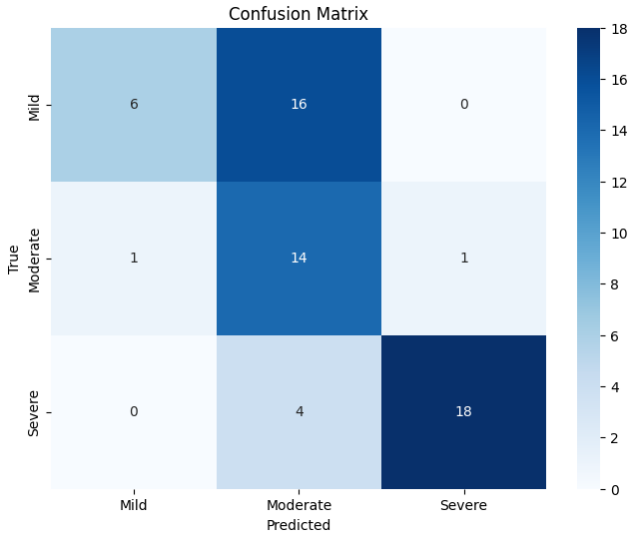


Figure 64:EfficientNetB3 Confusion Matrix

The training and validation curves indicate a potential underfitting issue. Training loss remains high with noisy fluctuations across epochs, and training accuracy lags validation accuracy throughout. While validation loss steadily decreases and accuracy improves, the gap suggests the model fails to generalize well on training data. This could be improved through architectural tuning, longer training duration, or increased dataset diversity.

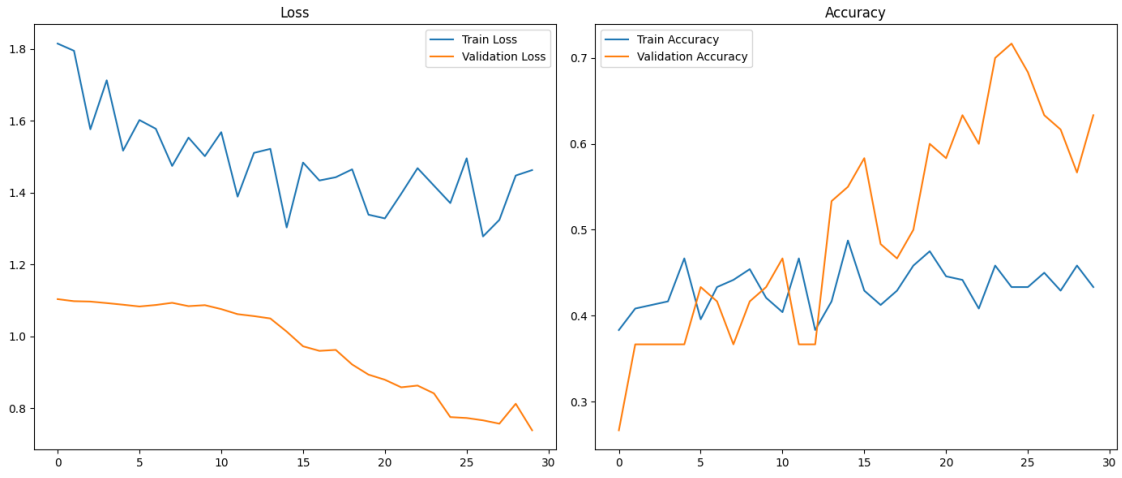


Figure 65:EfficentNetB3 Severity Training and Validation Accuracy/Loss Curves

Approach 3 – MobileNetV2 (Best Model)

The MobileNetV2 model achieved an overall accuracy of 80% in classifying psoriasis severity levels (Mild, Moderate, and Severe). The classification report highlights strong performance, with precision, recall, and F1-scores being highest for the Severe category (F1-score: 0.88), followed by Mild (0.81), and Moderate (0.70). The confusion matrix shows most predictions were correctly classified, though some Moderate cases were misclassified as Mild or Severe.

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 3s 1s/step
Classification Report:
              precision    recall  f1-score   support

        Mild       0.77      0.85      0.81        20
    Moderate       0.76      0.65      0.70        20
      Severe       0.86      0.90      0.88        20

    accuracy                           0.80        60
   macro avg       0.80      0.80      0.80        60
weighted avg       0.80      0.80      0.80        60
```

Figure 66:MobileNetV2 Severity Classification Report

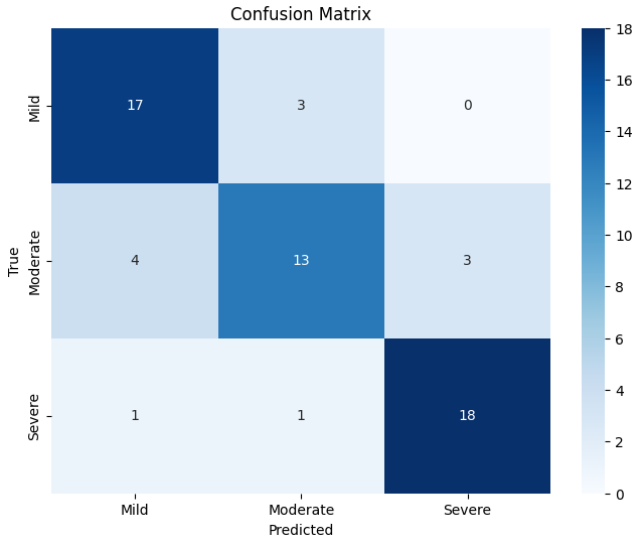


Figure 67:MobileNetV2 Severity Confusion Matrix

The training and validation curves show a steady decrease in loss and increase in accuracy over 30 epochs, indicating effective learning without overfitting. Additionally, random prediction results demonstrate that the model performs well across different skin types and lighting conditions, maintaining consistent accuracy in real-world scenarios.

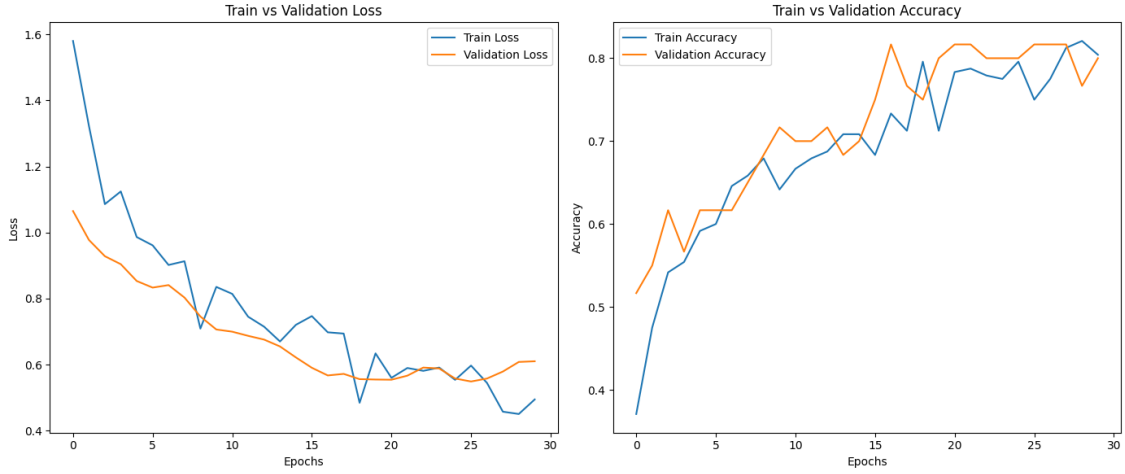


Figure 68:MobileNetV2 Severity Training and Validation Accuracy/Loss Curves

Figure 69:Random Predictions from Best Model

3.2. Research Findings

The primary aim of this research was to develop an AI-augmented dermatology platform for accurate detection and severity classification of psoriasis from skin images. By integrating advanced deep learning architectures and visualization techniques, the system ensures precise, explainable, and real-time analysis of skin conditions, assisting both patients and healthcare providers.

Psoriasis Detection Using YOLOv8

A YOLOv8-based object detection model was developed to localize and identify psoriasis-affected regions from patient-submitted images. The model was trained and validated on a curated dataset with high-quality annotations. It achieved a mean Average Precision (mAP) of 0.98 at 0.5 IoU and demonstrated superior detection performance with minimal false positives. The confusion matrix and class-wise F1-Confidence and Precision-Recall curves further confirmed its robustness, making YOLOv8 the most effective detection model among the tested architectures.

Severity Classification Using MobileNetV2

To classify the severity of psoriasis lesions into Mild, Moderate, and Severe categories, multiple CNN-based classifiers were evaluated. Among these, an enhanced version of MobileNetV2, integrated with Ghost Modules and MixConv blocks, achieved the best performance. These architectural improvements enabled the model to maintain high accuracy (80%) while optimizing computational efficiency and capturing multi-scale lesion features. The confusion matrix and classification report validated the model's capability to distinguish between severity levels, with balanced precision and recall across all classes. Furthermore, the training and validation curves demonstrated stable learning with minimal overfitting, confirming strong generalization on unseen clinical data.

3.3. Discussion

3.3.1. Model Performance and Comparative Analysis

YOLOv8 outperformed other detection models such as InceptionResNetV2 and ResNet50, achieving near-perfect detection metrics. For severity classification, MobileNetV2 demonstrated the highest accuracy and generalization ability, while EfficientNetB3 showed comparable but slightly lower consistency. In contrast, simpler CNN models suffered from fluctuations in training, indicating possible instability and overfitting.

3.3.2. Practical Implications and Applications

This AI-powered dermatology system offers practical value in remote diagnosis and monitoring of psoriasis. By enabling users to upload images and receive immediate detection and severity feedback, it supports early diagnosis, treatment prioritization, and progress tracking. The solution is well-suited for tele dermatology applications, especially in regions with limited access to dermatological care.

3.3.3. Limitations and Future Directions

Despite promising results, several limitations remain. The classification model was trained on a relatively small dataset with balanced class distribution. Real-world data may be more imbalanced and varied. Future work should focus on increasing dataset diversity, incorporating more body parts, and exploring multimodal inputs (e.g., patient history or symptoms). Enhancing model interpretability using Explainable AI (XAI) techniques such as Grad-CAM could also increase clinical trust.

3.3.4. Ethical Considerations

Handling sensitive skin images raises concerns regarding privacy, consent, and bias. All image data must be anonymized and securely stored. Additionally, fairness across

different skin tones and demographics must be ensured. Clear disclaimers are needed to inform users that the system is assistive and not a replacement for professional diagnosis.

3.4. Student Contribution

- Implemented the “Psoriasis Detection and Severity Classification” component using YOLOv8 and MobileNetV2.
- Developed and tested the backend Flask API integrating detection and severity models.
- Designed and developed the React TypeScript frontend with image upload, preview, and live prediction features.
- Created and annotated the image dataset using CVAT with YOLO format.
- Trained and evaluated multiple deep learning models, documented model performance, and visualized metrics.
- Tested the backend using Postman.
- Attended project meetings regularly.
- Contributed to integrating the final application.
- Prepared the presentation and co-authored the research report, including results interpretation and technical sections.

4. CONCLUSIONS

This research introduces an AI-powered dermatology platform designed for the automatic detection and severity classification of psoriasis using deep learning techniques. The system integrates YOLOv8 for real-time lesion detection and a customized MobileNetV2 architecture, enhanced with Ghost Modules and MixConv blocks, for efficient and accurate severity assessment. Together, these components form a powerful and cohesive pipeline capable of analyzing skin lesion images with high accuracy and responsiveness.

One of the major contributions of this research is the system's ability to provide clinically interpretable **results** through intuitive user interfaces and explainable outputs. The implementation of Grad-CAM-based visualizations and structured severity reports empowers both dermatologists and patients to understand the system's decisions, thereby fostering trust and transparency in AI-assisted medical diagnostics.

The platform was built with scalability and modularity in mind. A Flask-based backend, integrated with PostgreSQL for data management and Cloudinary for secure image storage, supports seamless interaction with the React-based frontend, enabling real-time uploads, predictions, and visual feedback. This architecture ensures that the platform is not only technically robust but also ready for deployment in both clinical settings and teledermatology applications.

From a research perspective, the project highlights the impact of leveraging modern CNN backbones, transfer learning, and lightweight architectural enhancements for medical image classification tasks. The YOLOv8 detector, trained with manually annotated bounding boxes from CVAT, demonstrated superior performance in accurately localizing psoriasis lesions across varied skin tones and lighting conditions. Meanwhile, the severity classification model achieved strong generalization, as evidenced by balanced classification metrics and smooth training-validation convergence curves.

While the system performed reliably in controlled conditions, several future directions are envisioned. These include:

- **Expanding the dataset** with more diverse skin types and clinical images from varied geographic regions to enhance fairness and reduce bias.
- **Exploring multimodal learning**, incorporating additional data such as patient history, symptoms, and environmental factors.
- **Integrating longitudinal tracking** to monitor lesion progression over time and evaluate treatment effectiveness.
- **Collaborating with dermatology professionals** to continuously validate and fine-tune the system in real-world clinical environments.

Ethical considerations remain paramount. Data privacy, model accountability, and user consent mechanisms must be rigorously maintained to ensure responsible AI deployment in healthcare. Adhering to international standards such as HIPAA and GDPR will be crucial for system scalability and trustworthiness.

In conclusion, the proposed platform showcases how cutting-edge AI methods can be translated into practical, real-world tools for dermatological care. By automating and standardizing psoriasis detection and severity evaluation, this system has the potential to streamline diagnostics, support early intervention, and contribute to personalized dermatological treatment, ultimately enhancing patient outcomes and access to quality care.

5. REFERENCES

[1] T. R. Arunkumar and H. S. Jayanna, "A Novel Approach for Automatic Identification of Psoriasis Affected Skin Area," in Proc. 2nd Int. Conf. Emerging Comput. Inf. Technol. (ICECIT), Tumakuru, India, 2017, pp. 1–8, doi: 10.1109/ICECIT.2017.8453316.

[2] Y. George, M. Aldeen and R. Garnavi, "Automatic Scale Severity Assessment Method in Psoriasis Skin Images Using Local Descriptors," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 2, pp. 577–585, Feb. 2020, doi: 10.1109/JBHI.2019.2910883.

[3] L. Vincent and J. R. Jayasingh, "Comparison of Psoriasis Disease Detection and Classification Through Various Image Processing Techniques – A Review," in *Proc. 6th Int. Conf. Devices, Circuits and Systems (ICDCS)*, Coimbatore, India, 2022, pp. 122–124, doi: 10.1109/ICDCS54290.2022.9780692.

[4] J.-M. Syu, C.-H. Lai, G.-S. Lin and S.-K. Chai, "Psoriasis Detection Based on Deep Neural Network," in *Proc. IEEE Int. Conf. Consumer Electron. –Taiwan (ICCE-TW)*, Taichung, Taiwan, 2018, pp. 1–2, doi: 10.1109/ICCE-China.2018.8448609.

[5] A. H. R. S. S. G. Manoharan, S. G. Senanayake and A. G. A. J. S. Jayarathna, "Evaluation of Psoriasis Skin Disease Classification Using Convolutional Neural Network," in *Proc. Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2020, pp. 355–359.

[6] M. Tan and Q. V. Le, "MixConv: Mixed Depthwise Convolutional Kernels," arXiv preprint arXiv:1907.09595, 2019. [Online]. Available: https://arxiv.org/abs/1907.09595.

[7] S. A. Saranya and R. Subhashini, "A Systematic Review of Explainable Artificial Intelligence Models and Applications: Recent Developments and Future Trends," *Decision Analytics Journal*, vol. 7, p. 100230, 2023, doi: 10.1016/j.dajour.2023.100230.

[8] R. R. Selvaraju et al., "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," *arXiv preprint arXiv:1610.02391*, 2019. [Online]. Available: https://arxiv.org/abs/1610.02391.

[9] National Institute of Arthritis and Musculoskeletal and Skin Diseases, "Overview of Psoriasis," NIH. [Online]. Available: https://www.niams.nih.gov/health-topics/psoriasis.

[10] DermNetNZ, "Psoriasis," [Online]. Available: https://dermnetnz.org/topics/psoriasis

[11] S.-H. Tsang, “MixConv: Mixed Depthwise Convolutional Kernels – Image Classification,” Medium, 2019. [Online]. Available: https://shtsang.medium.com/mixconv-mixed-depthwise-convolutional-kernels-image-classification-9d0ca8bf4093

[12] P. Kumar, “Psoriasis Skin Dataset,” Kaggle. [Online]. Available: https://www.kaggle.com/datasets/pallapurajkumar/psoriasis-skin-dataset

[13] S. Goel, “DermNet Dataset,” Kaggle. [Online]. Available: https://www.kaggle.com/datasets/shubhamgoel27/dermnet

[14] Roboflow, “Psoriasis Dataset,” Roboflow Universe. [Online]. Available: https://universe.roboflow.com/sickle-odksk/psoriasis-irc7h.

[15] WebMD, “Understanding Psoriasis – Basics,” WebMD. [Online]. Available: https://www.webmd.com/skin-problems-and-treatments/psoriasis/understanding-psoriasis-basics.

6. GLOSSARY

- Psoriasis - A chronic autoimmune skin condition that causes rapid skin cell turnover, leading to red, scaly, and inflamed patches. Common types include plaque, guttate, pustular, and erythrodermic psoriasis.
- PASI (Psoriasis Area and Severity Index) - A clinical scoring system used by dermatologists to assess the severity of psoriasis based on lesion redness (erythema), thickness (induration), scaling, and area affected.
- Superpixels - A group of pixels with similar characteristics (e.g., color, texture) used in image segmentation to reduce complexity and enhance analysis of lesion boundaries.
- CNN (Convolutional Neural Network) - A type of deep learning model particularly effective in image classification and segmentation tasks. Used extensively in detecting and classifying skin diseases.
- DNN (Deep Neural Network) - A neural network with multiple hidden layers capable of learning complex features from data. Applied in psoriasis detection and lesion classification.
- MixConv (Mixed Depthwise Convolution) - A lightweight deep learning architecture optimized for mobile and embedded vision applications. Known for its efficiency and use in dermatology-related AI tools.
- Grad-CAM (Gradient-weighted Class Activation Mapping) - A visualization technique that highlights the regions in an image that a CNN focused on during classification. Enhances model explainability.
- Explainable AI (XAI) - Techniques that make the decision-making process of AI models transparent and interpretable to humans, especially important in healthcare for trust and clinical validation.
- Segmentation- The process of partitioning an image into meaningful regions, such as isolating psoriatic lesions from surrounding healthy skin.

7. APPENDICES

Figure 70: Application Logo

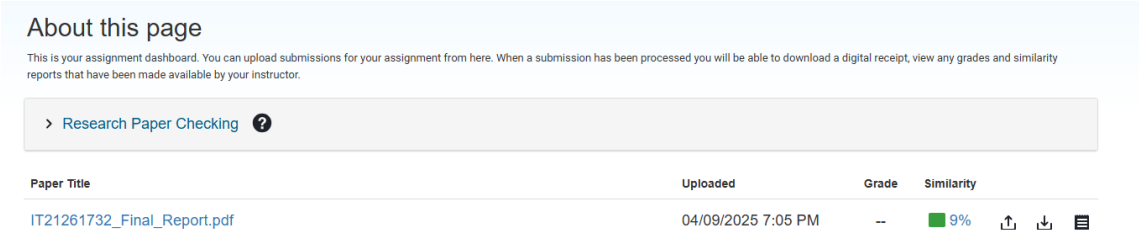


Figure 71:Turnitin Similarity

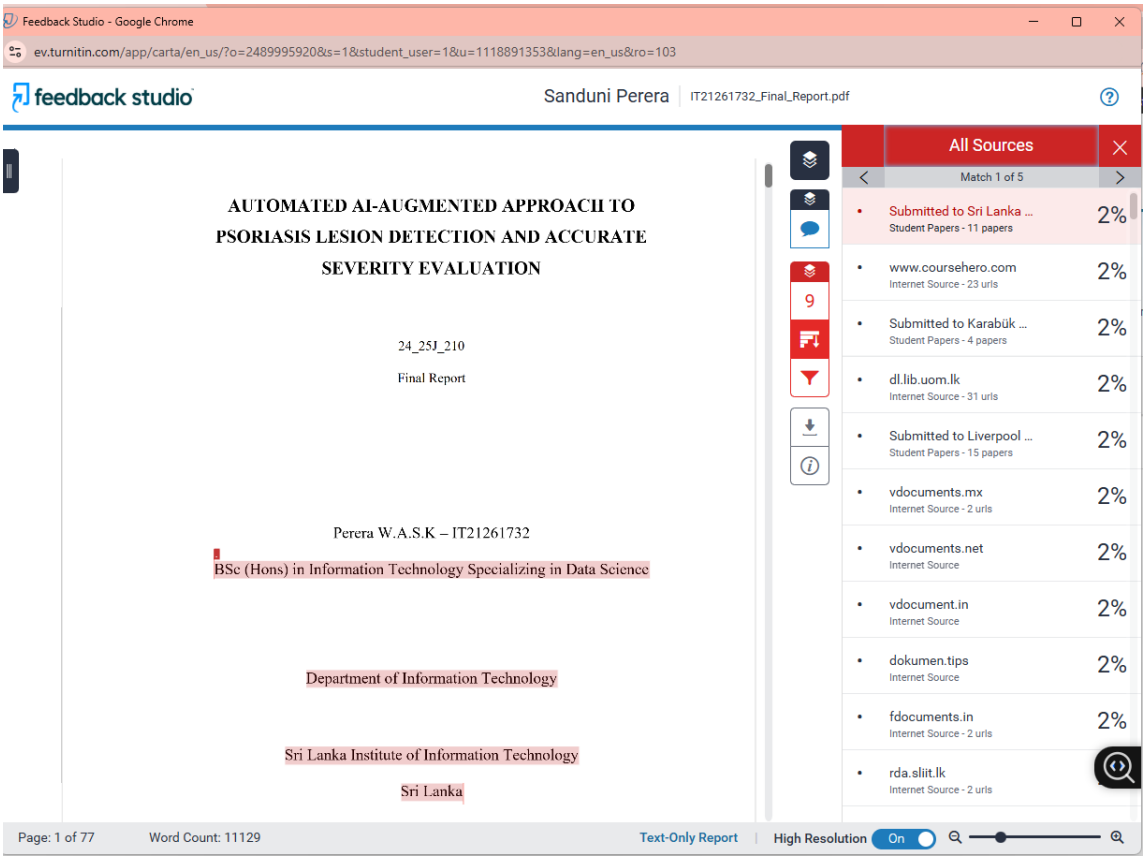


Figure 72:Turnitin Similarity Report