DERMASCOPEAI : A DEEP LEARNING APPROACH FOR SKIN DISEASE DIAGNOSIS AND SEVERITY ASSESSMENT

24-25J-210

Final Report

Gimmana M. R. M		IT21176456
Lakshani D. M. W. S		IT21262104
Perera W. A. S. K		IT21271732

BSc (Hons) in Information Technology Specializing in Data Science

Department of Computer Science

Sri Lanka Institute of Information Technology

Sri Lanka

April 2025

DERMASCOPEAI : A DEEP LEARNING APPROACH FOR SKIN DISEASE DIAGNOSIS AND SEVERITY ASSESSMENT

24-25J-210

Final Report

Gimmana M. R. M		IT21176456
Lakshani D. M. W. S		IT21262104
Perera W. A. S. K		IT21271732

BSc (Hons) in Information Technology Specializing in Data Science

Department of Computer Science

Sri Lanka Institute of Information Technology

Sri Lanka

April 2025

DECLARATION

We declare that this is our own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Gimmana M.R.M	IT21176456	
Lakshani D.M.W.S	IT21262104	
Perera W.A.S.K	IT21261732	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor: _____ Date: 2025 - 04 -11

(Mr. Samadhi Rathnayake)

Signature of the co-supervisor: _____ Date: _____

(Mr. Junius Anjana)

ABSTRACT

Skin diseases such as acne, eczema, and psoriasis are frequent chronic illnesses that significantly influence patients' physical and mental well-being. The traditional diagnostic approach relies on subjective clinical assessments, leading to inconsistency in diagnosis and treatment results. To address this challenge, DermaScope AI is suggested as an Artificial Intelligence (AI) system to accurately identify and assess the severity of these three dermatoses using advanced image processing and Machine Learning (ML) techniques. With deep learning and Explainable AI (XAI), DermaScope AI interprets clinical images, delineates affected areas, and provides standardized severity scores. The system enhances the accuracy of diagnosis, reduces subjectivity, and helps clinicians deliver timely and individualized treatment plans. Real-time data-driven analytics allow for effective and consistent evaluation, thereby ultimately improving patient care and expanding access to dermatology knowledge. Potential future clinical integration and validation of DermaScope AI can revolutionize dermatological diagnostics, with a viable and scalable option for the improvement of skin disease care across populations.

Keywords – acne, atopic dermatitis, deep learning, eczema, explainable AI, grad-cam, lesion detection, machine learning, psoriasis, severity.

ACKNOWLEDGMENT

We would like to extend our heartfelt gratitude to all those who contributed to the successful completion of this research. Foremost, we thank our academic supervisors, Mr. Samadhi Rathnayake and Dr. Junius Anjana, whose untiring efforts provided us with valuable guidance, insightful feedback, and much-needed encouragement at every step in the research process. Their expertise and thought-provoking suggestions shaped the direction and were instrumental in the primary development of this work. Also, we take this opportunity to specially thank our external supervisors, Dr. Saman Gunasekara and Dr. Kasun Nayanakantha for all the support they provided, including their information sharing at a very early time which crucially supported for initiating our research. And we would also like to send our thanks to the evaluation panel who have been so kind throughout our project. We are further profoundly indebted to the faculty and staff at SLIIT, who initially provided us with the grounds to conduct this research through support and resources. The conducive departmental atmosphere was, in fact, a steady source of inspiration and encouragement. We are deeply appreciative of the critical views and input shared by our co-researchers and fellow colleagues during the development process. Indeed, their collaborative approach in knowledge sharing significantly enhanced both the quality and rigor of this research. We would also like to extend our deepest appreciation to our families and friends, who supported us throughout this work. The realization of this research would not have been possible without the collective effort of all. Truly, we are very grateful for all the contributions and support that made it possible.

TABLE OF CONTENTS

DECLARATION iii
ABSTRACT iv
ACKNOWLEDGMENT v
TABLE OF CONTENTS vi
LIST OF FIGURES ix
LIST OF TABLES xiii
LIST OF ABBREVIATIONS xiv
1. INTRODUCTION 1
1.1 Background 1
1.2 Background Literature 3
1.3 Research Gap 10
1.4 Research Problem 14
1.5 Research Objective 15
1.5.1 Main Objective 15
1.5.2 Sub Objectives 16
2. METHODOLOGY 18
2.1 Requirement Gathering 18
2.1.1 Stakeholder Identification 18
2.1.2 Methods Used 19
2.1.3 Functional Requirements 19
2.1.4 Non-Functional Requirements 20
2.1.5 Personal Requirements 20
2.2 Data Collection 21
2.3 Data Exploration 23
2.3.1 Acne 23
2.3.2 Eczema 26
2.3.3 Psoriasis 30
2.4 Data Preprocessing 34

2.4.1 Acne 34
2.4.2 Eczema 36
2.4.3 Psoriasis 38
2.5 Model Implementation 39
2.5.1 Acne Detection Model Architecture 39
2.5.2 Acne Severity Classification Model Architecture 50
2.5.3 Eczema Detection Model Architecture 55
2.5.4 Eczema Severity Classification Model Architecture 59
2.5.5 Psoriasis Detection Model Architecture 65
2.5.6 Psoriasis Severity Classification Model Architecture 69
2.6 Commercialization Aspects of the Product 73
2.6.1 Market Potential 73
2.6.2 Role of Skin Disease Detection and Severity Assessment 74
2.6.3 Commercialization Strategy 74
2.6.4 Competitive Advantage 75
2.6.5 Scalability and Future Development 75
2.7 Testing and Implementation 76
2.7.1 Functional Requirements 76
2.7.2 Non-Functional Requirements 78
2.7.3 Backend Implementation 79
2.7.4 Frontend Implementation 82
2.8 System Diagram 86
2.8.1.1 Acne Component Diagram 87
2.8.1.2 Eczema Component Diagram 87
2.8.1.3 Psoriasis Component Diagram 88
3. RESULTS AND DISCUSSIONS 89
3.1 Acne 89
3.1.1 Acne Detection 89
3.1.2 Acne Severity Detection 99
3.2 Eczema 106

3.2.1 Eczema Detection 106
3.2.2 Eczema Severity Classification 112
3.3 Psoriasis 117
3.3.1 Psoriasis Detection 117
3.3.2 Psoriasis Severity Classification 125
4. SUMMARY OF EACH STUDENT'S CONTRIBUTION 130
5. CONCLUSION 131
6. REFERENCES 132
7. GLOSSARY 136
8. APPENDICES 139

LIST OF FIGURES

Figure 1: Sample images from the acne and non-acne classification dataset 21
Figure 2: Sample images from the eczema and non-eczema classification dataset. 22
Figure 3: Sample images from the Psoriasis classification dataset 22
Figure 4: Acne Dataset 23
Figure 5: Color distribution 24
Figure 6: Acne Severity Dataset 24
Figure 7: Color Distribution of Acne Severity Detection 25
Figure 8: Visualization showing the balanced distribution between normal skin and eczema skin images 26
Figure 9: Graph representing the mean pixel intensity for the red, green, and blue channels in the dataset 27
Figure 10: Visualization showing the distribution of eczema severity levels categorized as mild, moderate, and severe 28
Figure 11: Graphs depicting the mean pixel intensity distributions for mild, moderate, and severe eczema images 29
Figure 12: Visualization showing the distribution between normal skin and Psoriasis skin images 30
Figure 13: Graphs depicting the mean pixel intensity distributions for normal skin and Psoriasis skin images 31
Figure 14: Visualization showing the distribution of Psoriasis severity levels categorized as mild, moderate, and severe 32
Figure 15: Graphs depicting the mean pixel intensity distributions for mild, moderate, and severe Psoriasis image 33
Figure 16: CNN Model Architecture p 1 41
Figure 17: CNN Architecture p 2 41
Figure 18: Training Progress and Metrics 42
Figure 19: VGG16 - Based Model Architecture 43
Figure 20: VGG19 Based model architecture for acne classification 45
Figure 21: Resnet50 Model architecture 46
Figure 22: EfficientNetB0 Model architecture 48
Figure 23: Siamese Network 50
Figure 24: EfficientNetB0 Model architecture for acne severity 50
Figure 25: CNN Model architecture for acne severity 51
Figure 26: VGG16 Model architecture for acne severity 51
Figure 27: ResNet50V2 Model architecture for acne severity 52
Figure 28: Function to create pairs of images and labels 53
Figure 29: Base network with pre-trained ResNet50 53
Figure 30: Siamese Network 54
Figure 31: Multi classification model 54

Figure 32: Implementation of the Squeeze-and-Excitation (SE) block to reweight important feature channels 56
Figure 33: Building a MobileNetV2-based model with integrated SE attention block for improved skin condition focus 56
Figure 34: Model training with class weights and early stopping, learning rate scheduling, and checkpointing 57
Figure 35: Custom CNN model architecture with SE attention blocks embedded after each convolutional layer block 58
Figure 36: Implementation of the attention mechanism as a custom layer 59
Figure 37: ResNet50 model with custom attention mechanism and fine-tuning for eczema detection 59
Figure 38: Implementation of Focal Loss to enhance model sensitivity towards minority classes by applying dynamic scaling to hard-to-classify samples..... 60
Figure 39: Calculating class weights to assign higher importance to underrepresented classes and promote balanced learning during training 61
Figure 40: Implementation of the Squeeze-and-Excitation (SE) block to enhance feature relevance through channel-wise recalibration 61
Figure 41: Building the EfficientNetB0 model integrated with SE block and additional dense layers for classification 62
Figure 42: Building a custom CNN model with Residual Blocks, SE blocks, and fully connected layers for classification 63
Figure 43: Implementation of the CBAM block combining channel and spatial attention mechanisms to enhance feature relevance 64
Figure 44: Building the ResNet50-based model with integrated CBAM block and custom dense layers 65
Figure 45: Train the YOLO Model 66
Figure 46: Custom CNN architecture for classifying psoriasis vs. normal skin..... 67
Figure 47: EfficientNetV2B0 model architecture for binary classification of normal and psoriasis affected skin 67
Figure 48: ResNet50-based model for binary skin classification of normal and psoriasis-affected skin 68
Figure 49: InceptionResNetV2 model for binary skin classification of normal and psoriasis-affected skin 69
Figure 50: GhostModule implementation 69
Figure 51: MixConv block implementation 70
Figure 52: Full MobileNetV2 model with integrated blocks 70
Figure 53: Baseline CNN model used for psoriasis severity classification 71
Figure 54: Customized EfficientNetB3-based model architecture for psoriasis severity classification 72

Figure 55: Flask app setup with CORS and PostgreSQL connection pool initialization 79
Figure 56: Model loading setup for acne, eczema, and psoriasis detection and severity classification using TensorFlow and YOLOv8 79
Figure 57: Uploading images to secure cloud storage using the Cloudinary API integration 80
Figure 58: API for acne, eczema, and psoriasis detection 80
Figure 59: API for acne, eczema, and psoriasis severity assessment 81
Figure 60: React frontend fetching data from Flask backend using dynamic API integration 82
Figure 61: Image Uploading Interface 82
Figure 62: Response view with detection outputs 83
Figure 63: Severity interpretation card with confidence levels 83
Figure 64: Doctor dashboard 84
Figure 65: Patient history and progress tracking page 84
Figure 66: Dashboard Page 85
Figure 67: Home Page 85
Figure 68: Overall System Architecture 86
Figure 69: Acne Component Diagram 87
Figure 70: Eczema Component Diagram 87
Figure 71: Psoriasis Component Diagram 88
Figure 72: Confusion Matrix of CNN 90
Figure 73: Accuracy and loss graphs of CNN 90
Figure 74: Confusion Matrix of VGG16 91
Figure 75: Accuracy and Loss graphs of VGG16 92
Figure 76: Confusion Matrix of VGG19 93
Figure 77: Accuracy and Loss graphs of VGG19 93
Figure 78: Confusion Matrix ResNet 95
Figure 79: Accuracy and Loss graphs of ResNet 95
Figure 80: Confusion Matrix of EfficientNet 97
Figure 81: Accuracy and Loss graphs of EfficientNet 97
Figure 82: Confusion Matrix of Siamese neural Networks 98
Figure 83: Accuracy and Loss graphs of Siamese Neural Networks 99
Figure 84: Accuracy and Loss graphs of EfficientNet 100
Figure 85: Confusion Matrix of CNN 102
Figure 86: Accuracy and Loss graphs of CNN 102
Figure 87: Confusion Matrix of VGG16 104
Figure 88: Accuracy and Loss graphs of VGG16 104
Figure 89: Confusion Matrix of ViT 106
Figure 90: Accuracy and Loss graphs of ViT 106

Figure 91: Classification report and the confusion matrix for MobileNetV2..... 107
Figure 92: Training and validation accuracy/loss visualization for MobileNetV2 108
Figure 93: Confusion matrix and the classification report for CNN..... 109
Figure 94: Training and validation accuracy/loss visualization for CNN.....110
Figure 95: Classification report and confusion matrix for ResNet50.....111
Figure 96: Training and validation accuracy/loss visualization for ResNet50 111
Figure 97: Classification report and confusion matrix for EfficientNetB0.....112
Figure 98: Training and validation accuracy/loss visualization for EfficientNetB0 113
Figure 99: Classification report and confusion matrix for CNN 114
Figure 100:Classification report and confusion matrix for ResNet50.....115
Figure 101: Training and validation accuracy/loss visualization for ResNet50 116
Figure 102: YOLOv8 Model Performance Metrics 117
Figure 103: YOLOv8 Model Confusion Matrix 117
Figure 104: YOLOv8 Model Metric Curves.....118
Figure 105: CNN Classification Report 119
Figure 106: CNN Confusion Matrix.....119
Figure 107: CNN Training and Validation Accuracy/Loss Curves..... 120
Figure 108: EfficientNetB0 Classification Report..... 120
Figure 109: EfficientNetB0 Confusion Matrix 121
Figure 110: EfficienNetB0 Training and Validation Accuracy/Loss Curves..... 121
Figure 111: ResNet50 Classification Report 122
Figure 112: ResNet50 Confusion Matrix..... 122
Figure 113: ResNet50 Training and Validation Accuracy/Loss Curves 123
Figure 114: InceptionResNetV2 Classification Report..... 123
Figure 115: InceptionResNetV2 Confusion Matrix..... 124
Figure 116: InceptionResNetV2 Training and Validation Accuracy/Loss Curves 124
Figure 117: MobileNetV2 Severity Classification Report 125
Figure 118: MobileNetV2 Severity Confusion Matrix 125
Figure 119: MobileNetV2 Severity Training and Validation Accuracy/Loss Curves 126
Figure 120: Random Predictions from Best Model..... 126
Figure 121: CNN Severity Classification Report 127
Figure 122: CNN Severity Confusion Matrix 127
Figure 123: CNN Severity Training and Validation Accuracy/Loss Curves..... 128
Figure 124: EfficientNetB3 Classification Report..... 128
Figure 125: EfficientNetB3 Confusion Matrix 129
Figure 126: EfficentNetB3 Severity Training and Validation Accuracy/Loss Curves 129
Figure 127: Application logo 139

Figure 128: Turnitin Similarity 139
Figure 129:Turnitin Report 139

LIST OF TABLES

Table 1:Comparison of the features addressed by the previous research studies for Acne 11
Table 2:Comparison of the features addressed by the previous research studies for Eczema..... 12
Table 3:Comparison of the features addressed by the previous research studies for Psoriasis..... 13

LIST OF ABBREVIATIONS

Abbreviation	Description
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
AD	Atopic Dermatitis
CNN	Convolutional Neural Network
SE	Squeeze-and-Excitation
CBAM	Convolutional Block Attention Module
GLCM	Gray Level Co-occurrence Matrix
SVM	Support Vector Machine
EASI	Eczema Area and Severity Index
SCORAD	SCORing Atopic Dermatitis
XAI	Explainable Artificial Intelligence
UI	User Interface
UX	User Experience
IDE	Integrated Development Environment
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
REST API	Representative State Transfer Application Programming Interface

1. INTRODUCTION

1.1 Background

Skin diseases are a worldwide health issue impacting people from different backgrounds and age groups. Nevertheless, precise, and timely diagnosis of the disorders presents major difficulties. Proper diagnosis is essential to give proper treatment and enhance the well-being and quality of life of the subjects [7]. Among these, eczema, acne, and psoriasis are three of the most prevalent and impactful dermatological conditions, each contributing to physical discomfort, psychological stress, and reduced quality of life. Effective detection and severity assessment of these conditions play a vital role in ensuring timely and appropriate treatment, ultimately improving patient outcomes. However, current clinical diagnosis and grading methods for these diseases often rely heavily on visual inspection by dermatologists, making the process subjective, time consuming, and susceptible to variability between different clinicians.

Acne is a common skin condition affecting individuals of all ages, though it is particularly prevalent during adolescence. It arises from a combination of excessive sebum production, colonization follicular (primarily hyperkeratinization, Cutibacterium bacterial acnes), and inflammatory responses. Acne presents in various forms, ranging from mild comedonal acne to severe inflammatory lesions that can lead to scarring and psychological distress. Severity grading in acne is crucial for selecting appropriate treatment plans, yet manual grading is subjective and can vary between dermatologists, necessitating automated and objective systems for accurate evaluation.

Similarly, Eczema is primarily represented by atopic dermatitis (AD), which affects both children and adults [9]. It results in itchy, inflamed, and red skin patches, often appearing on the face, neck, hands, and flexural regions such as the elbows and knees. The condition is relapsing and chronic, with flare-ups influenced by genetic predisposition, environmental triggers, and immune system dysregulation.

Accurate diagnosis and severity grading are essential for managing eczema effectively, yet current methods rely heavily on subjective clinical evaluations, leading to inconsistent outcomes.

Psoriasis, another chronic inflammatory skin disorder characterized by abnormal immune activation and accelerated skin cell turnover, resulting in thick, scaly plaques on the skin. Its severity can range from localized mild lesions to widespread severe disease, significantly impacting patients' physical and psychological well-being. Traditionally, psoriasis severity has been assessed using clinical scores such as the PASI (Psoriasis Area and Severity Index), but these assessments require specialized expertise and suffer from inter-observer variability.

With the increasing global prevalence of these conditions and the limited availability of dermatologists, there is a growing need for automated, accurate, and objective tools to detect these diseases and assess their severity consistently. Recent advancements in AI, ML, and Computer Vision have introduced computer-aided diagnostic (CAD) systems capable of analyzing skin images to detect diseases and classify their severity levels with high accuracy. These systems leverage deep learning models for image analysis, lesion segmentation, and feature extraction, enabling faster, more reliable, and standardized assessments compared to traditional methods. The development of such AI-powered solutions not only enhances clinical efficiency but also empowers patients in remote or resource-limited settings to monitor their skin conditions effectively. This research aims to develop a comprehensive AI-based framework for the detection and severity classification of eczema, acne, and psoriasis, offering a scalable and cost-effective approach to modern dermatological care.

1.2 Background Literature

Recent advancements in deep learning and various other advanced machine learning algorithms have paved the way for the development of novel computer-aided diagnosis systems capable of alleviating human subjectivity in diagnosing skin diseases more accurately, determining them into correct categories, and assessing their severity. Utilizing machine learning to study vast collections of skin images helps in recognizing patterns and features indicative of various skin conditions, making it a highly prospective diagnostic tool. Convolutional Neural Networks (CNNs) have emerged as powerful methods for automatic skin lesion classification and severity assessment, offering a more efficient diagnostic approach compared to traditional methods that are slower and prone to variability due to evaluator interpretation. These models prove valuable for clinicians and patients alike, providing quicker and more reliable assessments. Numerous studies have implemented machine learning techniques for diagnosing and assessing the severity of different skin diseases. For example, image processing methods such as Gray Level Co-occurrence Matrix (GLCM) and machine learning models like Support Vector Machines (SVM) have demonstrated considerable accuracy in classifying images into specific types such as melanoma, psoriasis, dermatitis, or acne. Moreover, the application of U-Net architectures for lesion segmentation has proven effective in identifying lesion boundaries, which is essential for accurate severity assessment. This progress holds great promise for the integration of machine learning into routine dermatology practices, enabling personalized and more reliable therapy delivery for a broad spectrum of skin conditions.

Rohit Yadav, Aashika Jain, and Sanjiv Sharma proposed a deep learning-based acne detection system utilizing a ResNet-18 architecture. Their model was trained on a dataset encompassing diverse acne severities and incorporated data augmentation for enhanced robustness. The system categorized acne into four levels (normal, mild, moderate, severe) and achieved a reported accuracy of 90%. While their approach demonstrates promising results, the reliance on a specific dataset limits

the model's generalizability to individuals with varying skin tones. Furthermore, the absence of detailed information regarding acne localization and quantification hinders a comprehensive assessment of the system's capabilities. Future research should focus on expanding the dataset to include diverse skin types, refining acne localization techniques, and exploring the correlation between acne severity scores and clinical assessment.[2]

Naidu, Kareppa, and Menon developed Dermato, a web-based application that employs deep learning for acne subtype classification (comedonal, pustular, rosacea, papules) and severity assessment. Their system processes user-uploaded images using a pipeline incorporating pre-processing, HSV color-based segmentation, and a fine-tuned InceptionV3 model. Dermato achieved an impressive classification accuracy of 99.88%, showcasing its potential to revolutionize acne diagnosis. While the reported accuracy is remarkable, further validation through clinical trials is essential to establish its reliability in real-world settings. Additionally, exploring the generalizability of the model to diverse skin tones and acne presentations would strengthen its clinical utility.[3]

Humayun, Malik, Belhaouri, and Kamel proposed a template matching approach for acne lesion localization using the chi-square test. To enhance accuracy, N-mean kernel filtering was applied as a preprocessing step. The method's performance was evaluated using precision, recall, accuracy, and F1 score. While the chi-square test effectively handles image variations, the authors acknowledge the challenges posed by cluttered backgrounds and noisy lesions, which can impact the method's overall performance.[4]

Ravi, K., Kiran, M. K., & Umadevi, V. provide a comprehensive overview of acne classification research. They highlight the predominant focus on detection and severity assessment, with limited exploration of detailed subtyping. The authors discuss the evolution of techniques from traditional image processing to deep learning-based approaches, emphasizing the challenges posed by imbalanced

datasets and data scarcity. Data augmentation strategies, including both traditional methods and generative models like VAEs, have been explored to address these limitations. However, the optimal application of these techniques for acne classification remains an open research question. The study underscores the importance of careful architecture selection, hyperparameter optimization, and loss function design in achieving superior performance in this domain.[5]

Paluri Krishna Veni and Ashish Gupta introduced a novel hybrid deep learning model for acne severity classification. Their model effectively combines the robust feature extraction capabilities of VGG16 with the attention mechanism of a CBAM module, followed by a CapsNet for accurate classification. This innovative approach surpasses the limitations of traditional methods by achieving superior performance in distinguishing acne levels.[6]

Beyond eczema classification, similar machine learning techniques have been applied to the segmentation and classification of eczema conditions. Nisar et al. [8] worked with supervised machine learning methods for segmenting and categorizing eczema lesions. Supervised machine learning methods can improve the accuracy and efficiency of clinical assessments for identification or severity assessment where they have been developed. Researchers have successfully segmented and categorized eczema skin lesions with a high level of accuracy through the employment of SVM and CNN algorithms. Such techniques are especially useful in clinical situations where conventional methods can result in time constraints and subjectivity. These machine learning models could be a major advance in leading the dermatologic diagnosis to avoid errors which harm patient care as they are capable of reliably identifying multiple types of eczema.

Jardeleza et al. [9] proposed an algorithm combining GLCM and SVM in the detection of several forms of typical eczemas (atopic dermatitis, contact dermatitis and nummular dermatitis). In the study, researchers created a prototype system that

was able to obtain photos of skin and eczema locations as well as benchmarking accurately what kind of type dermatitis patients suffered from. This system, with an accuracy of 83.33%, is very useful in regions where access to highly trained dermatologists would seem difficult as the earlier and more accurate diagnosis could dramatically impact treatment outcomes. In this specific case, the mixture of GLCM and SVM demonstrates that machine learning can provide reliable diagnostic tools also to be easily understandable by everyone.

Junayed et al. [10] successfully introduced EczemaNet, to achieve high accuracy in classification of different types of eczema diseases. CNN has demonstrated an impressive performance in classifying various eczema conditions. Many still call for deep learning models as EczemaNet, a deep CNN-based model specifically developed to handle the difficulty of telling eczemas apart from each other. Trained on a large set of annotated eczema photos, the model was more accurate and provided qualitatively consolidated classification results over previous models. The ability of EczemaNet to make accurate and actionable diagnoses may not only aid clinical decision-making but also suggest the potential adoption of such models into routine dermatological practice, which could lead to notable time and accuracy gains.

Attar et al. [11] explored the use of digital camera images for automatic identification and objective assessment of eczema severity in an automated manner to advance the process of evaluating eczema. Through the utilization of advanced image processing methods, these systems are able to programmatically detect areas where there is eczema and then grade severity scores for each area based on specific clinical criteria. All automated assessments were validated for their accuracy and reliability against the opinions of board-certified dermatologists. Results indicate that the systems can achieve equivalent levels of performance besides saving medical staff time. Automatic systems also ensure the objectiveness and uniformity

of assessments that are crucial for tracking disease progression and tailoring treatment plans.

Nisar et al. [12] successfully applied a modified U-Net network for skin lesion segmentation in eczema within the image. It has shown much better results than prior pixel-based classification. This is useful for medical image segmentation tasks since U-Net was designed to address complex lesion shapes and varying scales sizes. This is particularly valuable for dermatologists who will be able to identify the exact lesion by using those highly precise boundaries of eczema-affected regions drawn with U-Net in this case. This specificity is important to later detection and treatment planning for patients to ensure that an accurate diagnosis of the skin condition diseases they have.

Beyond acne and eczema classification, traditional clinical assessments of psoriasis have predominantly relied on dermatologists' subjective evaluations based on visual inspection and patient-reported symptoms. The most widely used method, the Psoriasis Area and Severity Index (PASI), often suffers from inter-observer variability and inconsistent scoring across different clinical environments due to individual interpretations [13]. Psoriasis, a chronic autoimmune condition, exhibits diverse lesion morphologies—including plaque, guttate, pustular, and erythrodermic forms—which further complicates the development of standardized and objective evaluation frameworks [14].

To mitigate these limitations, researchers have increasingly turned to computer vision and machine learning techniques to enable more objective, reproducible, and scalable approaches for psoriasis detection and severity analysis. Arunkumar and Jayanna [15] introduced one of the earliest automated methods by analyzing RGB color histograms to distinguish affected skin regions from healthy skin based on color features like redness, greenness, and blueness. While this approach marked a step toward automation, it lacked robustness due to its reliance on shallow features and omission of deeper semantic representations.

George et al. [16] addressed these shortcomings by proposing a scale severity assessment model based on local descriptors and a bag-of-visual-words (BoVWs) framework. Their technique employed superpixel segmentation combined with diverse color and texture features and classifiers such as Support Vector Machine (SVM) and Random Forest. The method achieved a scoring accuracy of 80.81% for scaling severity but remained limited in scope, as it did not encompass additional PASI criteria like erythema or induration.
Vincent and Jayasingh [17] provided a comprehensive review of psoriasis detection systems, analyzing various image preprocessing, segmentation, and feature extraction techniques. Their findings highlighted the fragmented nature of current solutions and emphasized the need for integrated deep learning models capable of addressing multiple PASI parameters within a single architecture.

In a significant advancement, Syu et al. [18] developed a deep neural network (DNN)-based model designed for multi-level feature extraction using stacked convolutional layers. Their architecture demonstrated improved accuracy in lesion detection compared to traditional diagnostic methods. However, it lacked structured mechanisms for severity classification, revealing an important gap in end-to-end psoriasis management systems.

Roslan et al. [19] extended the deep learning application to psoriasis subtype classification using convolutional neural networks (CNNs). Their model attained an accuracy of 82.9% in identifying plaque psoriasis, showcasing the clinical potential of CNNs. Nevertheless, challenges remain, particularly in terms of dataset diversity, generalization, and real-time deployment in clinical settings.

To further enhance model performance, Tan and Le [20] introduced MixConv—a mixed depthwise convolutional strategy integrating multiple kernel sizes within a single convolutional layer. This architecture allows models such as MobileNetV2 to simultaneously capture local fine-grained and global contextual features, significantly benefiting dermatological applications where lesion size and texture

vary greatly. Tsang [21] emphasized the relevance of such multi-scale modeling in improving lesion characterization.

Explainability is another essential aspect in clinical AI deployment. Grad-CAM, proposed by Selvaraju et al. [22], enables visual explanation by highlighting the most important image regions influencing CNN predictions. This helps clinicians interpret model decisions and promotes trust in AI-assisted diagnoses. Saranya and Subhashini [23] further reviewed the importance of Explainable AI (XAI) across healthcare domains, stressing the need for interpretable and transparent systems particularly in sensitive areas such as dermatology.

Beyond progress in segmentation and classification, a recent avenue of investigation has focused on closing the loop through mobile applications or real-time diagnostic systems that integrate these machine learning models. The aim of such integration is to make these valuable diagnostic tools more accessible for patients and their healthcare providers by enabling examinations and continuous monitoring of various skin diseases at home or during scheduled visits. These systems empower patients to actively engage with and adhere to their treatment plans, providing rapid feedback that allows them to receive immediate guidance on areas requiring attention. Such advancements signify a paradigm shift in dermatological care, as widespread technologies and advanced deep learning approaches have the potential to transform how skin diseases are diagnosed and monitored both within clinical settings and beyond. This growing body of research highlights the importance of interdisciplinary collaboration to develop robust care models that are not only effective but also scalable and feasible for patients.

The novel application, DermaScope AI, leverages proprietary machine learning capabilities to optimize the diagnosis of various skin diseases. By utilizing skin image data, DermaScope AI can accurately detect and classify a wide range of skin conditions. As a cutting-edge solution compared to traditional diagnostic

techniques, DermaScope AI streamlines the diagnostic process with significant time savings and improved accessibility, particularly in areas with limited availability of dermatologists. This solution further demonstrates the vital role of machine learning in providing more effective and efficient dermatological care, serving as a generalizable and objective tool for both clinicians and patients across a broad spectrum of skin conditions.

1.3 Research Gap

The current literature on acne detection and classification, though demonstrating considerable progress using deep learning methods, presents several relevant limitations that discourage their wider adoption and performance. Many studies rely on the datasets with limited diversity in skin tones and acne severity, leading to the models that may not generalize well across diverse population. Besides, the majority of systems perform either acne detection or severity classification and not both, overlooking the necessity of a comprehensive solution with both functionalities. Furthermore, systems with longitudinal tracking of patient enhancement are few, but its importance is in allowing dermatologists to assess treatment efficacy in the long term. These variations point to the necessity of having a more integrated and complete solution in acne detection and treatment. In contrast, the system we present transcends these limitations by utilizing a diverse dermatologist-approved dataset, incorporating twin models for both acne detection and grade classification, and including a patient progress tracking feature. This synergy not only improves generalizability to different skin types and acne severity but also clinical utility by providing actionable information for both patients and clinicians.

Although significant progress has been made in leveraging machine learning for the diagnosis of eczema and the assessment of its severity, several key gaps still remain unmet. Most systems proposed so far either perform simple classification

or segmentation tasks with no comprehensive severity assessment framework or advanced technique to handle complex lesion patterns. Explainability is also a limitation of most of these systems. It would be hard for clinicians to interpret and trust the outputs, hence limiting their adoption in real-world settings. What affects the generalization of these models across diverse populations is the absence of mechanisms to handle variability in lesion shapes, skin tones, and lightning conditions. To bridge these gaps, a novel approach integrating attention mechanisms with machine learning is required. This would enhance the system's ability to focus on critical regions of interest, improve accuracy in diagnosis and severity assessment, and provide explainable and reliable outputs suitable for clinical use.

Although machine learning has been applied in psoriasis severity estimation, there are some important gaps that have not been filled. Most existing methods rely on either simple lesion classification or segmentation but not on full severity assessment frameworks and generalizability to diverse clinical environments. Existing methods do not have the capability to generalize across skin types, lesion types, and patient populations, thus constraining practical usability. The absence of XAI methods also does not allow clinicians to interpret and trust the outputs, constraining clinical translation. This study demolishes these limitations by introducing an effective AI based framework that combines state-of-the-art deep learning architectures and XAI methods to offer accurate, interpretable, and scalable psoriasis severity scoring for clinical practice.

Table 1:Comparison of the features addressed by the previous research studies for Acne

Feature / Gap	**[1]**	**[2]**	**[3]**	**[4]**	**[5]**	**Proposed Solution**
Incorporate advanced deep learning techniques	Yes	Yes	Yes	Yes	Yes	Yes

Enhanced explainability	No	No	No	No	No	Yes
Diverse Datasets	No	Yes	Yes	Yes	Yes	Yes
Acne severity assessment	No	Yes	No	No	Yes	Yes
Healing Progress Tracking	No	No	No	No	No	yes

Table 2:Comparison of the features addressed by the previous research studies for Eczema

Feature / Gap	**[8]**	**[9]**	**[10]**	**[11]**	**[12]**	**Proposed Solution**
Incorporate advanced deep learning techniques	Yes	No	Yes	Yes	Yes	Yes
Enhanced explainability	No	No	No	No	No	Yes
Improved robustness	Yes	No	Yes	Yes	Yes	Yes
Eczema severity assessment	No	No	No	Yes	No	Yes
Incorporation of attention mechanisms	No	No	No	No	No	Yes
Integrated system	No	No	No	No	No	Yes

Report generation	No	No	No	No	No	Yes
Progress analysis	No	No	No	No	No	Yes

Table 3:Comparison of the features addressed by the previous research studies for Psoriasis

Features	**[15]**	**[16]**	**[17]**	**[18]**	**[19]**	Proposed System
RGB-based Psoriasis Detection	Yes	No	Yes	No	No	Yes
Deep Learning for Lesion Classification	No	Yes	Yes	Yes	Yes	Yes
Robustness Across Diverse Skin Tones & Lesion Types	No	No	No	No	Yes	Yes
Real-Time, Clinically Scalable Psoriasis Severity Classification	No	No	No	No	No	Yes
Explainable AI Integration (e.g., Grad-CAM)	No	No	No	No	No	Yes
Mix Convolution for Multi-Scale Feature Extraction	No	No	No	No	No	Yes
Ghost Module for Lightweight and Efficient Architecture	No	No	No	No	No	Yes

1.4 Research Problem

Prevalent chronic skin disorders such as eczema, psoriasis, and acne affect millions of individuals worldwide, resulting in substantial physical, emotional, and social burdens. Achieving accurate diagnosis and severity grading is essential for optimal disease management and personalized treatment strategies. However, current diagnostic procedures predominantly rely on subjective visual examination by dermatologists, which is often time-consuming, inconsistent, and subject to clinician-to-clinician variability. Factors such as presentation complexity, environmental conditions, and varying levels of clinical expertise contribute to inconsistencies in diagnosis and severity grading, ultimately impacting treatment outcomes. Furthermore, manual assessment processes are resource-intensive and inefficient for large-scale screenings or in remote areas with limited access to dermatological expertise.

Despite advancements in artificial intelligence (AI) and computer vision, existing automated systems for skin disease diagnosis present significant limitations. Traditional machine learning models and simplistic image processing algorithms often fail to generalize across diverse datasets, particularly when confronted with variations in skin tones, lesion morphology, and lighting conditions. Additionally, many AI models lack explainability and real-time capabilities, limiting their acceptance and usability within clinical settings.

There is a pressing need for a robust, scalable, and unbiased AI-driven system capable of accurately detecting and classifying the severity of various skin diseases while addressing challenges related to dataset diversity and model interpretability. This research aims to bridge this gap by developing an advanced AI-based framework that effectively interprets clinical images, delivers consistent severity scores, and integrates seamlessly into dermatological practice. Such a system will enhance diagnostic accuracy, streamline decision-making, and improve overall patient care through a more objective and accessible approach.

1.5 Research Objective

1.5.1 Main Objective

The main objective of our investigation is to develop a comprehensive system for advanced diagnosis and severity evaluation using machine learning in various skin diseases. DermaScope AI aims to overcome challenges associated with subjective traditional methods by providing an objective, faster, and more accurate tool to support clinical decision-making and enhance patient outcomes.

DermaScope AI will focus on precise diagnosis and detailed severity assessment of a broad range of skin conditions, including but not limited to eczema, psoriasis, acne, and melanoma. Leveraging the most advanced developments in machine learning, particularly in image analysis and pattern recognition, this system will accurately evaluate the severity of skin disorders by analyzing clinical symptoms that align with established diagnostic criteria. By integrating these state-of-the-art technologies, DermaScope AI will deliver in-depth and detailed analyses, empowering healthcare professionals to make more data-driven and precise decisions about patient care. Such a system will enable them to refine and personalize treatment strategies far beyond current capabilities.

Another essential functionality of DermaScope AI is its ability to track and update comprehensive patient profiles. This feature will highlight critical aspects related to specific conditions, past treatments, and progress monitoring. The personalized approach ensures that assessments are based not only on general diagnostic criteria but also on the unique characteristics and needs of each patient. This functionality will facilitate more accurate logging of skin conditions, enabling early detection before they reach advanced stages and ensuring timely, appropriate treatment.

Overall, this study aims to create an AI-based solution for diagnosing and managing a wide range of skin diseases called DermaScope AI. It will enhance diagnostic accuracy and clinical assessment, providing a more objective and consistent

method for evaluation. Ultimately, DermaScope AI will improve the patient care journey by enabling timely and precise assessments, which are critical for recommending appropriate treatments and improving overall patient well-being.

1.5.2 Sub Objectives

Sub Objective 1: Develop a System for Real-Time Monitoring and Management of Skin Diseases Using AI and Machine Learning Techniques

This objective focuses on designing and implementing a real-time monitoring system that leverages AI and machine learning models to continuously track and manage various skin diseases, including acne, eczema, and psoriasis. By integrating machine learning algorithms with real-time image processing capabilities, the system will provide immediate feedback to patients and clinicians, enhancing accessibility and diagnostic accuracy. The proposed system aims to facilitate consistent disease tracking over time, allowing for timely interventions and improved decision-making in both clinical and non-clinical settings.

Sub Objective 2: Enhance Diagnostic Accuracy in Skin Disease Detection and Management Through Advanced Machine Learning Techniques

The objective aims to improve diagnostic accuracy by applying state-of-the-art machine learning techniques to classify and assess skin diseases more effectively. By training models on diverse datasets containing images of acne, eczema, and psoriasis across varying demographics and conditions, the system will achieve robust performance and generalization capabilities. Incorporating deep learning architectures such as Convolutional Neural Networks (CNNs) and attention mechanisms will enhance pattern recognition, contributing to reliable and objective diagnostic outcomes that are less susceptible to clinician bias and variability.

Sub Objective 3: Develop an Advanced System for Accurate Detection and Severity Assessment of Eczema Using Image Processing Techniques

This objective focuses on building a specialized framework within DermaScope AI to accurately detect and assess the severity of eczema using advanced image processing and machine learning techniques. By analyzing clinical images against established diagnostic criteria such as EASI scores, the system aims to provide consistent severity grading for erythema, induration, lichenification, and other relevant symptoms. Automation of this process will streamline diagnostic workflows, allowing for quicker, more objective, and standardized assessments that support individualized treatment planning.

Sub Objective 4: Develop a Comprehensive Suite of Machine Learning and AI-Based Models for Acne Assessment and Management

This objective aims to construct a dedicated module within DermaScope AI that focuses on the detection, classification, and severity assessment of acne. Leveraging techniques such as deep learning, feature extraction, and pattern recognition, the system will evaluate various clinical indicators including lesion count, inflammation, and scarring. Through automated analysis, this module will provide reliable and quantitative assessments, assisting clinicians in making data-driven decisions about treatment strategies and monitoring progress over time.

Sub Objective 5: Apply Machine Learning and Computer Vision to Detect Psoriasis and Assess Its Scaling Severity from Images

This objective is centered on developing a powerful machine learning model for accurately detecting psoriasis and evaluating its severity, particularly focusing on scaling and plaque characteristics. By employing image processing techniques and deep learning architectures, the system will quantify scaling, erythema, and plaque thickness to provide consistent severity scores. Integrating this model into the

broader DermaScope AI framework will enhance its capability to address multiple skin diseases, ultimately providing comprehensive support for clinicians and patients dealing with psoriasis.

2. METHODOLOGY

2.1 Requirement Gathering

The requirement gathering phase played a pivotal role in ensuring that the development of the proposed system, DermaScope AI, is closely aligned with the practical needs and expectations of its intended end users, including dermatologists and patients. To achieve this, a comprehensive and structured approach was adopted, including stakeholder interviews, expert consultations, and literature reviews, each aimed at extracting detailed insights into clinical workflows, diagnostic criteria, user interface preferences, and pain points associated with existing diagnostic methods for acne, eczema, and psoriasis. The outcomes of this phase were used to define both functional requirements and non-functional requirements. By thoroughly understanding these requirements early in the development process, DermaScope AI is designed to deliver a reliable, accessible, and clinically meaningful solution for the diagnosis, assessment, and management of acne, eczema, and psoriasis.

2.1.1 Stakeholder Identification

- Dermatologists - to provide clinical insights and validate diagnostic criteria.
- Patients - to understand user experience needs.
- Software developers - to assess technical feasibility and implementation constraints.

2.1.2 Methods Used

- Interviews with dermatologists and clinicians to understand diagnostic workflows, image assessment criteria (e.g., EASI for eczema, PASI for psoriasis, and standardized grading systems for acne), and the key indicators of severity for each condition.
- Review of existing systems and literature to identify the limitations and strengths of current machine learning tools used for diagnosing and assessing acne, eczema, and psoriasis.
- Consultations with technical experts to assess the feasibility of integrating Explainable AI (XAI), attention mechanisms, and multi-scale image processing techniques applicable to all three skin diseases for enhanced diagnostic accuracy and interpretability.

2.1.3 Functional Requirements

- The application should allow users to upload affected skin lesion images at high resolution and process the uploaded images for acne, eczema, and psoriasis.
- The application should be able to identify, and segment affected regions of acne, eczema, and psoriasis from the input images using a system that employs advanced machine learning techniques.
- The application should offer an end-to-end analysis of the severity of acne, eczema, and psoriasis using user-friendly dashboards designed for healthcare providers.
- The application should establish personalized profiles for each user, allowing the service provider to save various information such as severity evaluations, treatment history, or progress assessments for acne, eczema, and psoriasis.

- The application must assess the severity of acne, eczema, and psoriasis based on clinical indicators and established criteria, such as EASI for eczema, PASI for psoriasis, and standardized grading systems for acne.

2.1.4 Non-Functional Requirements

- Usability: The system interface must be user-friendly and intuitive for both clinicians and non-technical users.
- Accuracy: The ML model must provide clinically reliable results with high accuracy.
- Performance: The system should deliver real-time or near real-time predictions.
- Scalability: It should be scalable to integrate with mobile platforms or remote healthcare systems.
- Security and Privacy: All patient data must be securely stored and processed, adhering to relevant data protection regulations.

2.1.5 Personal Requirements

- Guidance from supervisors on the domains of acne, eczema, psoriasis, and data science to ensure a well-rounded and interdisciplinary approach.
- Understanding the specific areas of each domain, acne, eczema, psoriasis, and data science to effectively proceed with the selected research area and develop a robust AI-based diagnostic and management system.

2.2 Data Collection

The dataset used in this study was obtained from online sources [24][25][26][27], specifically collected to support the identification, classification, and severity assessment of acne, eczema, and psoriasis. The dataset comprises a variety of images categorized into relevant classes for each skin condition: acne, eczema, psoriasis, and non-disease. These images were sourced from publicly available databases and online resources, ensuring a diverse range of skin conditions, lighting conditions, and image quality to enhance model robustness. The dataset serves as a foundational resource for developing and testing the proposed system, facilitating accurate detection and classification of acne, eczema, and psoriasis symptoms.

To enhance the clinical relevance and accuracy of the severity classification dataset, expert dermatologists were consulted to provide insights for categorizing the images based on their severity. This process involved carefully examining the collected images and categorizing them according to established clinical grading systems for each condition. For acne, images were classified into four distinct severity levels: Mild, Moderate, Severe, and High Severe. For eczema, images were categorized into three levels: Mild, Moderate, and Severe. Psoriasis images were categorized into three levels: Mild, Moderate, and Severe.

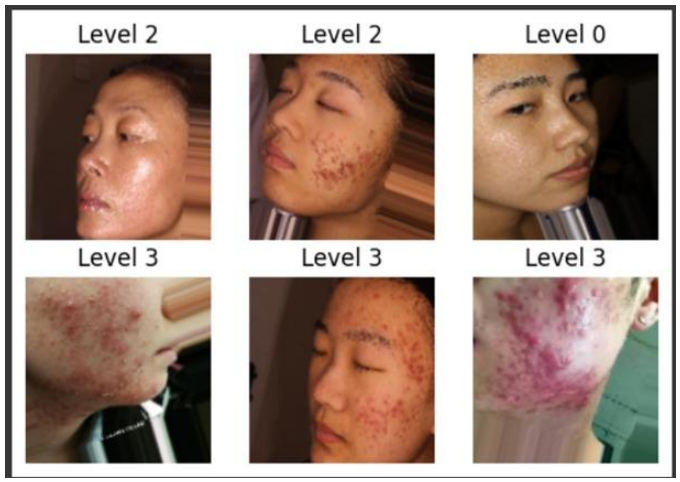


Figure 1: Sample images from the acne and non-acne classification dataset

Figure 2: Sample images from the eczema and non-eczema classification dataset

Figure 3: Sample images from the Psoriasis classification dataset

2.3 Data Exploration

To gain insights into the characteristics and distribution of the dataset, various visualizations were generated to analyze key aspects such as class distribution, severity levels, and feature patterns. These diagrams and charts provide a comprehensive overview of the dataset, illustrating the proportion of images, as well as the distribution of severity levels. Visualizing the data not only helps identify potential class imbalances but also offers valuable insights into the dataset's structure, aiding in the development of effective machine learning models for the detection and severity classification. The following visual representations present these key aspects in detail.

2.3.1 Acne

Acne and non-acne classification dataset

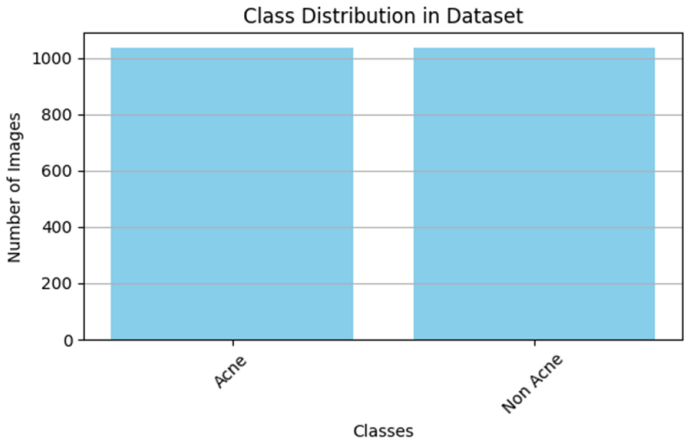


Figure 4: Acne Dataset

- The dataset shows a balanced distribution between "Non Acne" and "Acne" ensuring equal representation of both classes.
- This balance enhances the model's ability to learn effectively and reduces potential bias in predictions.

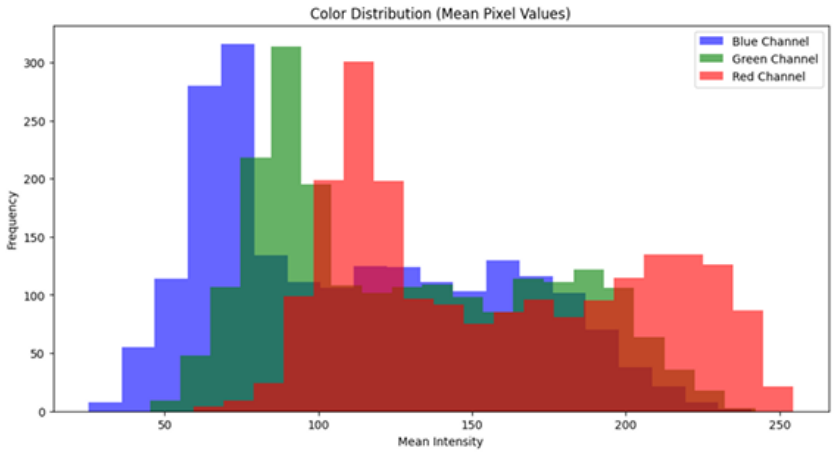


Figure 5: Color distribution

Red Dominance: The red channel generally has higher mean intensities compared to the blue and green channels. This suggests that your images tend to have a warmer or reddish overall tone. Green Peaks: The green channel shows some prominent peaks around the 100-125 intensity range, indicating a concentration of images with moderate green tones. Blue Variation: The blue channel has a wider spread of mean intensities, suggesting more variation in blue tones across the images. Random images exploration from the dataset

Acne severity classification dataset

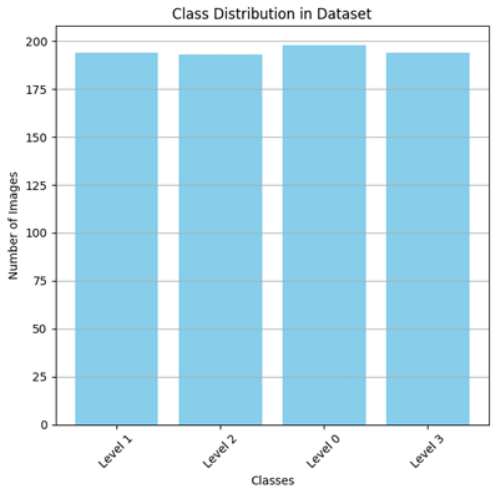


Figure 6: Acne Severity Dataset

The dataset shows a nearly balanced distribution between "Level 0" , "Level 1" , "Level 2" , "Level 3" ensuring equal representation of both classes.

This balance enhances the model's ability to learn effectively and reduces potential bias in predictions.

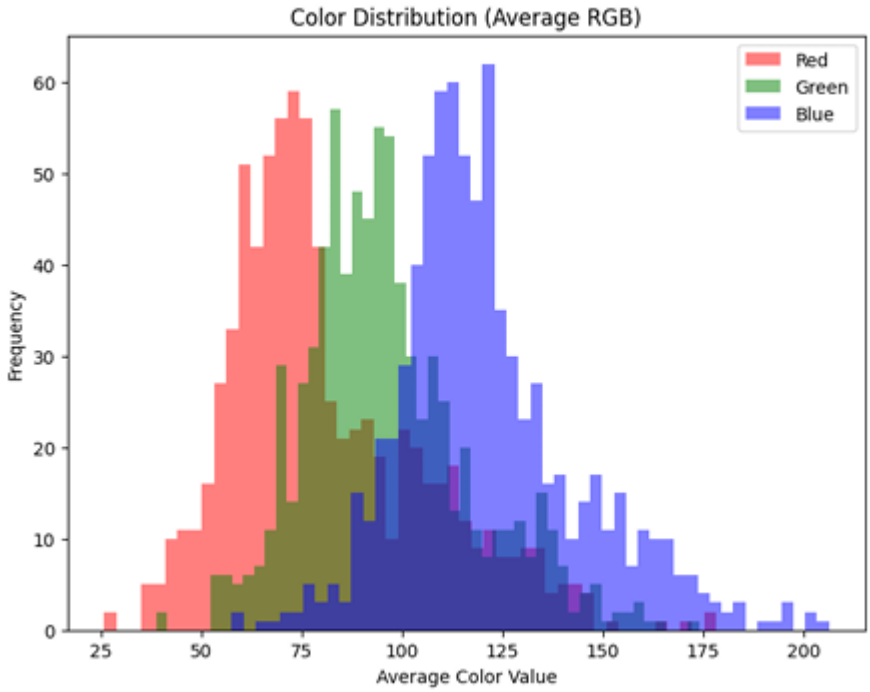


Figure 7: Color Distribution of Acne Severity Detection

Overall Distribution: The graph indicates that the images in the dataset tend to have a higher average blue value compared to red and green. This suggests the images might be slightly bluish in tone overall.

2.3.2 Eczema

Eczema and non-eczema classification dataset

The Class Distribution chart indicates a well-balanced dataset, with 604 images classified as eczema and 589 images categorized as non-eczema. This near-equal distribution is essential for developing a robust and unbiased model, reducing the likelihood of skewed predictions and improving overall classification accuracy.

Figure 8: Visualization showing the balanced distribution between normal skin and eczema skin images

Lastly, the Color Distribution plot displays the mean pixel intensity for the red, green, and blue channels across the dataset. This visualization highlights the range of color intensities, which can influence the model's ability to distinguish between eczema and non-eczema skin. Identifying these patterns helps in understanding potential color differences between classes, contributing to better feature extraction during model development.

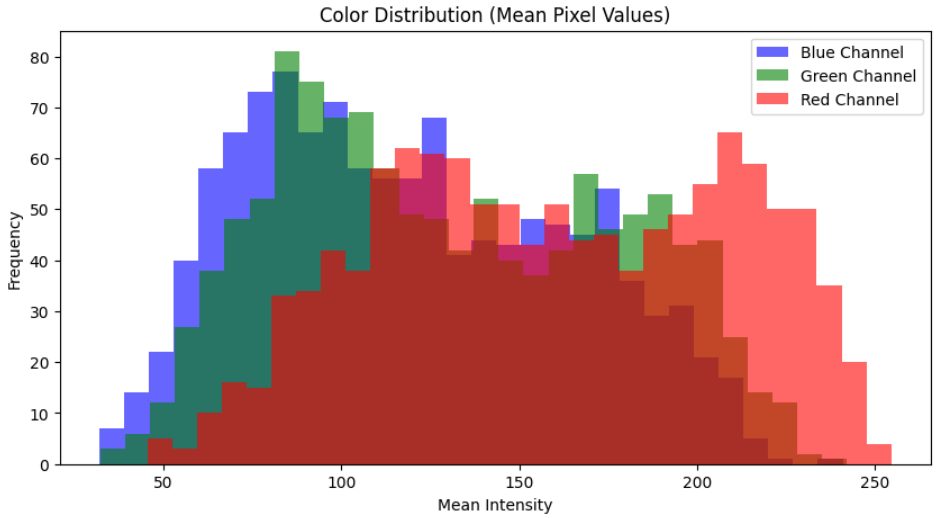


Figure 9: Graph representing the mean pixel intensity for the red, green, and blue channels in the dataset

Eczema severity classification dataset

The Severity Class Distribution chart illustrates the distribution of eczema severity levels within the dataset. The dataset shows an imbalance, with a significantly higher number of mild cases compared to moderate and severe cases. This uneven distribution highlights the need for techniques such as data augmentation or weighted loss functions to ensure balanced model training and improved performance.

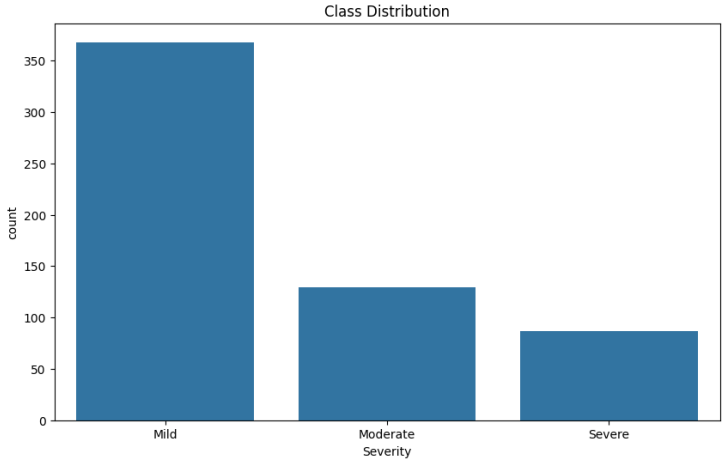


Figure 10: Visualization showing the distribution of eczema severity levels categorized as mild, moderate, and severe

The Color Distribution plots present the mean pixel intensity for the red, green, and blue channels across the dataset, categorized by severity levels. The color intensity distribution reveals distinct patterns, with mild eczema images displaying a broader range of intensities, while moderate and severe cases generally show lower intensity distributions. Identifying these color characteristics is valuable for enhancing feature extraction strategies and improving the model's ability to differentiate between eczema severity levels.

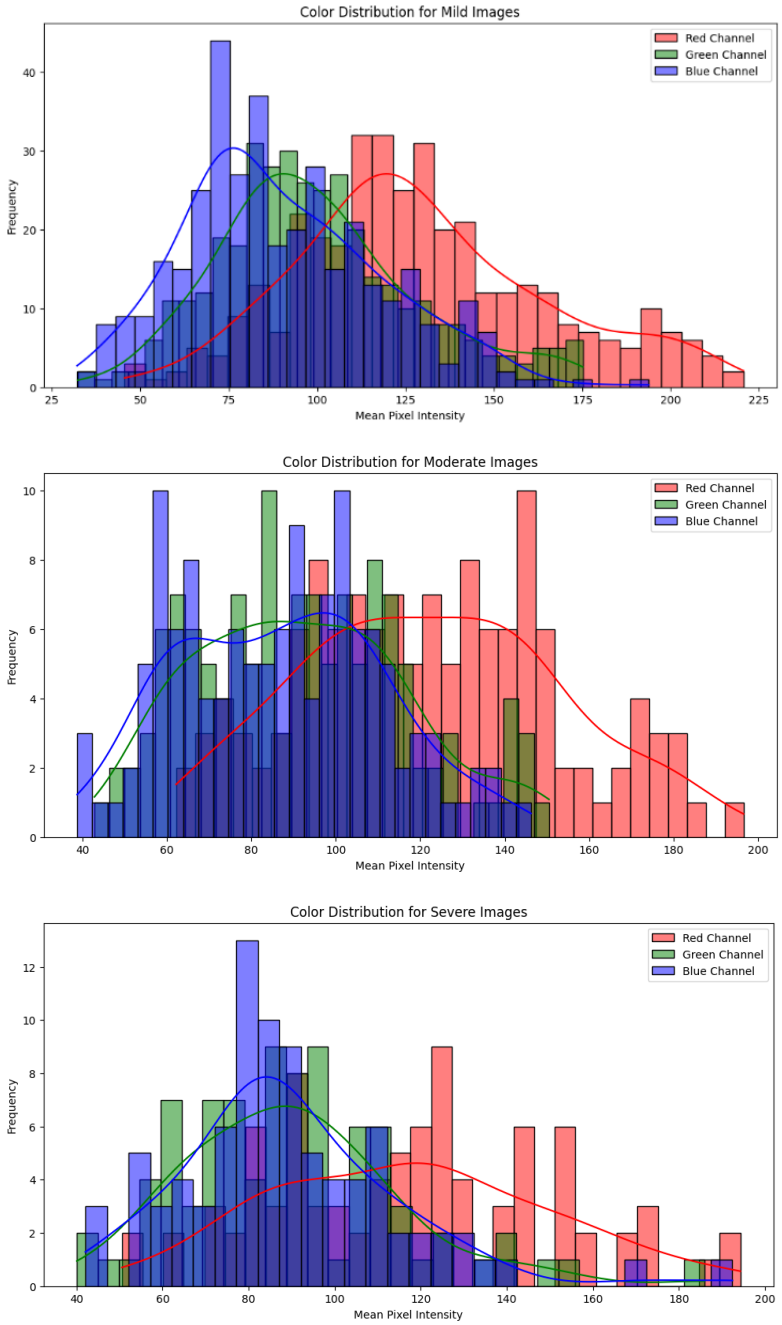


Figure 11: Graphs depicting the mean pixel intensity distributions for mild, moderate, and severe eczema images

These visual insights provide valuable information for refining data preprocessing steps and optimizing model performance.

2.3.3 Psoriasis

Psoriasis and non- Psoriasis classification dataset

The Class Distribution chart demonstrates a well-balanced dataset, with 273 images categorized under NORMAL_SKIN and 271 images under PSORIASIS_SKIN. This near-equal distribution is crucial for training a fair and unbiased classification model, minimizing the risk of class imbalance that could lead to overfitting or biased predictions.

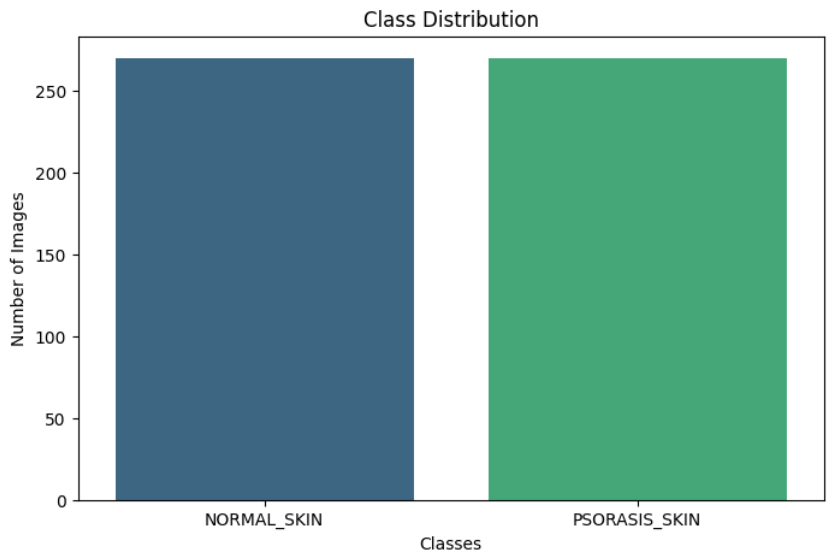


Figure 12: Visualization showing the distribution between normal skin and Psoriasis skin images

The Color Distribution (Mean Pixel Values) plots further enhance understanding by illustrating the mean intensity values for the red, green, and blue (RGB) channels in both classes. For NORMAL_SKIN, red and green channel intensities tend to be higher, with peaks around the 200 range, whereas the blue channel is more evenly spread with a peak around 150. This indicates the presence of more natural and warm tones in healthy skin.

In contrast, the PSORIASIS_SKIN class shows a shift in pixel intensity values. The red channel peaks around 150, while the green and blue channels exhibit broader distributions between 75 and 125. These shifts in color intensity reflect the visual differences caused by psoriatic lesions, such as redness, scaling, and discoloration. These color distribution insights can guide effective feature extraction, enhancing the model's ability to differentiate between normal and psoriatic skin during classification. Understanding such visual differences is fundamental in designing CNNs and preprocessing pipelines tailored for dermatological image classification tasks.

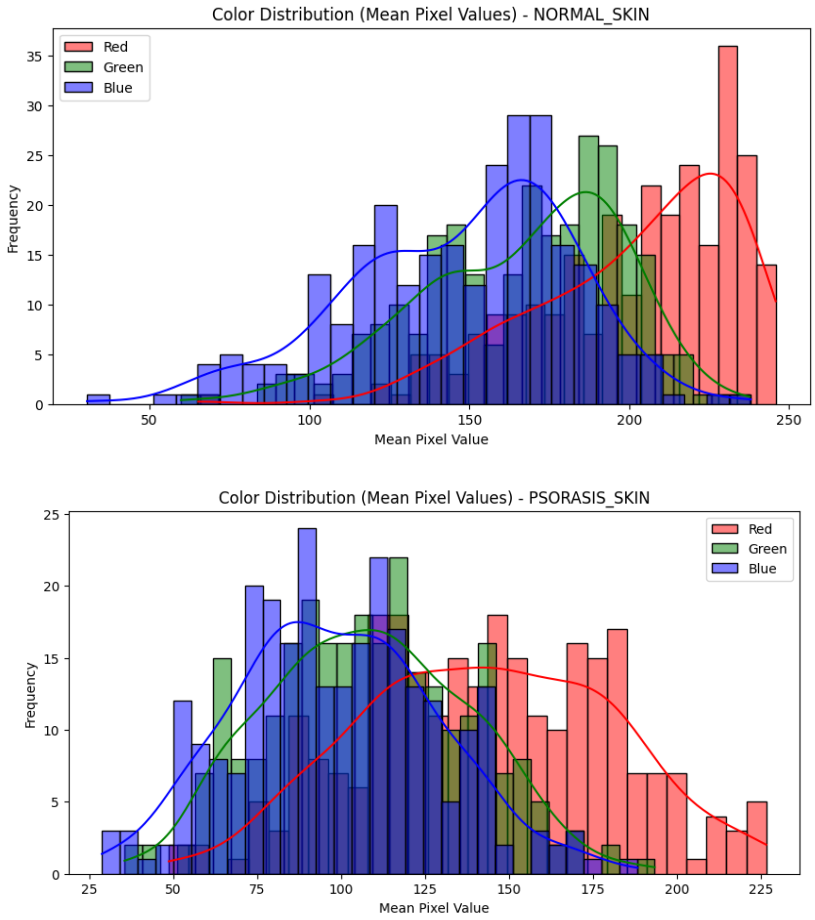


Figure 13: Graphs depicting the mean pixel intensity distributions for normal skin and Psoriasis skin images

Psoriasis severity classification dataset

The Class Distribution by Severity Level chart reveals an evenly balanced dataset across the three severity levels: Mild, Moderate, and Severe, with 100 images allocated to each class. This uniform distribution is essential for ensuring that the model does not exhibit bias toward any specific severity level, thereby enhancing its generalization ability and diagnostic reliability.

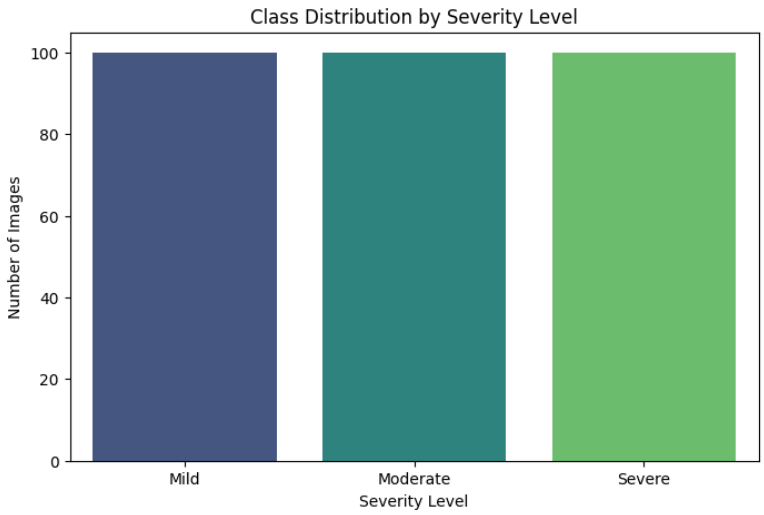


Figure 14: Visualization showing the distribution of Psoriasis severity levels categorized as mild, moderate, and severe

The Color Distribution (Mean Pixel Values) plots reveal noticeable differences across mild, moderate, and severe psoriasis images. Mild cases show higher red intensity values (150–200) with balanced green and blue channels, indicating subtle inflammation. Moderate cases exhibit a broader color spread with lower green and blue intensities, reflecting increased lesion severity. Severe cases display the most intense red values (140–170) and significantly reduced green and blue tones, highlighting severe erythema and scaling. These color differences are useful for feature extraction in deep learning models to improve severity classification accuracy.

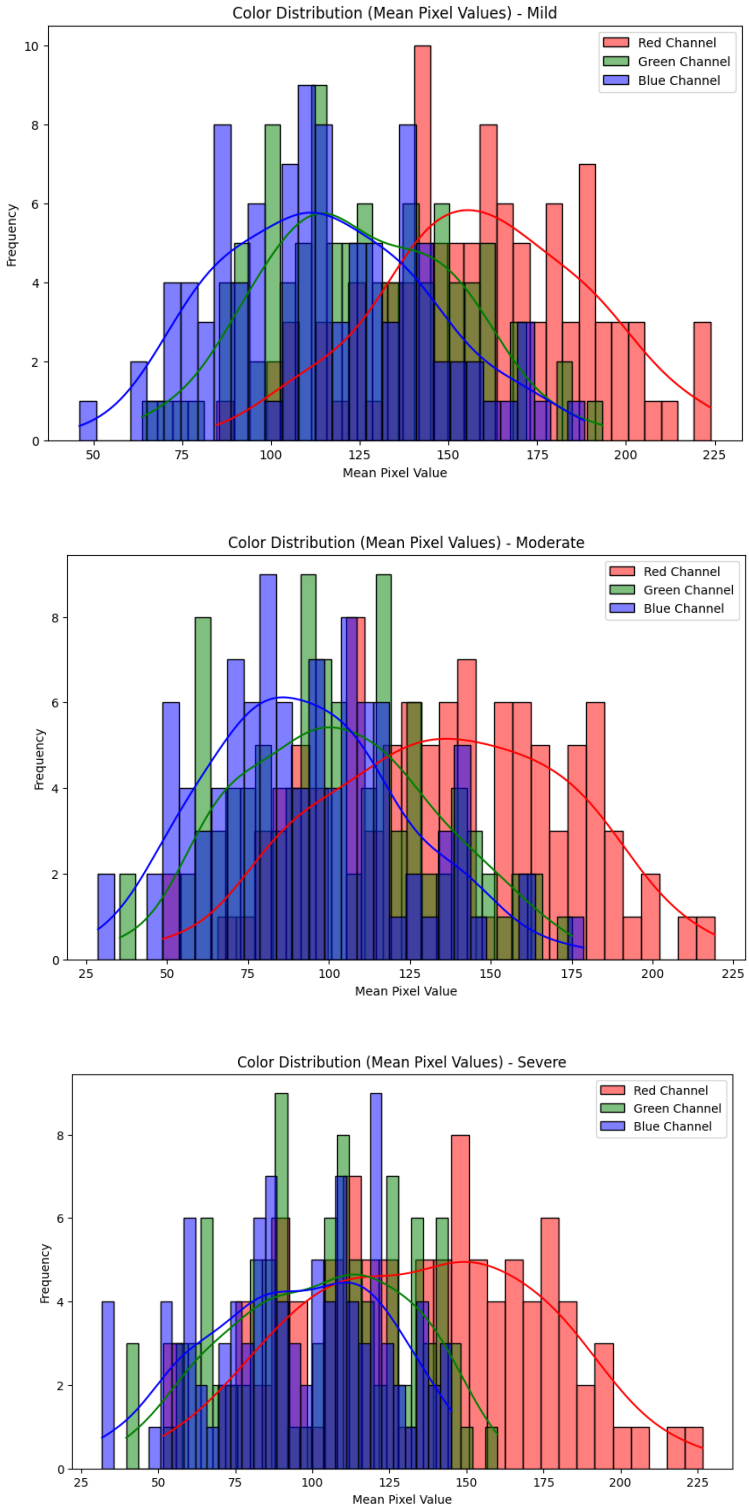


Figure 15: Graphs depicting the mean pixel intensity distributions for mild, moderate, and severe Psoriasis image

2.4 Data Preprocessing

2.4.1 Acne

The development of an effective acne detection model necessitated a carefully designed data preprocessing and augmentation pipeline to ensure robust performance across varying clinical presentations. The preprocessing methodology was structured to address key challenges in dermatological image analysis while optimizing the dataset for deep learning applications.

At the core of the preprocessing workflow was the standardization of input images to ensure consistency in model training. All acne images were systematically resized to dimensions of 224×224 pixels, a resolution chosen to balance computational efficiency with sufficient detail preservation for lesion identification. This resizing operation was accompanied by pixel value normalization, where raw image intensities were scaled to a [0,1] range through division by 255. This normalization step proved crucial for stabilizing gradient updates during the subsequent neural network training process. The conversion from BGR to RGB color space addressed potential inconsistencies arising from different image acquisition systems, ensuring color representation uniformity across the dataset.

The augmentation strategy incorporated both geometric transformations and color-space manipulations to comprehensively address the variability inherent in clinical dermatological images. The geometric augmentation pipeline, implemented through Keras' ImageDataGenerator, introduced controlled variations in image perspective and orientation. A 20-degree rotation range accommodated natural variations in camera angle during image capture, while width and height shifts of 20% accounted for minor framing inconsistencies. Shear transformations at 20% intensity and zoom variations within a 20% range further enhanced the model's ability to recognize lesions under different presentation conditions. Horizontal flipping was employed to enforce left-right symmetry awareness, particularly

valuable given the bilateral nature of many acne presentations. The 'nearest' fill mode selection ensured that these transformations did not introduce artificial artifacts at image boundaries.

Complementing the geometric augmentations, a sophisticated color-based processing approach was developed to specifically target the characteristic features of acne lesions. The implementation of HSV color space conversion facilitated more accurate detection of erythematous regions associated with inflammatory acne. Through the application of dual-range thresholding (0-10° and 160-180° hue ranges), the system could reliably identify both bright red active lesions and deeper inflammatory regions. Subsequent morphological processing, including opening and closing operations with a 5×5 kernel, served to refine the lesion masks by eliminating noise and filling discontinuities in detected regions. This color-based processing not only supported data augmentation but also enabled precise bounding box generation for lesion localization tasks.

The preprocessing pipeline incorporated intelligent dataset partitioning through stratified sampling, maintaining the original distribution of acne severity levels across both training and validation subsets. This approach guarded against potential biases that might arise from uneven class representation. The implementation included comprehensive logging mechanisms to track all preprocessing operations, facilitating reproducibility and error diagnosis. These logs captured critical metadata including processing timestamps, image sources, transformation parameters, and any encountered exceptions.

The augmentation and preprocessing methodology demonstrated particular effectiveness in addressing several key challenges in acne image analysis. Variations in lighting conditions, a common issue in clinical photography, were mitigated through the combined effects of color normalization and HSV-based processing. Differences in patient skin tones were accommodated by the color-agnostic nature of the geometric augmentations coupled with the adaptive

thresholding in HSV space. The preprocessing pipeline's ability to generate diverse training examples from limited original data proved invaluable in preventing overfitting, a significant concern given the relatively small size of many medical image datasets.

This comprehensive preprocessing framework supported the subsequent development of a deep learning system capable of both acne severity classification and precise lesion localization. The attention to both geometric and color-based transformations resulted in a model that maintained diagnostic accuracy across a wide range of image acquisition conditions and patient presentations. The success of this approach underscores the importance of tailored preprocessing in medical computer vision applications, where biological variability and imaging inconsistencies present unique challenges not typically encountered in general image recognition tasks. Future refinements could explore adaptive augmentation strategies that dynamically adjust parameters based on individual image characteristics, potentially offering further improvements in model performance and generalization capability.

2.4.2 Eczema

The data preprocessing phase was a crucial step in preparing the raw image data for training an accurate and reliable machine learning model for eczema detection and severity classification. Initially, the images from the eczema detection dataset were organized into two distinct categories: 'Normal Skin' and 'Eczema Skin', while the eczema severity classification dataset was organized into three categories, ‘Mild’, ‘Moderate’, and ‘Severe’. A custom Python function was implemented to iterate through these directories, load each image using the OpenCV library, and convert the image from BGR to RGB color space to align with standard

conventions used in deep learning models. Each image was then resized to a fixed resolution of 224x224 pixels, ensuring uniform input dimensions compatible with popular convolutional neural network architectures. During this process, images were labeled as 0 for normal skin and 1 for eczema, establishing the binary classification foundation for model development.

Following the resizing process, all image pixel values were normalized to a range between 0 and 1 by dividing each pixel value by 255.0. This normalization step was essential for reducing the influence of varying lighting conditions and accelerating the training convergence of the model. Once preprocessing was completed, the dataset was split into training and validation subsets using an 80:20 ratio. This split ensured that the model would be evaluated on a distinct portion of data not seen during training, aiding in assessing its ability to generalize and preventing overfitting. To enhance the model's generalization capability and mitigate the risk of overfitting, particularly due to the limited size and variability of the dataset, comprehensive data augmentation techniques were applied using the ImageDataGenerator class from the Keras library.

Data augmentation, a critical preprocessing step in medical image analysis, synthetically increases the diversity of the training dataset by applying a range of transformations to existing images. In this study, the augmentation pipeline incorporated several operations including random rotations up to 30 degrees, horizontal and vertical shifts, zooming in and out, shearing transformations, and horizontal flipping of images. These transformations simulate real-world variations in clinical scenarios where skin conditions might be photographed from different angles, under inconsistent lighting conditions, or at varying distances. By artificially expanding the dataset in this manner, the model was exposed to a broader set of visual patterns, allowing it to learn more robust and invariant features essential for accurately distinguishing between normal and eczema-affected skin. The final step involved converting the preprocessed data into TensorFlow Dataset

objects, shuffling the data, batching it into groups of 32 images, and prefetching it to enable asynchronous data loading. This structured preprocessing workflow ensured efficient and optimized data feeding during the training process, laying the groundwork for developing a high-performance model capable of accurate and consistent eczema classification.

2.4.3 Psoriasis

The data preprocessing phase was a crucial step in preparing the psoriasis image dataset for accurate detection and severity classification. Initially, the dataset was organized into two main categories for binary classification 'Normal Skin' and 'Psoriasis Skin' and into three distinct classes ‘Mild’, ‘Moderate’, and ‘Severe’ for severity-level classification. To facilitate automated loading, a custom Python script was developed to iterate through the directory structure, load images using OpenCV, and convert them from BGR to RGB color space to comply with deep learning model input standards. Each image was then resized to a fixed resolution of 224×224 pixels for classification tasks, while a resolution of 640×640 pixels was used for YOLOv8-based lesion detection, ensuring consistency across training inputs.

Following resizing, all pixel values were normalized to the range [0, 1] by dividing each value by 255.0. This normalization helped mitigate the effects of lighting variations and accelerated model convergence during training. The entire dataset was then split into training and validation subsets using an 80:20 ratio, allowing for reliable evaluation of the model’s generalization capability on unseen data. To combat overfitting and enrich the training data’s variability, extensive data augmentation techniques were applied. These included horizontal and vertical flips, random rotations up to ±30°, zooming, cropping, brightness and contrast adjustments, and the addition of Gaussian noise—all implemented using TensorFlow and Keras augmentation pipelines.

These transformations simulated real-world variations in clinical settings, such as changes in camera angle, lighting, and lesion presentation, thereby enhancing the model's robustness. Once augmentation was complete, the dataset was converted into optimized TensorFlow Dataset objects. The pipeline included shuffling the data to ensure random sampling, batching it into groups of 32, and enabling prefetching for efficient asynchronous loading during training. This comprehensive preprocessing strategy ensured standardized, enriched, and efficiently managed data input, laying a strong foundation for building a high-performing deep learning model for automated psoriasis detection and severity classification.

2.5 Model Implementation

To determine the most effective deep learning architecture for the detection of acne, eczema, and psoriasis, a comparative evaluation was conducted using three distinct models. Each model was assessed based on its architecture, training enhancements, classification accuracy, precision, and recall. The goal was to identify a model that could accurately distinguish between normal skin and skin affected by acne, eczema, and psoriasis, while balancing computational efficiency, generalizability, and sensitivity to subtle dermatological features. This comparative analysis aimed to establish a robust and scalable system capable of accurately detecting and classifying these common skin diseases. The following sections provide a detailed breakdown of how each model performed in the context of medical image analysis for acne, eczema, and psoriasis detection.

2.5.1 Acne Detection Model Architecture

Six deep learning architectures, VGG16, CNN, VGG19, EfficientNetB0, Siamese network, and ResNet were rigorously evaluated for the acne detection task, with the optimal model being selected based on comparative performance analysis.

2.5.1.1 CNN Model (Best Model)

To address the task of acne severity classification, a deep Convolutional Neural Network (CNN) model was developed from scratch. The architecture was designed to efficiently extract hierarchical features from input facial images, facilitating robust classification.

The model comprises multiple blocks of 2D convolutional layers followed by max-pooling operations. Each convolutional layer utilizes a kernel size of 3×3 with ReLU activation functions. Batch normalization and L2 regularization are applied to reduce overfitting and ensure better generalization. As the network deepens, the number of filters doubles, allowing the model to learn increasingly complex patterns. A sequence of fully connected layers follows the convolutional base, incorporating dropout regularization to combat overfitting. The final layer uses a sigmoid activation function, suitable for binary classification tasks.

The model was trained for up to 70 epochs using the Adam optimizer with a learning rate of 0.0001. EarlyStopping was employed with a patience of 10 epochs, monitoring the validation loss and restoring the best weights accordingly. The model demonstrated excellent convergence, with training accuracy reaching above 98% and validation accuracy peaking at approximately 93.95%.

```
model_cnn = Sequential()

model_cnn.add(tf.keras.layers.Conv2D(filters=32,kernel_size=3,padding='same',
                                     activation='relu',input_shape=[224,224,3],kernel_regularizer=tf.keras.regularizers.l1(0.001)))
model_cnn.add(tf.keras.layers.Conv2D(filters=32,kernel_size=3,activation='relu'))
model_cnn.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))

model_cnn.add(tf.keras.layers.Conv2D(filters=64,kernel_size=3,padding='same',
                                     activation='relu',kernel_regularizer=tf.keras.regularizers.l1(0.001)))
model_cnn.add(tf.keras.layers.Conv2D(filters=64,kernel_size=3,activation='relu'))
model_cnn.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))

model_cnn.add(tf.keras.layers.Conv2D(filters=128,kernel_size=3,padding='same',
                                     activation='relu',kernel_regularizer=tf.keras.regularizers.l1(0.001)))
model_cnn.add(tf.keras.layers.Conv2D(filters=128,kernel_size=3,activation='relu'))
model_cnn.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))

model_cnn.add(tf.keras.layers.Conv2D(filters=256,kernel_size=3,padding='same',
                                     activation='relu',kernel_regularizer=tf.keras.regularizers.l1(0.001)))
model_cnn.add(tf.keras.layers.Conv2D(filters=256,kernel_size=3,activation='relu'))
model_cnn.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))

model_cnn.add(tf.keras.layers.Conv2D(filters=512,kernel_size=3,padding='same',
                                     activation='relu',kernel_regularizer=tf.keras.regularizers.l1(0.001)))
model_cnn.add(tf.keras.layers.Conv2D(filters=512,kernel_size=3,activation='relu'))
model_cnn.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))
```

Figure 16: CNN Model Architecture p 1

Initial convolutional and max pooling layers. The model begins with two 2D convolutional layers with 32 filters and gradually increases to 64 filters. Padding is set to 'same' to maintain spatial dimensions.

```
model_cnn.add(tf.keras.layers.Dropout(0.25))

model_cnn.add(tf.keras.layers.Flatten())

model_cnn.add(tf.keras.layers.Dense(units=1500,activation='relu'))

model_cnn.add(tf.keras.layers.Dropout(0.4)) #To avoid overfitting

#Output Layer
model_cnn.add(tf.keras.layers.Dense(units=1,activation='sigmoid'))
```

Figure 17: CNN Architecture p 2

Deeper convolutional layers with increasing filter sizes (up to 512), followed by dense and dropout layers to flatten and regularize the feature maps before classification.Implementation of EarlyStopping to halt training when validation loss stops improving. The model uses binary crossentropy loss and tracks accuracy during training.

```
Epoch 3/70
52/52 ━━━━━━━━━━━━━━━━━━━━ 265s 5s/step - accuracy: 0.8805 - loss: 13.8195 - val_accuracy: 0.9060 - val_loss: 10.7192
Epoch 4/70
52/52 ━━━━━━━━━━━━━━━━━━━━ 268s 5s/step - accuracy: 0.8965 - loss: 9.8720 - val_accuracy: 0.9181 - val_loss: 7.4775
Epoch 5/70
52/52 ━━━━━━━━━━━━━━━━━━━━ 268s 5s/step - accuracy: 0.9090 - loss: 6.8621 - val_accuracy: 0.9108 - val_loss: 5.2473
Epoch 6/70
52/52 ━━━━━━━━━━━━━━━━━━━━ 559s 11s/step - accuracy: 0.9212 - loss: 4.8591 - val_accuracy: 0.9253 - val_loss: 3.9330
Epoch 7/70
52/52 ━━━━━━━━━━━━━━━━━━━━ 194s 4s/step - accuracy: 0.9359 - loss: 3.6741 - val_accuracy: 0.9301 - val_loss: 3.1426
Epoch 8/70
52/52 ━━━━━━━━━━━━━━━━━━━━ 179s 3s/step - accuracy: 0.9168 - loss: 2.9733 - val_accuracy: 0.9060 - val_loss: 2.6186
Epoch 9/70
52/52 ━━━━━━━━━━━━━━━━━━━━ 176s 3s/step - accuracy: 0.9374 - loss: 2.4610 - val_accuracy: 0.9229 - val_loss: 2.1823
Epoch 10/70
52/52 ━━━━━━━━━━━━━━━━━━━━ 175s 3s/step - accuracy: 0.9367 - loss: 2.0866 - val_accuracy: 0.9205 - val_loss: 1.9065
Epoch 11/70
52/52 ━━━━━━━━━━━━━━━━━━━━ 174s 3s/step - accuracy: 0.9440 - loss: 1.8057 - val_accuracy: 0.9325 - val_loss: 1.6872
Epoch 12/70
52/52 ━━━━━━━━━━━━━━━━━━━━ 179s 3s/step - accuracy: 0.9402 - loss: 1.6034 - val_accuracy: 0.9181 - val_loss: 1.5237
Epoch 13/70
...
Epoch 69/70
52/52 ━━━━━━━━━━━━━━━━━━━━ 167s 3s/step - accuracy: 0.9703 - loss: 0.2829 - val_accuracy: 0.9398 - val_loss: 0.4412
Epoch 70/70
52/52 ━━━━━━━━━━━━━━━━━━━━ 169s 3s/step - accuracy: 0.9834 - loss: 0.2538 - val_accuracy: 0.9325 - val_loss: 0.4502
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Figure 18: Training Progress and Metrics

Training configuration indicating a maximum of 70 epochs with EarlyStopping and validation data specified.

Snapshot of training metrics over epochs showing accuracy and loss for both training and validation datasets. The model achieved strong generalization with validation accuracy close to 94%.

2.5.1.2 VGG16

In addition to the custom CNN architecture, transfer learning was explored using the VGG16 model pre-trained on the ImageNet dataset. VGG16 is a deep convolutional neural network known for its simplicity and effectiveness, making it a suitable backbone for feature extraction in image classification tasks.

To adapt VGG16 for binary acne detection, the model was imported without its top classification layer (include_top=False) and the input shape was set to 224 × 224 × 3 to match the dataset dimensions. Fine-tuning was applied selectively by freezing the initial 10 layers, allowing the deeper layers to be retrained on the target acne

dataset. This approach helps retain low-level features while adapting high-level features to the new domain.

The output from the VGG16 base was flattened and passed through two fully connected dense layers with 256 and 128 units, respectively, interleaved with batch normalization and dropout layers. This additional architecture supports improved generalization and mitigates overfitting. A final sigmoid activation layer was used for binary classification.

For optimization, the Adam optimizer with a learning rate of 1×10^{-4} was selected, and binary cross-entropy was used as the loss function. To further improve training dynamics, a ReduceLROnPlateau callback was used to reduce the learning rate when validation loss plateaued, ensuring fine-grained tuning.

The model was trained for 100 epochs with early stopping and dynamic learning rate scheduling. It achieved high accuracy with strong convergence and stability in validation performance.

```
# 3. Architecture Optimization
model_vgg16 = Sequential([
    base_model,
    Flatten(),
    Dense(256, activation='relu'),  # Increased units
    BatchNormalization(),           # Added Batch Normalization
    Dropout(0.3),                   # Adjusted dropout rate
    Dense(128, activation='relu'),  # Added another Dense layer
    BatchNormalization(),           # Added Batch Normalization
    Dropout(0.2),                   # Added dropout
    Dense(1, activation='sigmoid')
])
```

```
model_vgg16.compile(optimizer=Adam(learning_rate=1e-4),  # Higher learning rate for new layers
                    loss='binary_crossentropy',
                    metrics=['accuracy'])
```

Figure 19: VGG16 - Based Model Architecture

2.5.1.3 VGG19

In addition to the custom CNN architecture, transfer learning was explored using the VGG16 model pre-trained on the ImageNet dataset. VGG16 is a deep convolutional neural network known for its simplicity and effectiveness, making it a suitable backbone for feature extraction in image classification tasks.

To adapt VGG16 for binary acne detection, the model was imported without its top classification layer (include_top = False) and the input shape was set to 224×224×3 to match the dataset dimensions. Fine-tuning was applied selectively by freezing the initial 10 layers, allowing the deeper layers to be retrained on the target acne dataset. This approach helps retain low-level features while adapting high-level features to the new domain.

The output from the VGG16 base was flattened and passed through two fully connected dense layers with 256 and 128 units, respectively, interleaved with batch normalization and dropout layers. This additional architecture supports improved generalization and mitigates overfitting. A final sigmoid activation layer was used for binary classification.

For optimization, the Adam optimizer with a learning rate of 1×10^{-4} was selected, and binary cross-entropy was used as the loss function. To further improve training dynamics, a ReduceLROnPlateau callback was used to reduce the learning rate when validation loss plateaued, ensuring fine-grained tuning.

The model was trained for 100 epochs with early stopping and dynamic learning rate scheduling. It achieved high accuracy with strong convergence and stability in validation performance.

```
base_model_vgg19 = VGG19(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
base_model_vgg19.trainable = False
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg19/vgg19_weights_tf_dim_ordering_tf_kernels_notop.h5
80134624/80134624 ━━━━━━━━━━━━━━━━━━━━ 160s 2us/step
```

```
model_vgg19 = Sequential([
    base_model_vgg19,
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.2),
    Dense(1, activation='sigmoid')
])
```

```
model_vgg19.compile(optimizer=Adam(), loss='binary_crossentropy', metrics=['accuracy'])
model_vgg19.summary()
```

Figure 20: VGG19 Based model architecture for acne classification

2.5.1.4 ResNet

To further evaluate the performance of deep transfer learning models on acne classification, the ResNet50 architecture was utilized. ResNet50, known for its deep residual learning framework, allows efficient training of very deep networks by using skip connections to mitigate vanishing gradients.

The pre-trained ResNet50 model was loaded with ImageNet weights and its top classification layers were excluded (include_top=False). To preserve the learned features, the entire base model was frozen (trainable=False). This ensures that only the newly added layers are trained on the acne dataset.

A simple classifier head was constructed on top of ResNet50. It includes a Flatten layer to transform the output feature maps into a 1D vector, followed by a dense layer with 128 neurons and a ReLU activation function. A dropout layer with a rate

of 0.2 was added for regularization, and the final layer consisted of a single neuron with sigmoid activation to perform binary classification.

The model was compiled using the Adam optimizer and trained for 50 epochs on the preprocessed acne dataset. Initially, the model exhibited moderate performance, but it showed significant improvement across epochs, achieving validation accuracy exceeding 93% by the final epoch.

Model: "sequential_3"

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 7, 7, 2048)	23,587,712
flatten_3 (Flatten)	(None, 100352)	0
dense_7 (Dense)	(None, 128)	12,845,184
dropout_5 (Dropout)	(None, 128)	0
dense_8 (Dense)	(None, 1)	129

Total params: 36,433,025 (138.98 MB)

Trainable params: 12,845,313 (49.00 MB)

Non-trainable params: 23,587,712 (89.98 MB)

Figure 21: Resnet50 Model architecture

2.5.1.5 EfficientNetB0

EfficientNetB0 is a state-of-the-art convolutional neural network architecture that balances network depth, width, and resolution using a compound scaling method. Its efficiency and competitive performance on a variety of tasks make it a strong candidate for acne classification using transfer learning.

In this implementation, the EfficientNetB0 model pre-trained on ImageNet was used with the top classification layers removed (include_top=False). The model was set to non-trainable (trainable=False), serving purely as a feature extractor. This ensures the base model retains the generalized features it learned on a large-scale dataset.

A custom classifier head was added, consisting of a Flatten layer, a dense layer with 128 units using the ReLU activation function, and a dropout layer with a rate of 0.2 for regularization. A final dense layer with a sigmoid activation function completes the model for binary classification.

The model was compiled using the Adam optimizer and binary_crossentropy loss function, and trained over 40 epochs using the training and validation datasets. However, unlike other architectures tested, EfficientNetB0 showed stagnation in both training and validation performance. The model plateaued early with validation accuracy stuck around 51%, indicating potential underfitting or incompatibility without further tuning.

```
base_model_efficientnet = EfficientNetB0(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
base_model_efficientnet.trainable = False
```

```
Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb0_notop.h5
16705208/16705208 ━━━━━━━━━━━━━━━━━━━━ 14s 1us/step
```

```
model_efficientnet = Sequential([
    base_model_efficientnet,
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.2),
    Dense(1, activation='sigmoid')
])
```

```
model_efficientnet.compile(optimizer=Adam(), loss='binary_crossentropy', metrics=['accuracy'])
model_efficientnet.summary()
```

Figure 22: EfficientNetB0 Model architecture

2.5.1.6 Siamese Neural Networks

The Siamese Neural Network (SNN) offers a unique architecture especially well-suited for tasks requiring similarity learning, such as comparing the severity of acne between pairs of facial images. Unlike traditional classification models, a Siamese network learns to differentiate between pairs of inputs by assessing their similarity rather than assigning absolute class labels.

Dataset Preparation

The dataset was organized with images grouped by class (acne severity levels). Image paths and corresponding labels were extracted and stored in a DataFrame. The dataset was then split into training, validation, and testing sets using a 70/15/15 ratio. Labels were encoded using LabelEncoder and converted to one-hot encoding for compatibility with the loss function.

To train the Siamese network, image pairs were generated:

- Positive pairs consist of two images from the same class.
- Negative pairs consist of two images from different classes.

Each pair was labeled as 1 (similar) or 0 (dissimilar).

Model Architecture

The architecture consists of a shared base network (CNN) used to extract feature embeddings from both input images. The base network includes three convolutional layers with increasing filter sizes (64, 128, and 256), followed by max pooling, flattening, and a final dense layer with 256 units.

Two identical input branches pass through the shared base network. The absolute difference between their outputs is computed using a Lambda layer. This difference is passed through a sigmoid-activated dense layer to predict the similarity score.

Training Strategy

The model was compiled using the Adam optimizer and binary_crossentropy loss function, ideal for binary similarity learning. A custom Callback was implemented to stop training early if validation accuracy exceeded 90%.

The network was trained for 20 epochs. Results demonstrated consistent learning with the final validation accuracy exceeding 91.3%, indicating the model effectively learned to compare image pairs for acne classification.

```python
# Define the Siamese network
input_shape = (IMAGE_SIZE[0], IMAGE_SIZE[1], 3)
base_network = create_base_network(input_shape)

input_a = Input(shape=input_shape)
input_b = Input(shape=input_shape)

output_a = base_network(input_a)
output_b = base_network(input_b)

distance = layers.Lambda(lambda x: tf.abs(x[0] - x[1]))([output_a, output_b])
output = layers.Dense(1, activation='sigmoid')(distance)

siamese_model = Model([input_a, input_b], output)

# Compile the Siamese network
# Adam optimizer and binary cross-entropy loss, which is suitable for binary classification tasks.
siamese_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
WARNING:tensorflow:From e:\sliit\Year 4\Research Project\DEMO\.venv\Lib\site-packages\keras\src\backend\tensc
```

Figure 23: Siamese Network

2.5.2 Acne Severity Classification Model Architecture

2.5.2.1 EfficientNetB0

```python
# EfficientNetB0 Model with Additional Layers
base_model = EfficientNetB0(weights='imagenet', include_top=False, input_shape=(img_height, img_width, 3))
base_model.trainable = True  # Enable fine-tuning
```

```python
model1 = Sequential([
    base_model,
    BatchNormalization(),
    GlobalAveragePooling2D(),
    Dense(512, activation='relu'),
    Dropout(0.5),
    Dense(256, activation='relu'),
    Dropout(0.3),
    Dense(128, activation='relu'),
    Dropout(0.3),
    Dense(4, activation='softmax')
])
```

Figure 24: EfficientNetB0 Model architecture for acne severity

2.5.2.2 CNN

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, BatchNormalization
model4 = Sequential()
# First convolutional block
model4.add(Conv2D(32, (3, 3), padding='same', input_shape=(224, 224, 3), activation='relu'))
model4.add(MaxPooling2D(pool_size=(2, 2)))
model4.add(Dropout(0.3))
# Second convolutional block
model4.add(Conv2D(64, (3, 3), padding='same', activation='relu'))
model4.add(MaxPooling2D(pool_size=(2, 2)))
model4.add(Dropout(0.3))
# Flatten layer
model4.add(Flatten())
# Fully connected layer
model4.add(Dense(64, activation='relu'))
model4.add(Dropout(0.4))  # Increased dropout to prevent overfitting
# Output layer (4 classes, softmax)
model4.add(Dense(4, activation='softmax'))
# Compile the model
model4.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model4.summary()
```

Figure 25: CNN Model architecture for acne severity

2.5.2.3 VGG16

```python
from tensorflow.keras.applications import VGG16
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten

# Define custom callback to stop training once the accuracy reaches 0.9
class MyCallback(Callback):
    def on_epoch_end(self, epoch, logs=None):
        val_accuracy = logs.get("val_accuracy")
        print(f"Validation Accuracy at epoch {epoch + 1}: {val_accuracy}")
        if val_accuracy >= 0.9:
            print("\nReached 90% accuracy! Stopping training...")
            self.model.stop_training = True

# Load the pre-trained VGG16 model
base_model = VGG16(weights="imagenet", include_top=False, input_shape=(IMAGE_SIZE[0], IMAGE_SIZE[1], 3))

# Freeze the convolutional base
base_model.trainable = False

# Build the model
model5 = Sequential([
    base_model,
    Flatten(),
    Dense(256, activation="relu"),
    Dense(num_classes, activation="softmax")
])
```

Figure 26: VGG16 Model architecture for acne severity

2.5.2.4 ResNet50V2

```python
# Data Augmentation (Important for improving model generalization)
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

valid_datagen = ImageDataGenerator(rescale=1./255)  # Only rescaling for validation

train_generator = train_datagen.flow(train_images, train_labels, batch_size=BATCH_SIZE)
validation_generator = valid_datagen.flow(valid_images, valid_labels, batch_size=BATCH_SIZE)

# Model Building (ResNet50V2 with some modifications)
base_model = ResNet50V2(weights="imagenet", include_top=False, input_shape=(IMAGE_SIZE[0], IMAGE_SIZE[1], 3))
base_model.trainable = False  # Start with freezing, fine-tune later
```

```python
model8 = Sequential()
model8.add(base_model)
model8.add(GlobalAveragePooling2D())  # Use GAP for dimensionality reduction
model8.add(BatchNormalization()) # Added Batch Normalization
model8.add(Dense(512, activation='relu')) # Increased Dense layer size
model8.add(Dropout(0.5))  # Added Dropout
model8.add(Dense(NUM_CLASSES, activation='softmax'))

# Compile the model
model8.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Callbacks (Improved)
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True) # More patient
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=3, min_lr=0.00001) # Reduce learning rate
model_checkpoint = ModelCheckpoint("acne_sever_VGG1.keras", monitor='val_accuracy', save_best_only=True) # Save best model

# Train the model with generators and improved callbacks
history = model8.fit(
    train_generator,
    steps_per_epoch=len(train_images) // BATCH_SIZE,
    epochs=EPOCHS,
    validation_data=validation_generator,
    validation_steps=len(valid_images) // BATCH_SIZE,
    callbacks=[early_stopping, reduce_lr, model_checkpoint]
)
```

Figure 27: ResNet50V2 Model architecture for acne severity

2.5.2.5 Siamese with CNN

```python
# Function to create pairs of images and labels
def create_pairs(images, labels, num_pairs=500):
    pairs = []
    pair_labels = []

    num_classes = len(set(labels))
    class_indices = [np.where(labels == i)[0] for i in range(num_classes)]

    for _ in range(num_pairs):
        # Randomly select two images from the same class
        class_idx = random.randint(0, num_classes - 1)
        idx1, idx2 = random.sample(list(class_indices[class_idx]), 2)
        pairs.append([images[idx1], images[idx2]])
        pair_labels.append(1)  # Same class

        # Randomly select two images from different classes
        class_idx1, class_idx2 = random.sample(range(num_classes), 2)
        idx1 = random.choice(class_indices[class_idx1].tolist())
        idx2 = random.choice(class_indices[class_idx2].tolist())
        pairs.append([images[idx1], images[idx2]])
        pair_labels.append(0)  # Different classes

    return np.array(pairs), np.array(pair_labels)

# Create pairs for training, validation, and testing
train_pairs, train_pair_labels = create_pairs(train_images, train_labels_int, num_pairs=NUM_PAIRS)
valid_pairs, valid_pair_labels = create_pairs(valid_images, valid_labels_int, num_pairs=200)
test_pairs, test_pair_labels = create_pairs(test_images, test_labels_int, num_pairs=100)
```

Figure 28: Function to create pairs of images and labels

```python
# Define the base network using a pre-trained ResNet50 model
def create_base_network(input_shape):
    base_model = ResNet50(weights='imagenet', include_top=False, input_shape=input_shape)
    x = base_model.output
    x = layers.GlobalAveragePooling2D()(x)
    x = layers.Dense(256, activation='relu')(x)
    x = layers.BatchNormalization()(x)
    x = layers.Dropout(0.5)(x)
    return Model(base_model.input, x)
```

Figure 29: Base network with pre-trained ResNet50

```python
# Define the Siamese network
input_shape = (IMAGE_SIZE[0], IMAGE_SIZE[1], 3)
base_network = create_base_network(input_shape)

input_a = Input(shape=input_shape)
input_b = Input(shape=input_shape)

output_a = base_network(input_a)
output_b = base_network(input_b)

distance = layers.Lambda(lambda x: tf.abs(x[0] - x[1]))([output_a, output_b])
output = layers.Dense(1, activation='sigmoid')(distance)

siamese_model = Model([input_a, input_b], output)

# Compile the Siamese network
siamese_model.compile(optimizer=Adam(learning_rate=0.0001), loss='binary_crossentropy', metrics=['accuracy'])

# Callbacks for Siamese network training
siamese_callbacks = [
    EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True),
    ModelCheckpoint('best_siamese_model.keras', monitor='val_accuracy', save_best_only=True, mode='max'),
    ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=1e-6),
    TensorBoard(log_dir='./logs/siamese')
]
```

Figure 30: Siamese Network

```python
# Define the multi-class classification model
def create_multi_class_model(base_network, input_shape, num_classes):
    input = Input(shape=input_shape)
    features = base_network(input)
    output = layers.Dense(num_classes, activation='softmax')(features)
    return Model(input, output)

# Create the multi-class classification model
multi_class_model = create_multi_class_model(base_network, input_shape, num_classes)

# Compile the model
multi_class_model.compile(optimizer=Adam(learning_rate=0.0001), loss='categorical_crossentropy', metrics=['accuracy'])
```

```python
# Callbacks
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
model_checkpoint = ModelCheckpoint('best_multi_class_model.keras', monitor='val_accuracy', save_best_only=True, mode='max')
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=1e-6)
tensorboard = TensorBoard(log_dir='./logs', histogram_freq=1)
```

+ Code + Markdown

Figure 31: Multi classification model

2.5.3 Eczema Detection Model Architecture

2.5.3.1 MobileNetV2 Model (Best Model)

To enhance the diagnostic performance of the system, a transfer learning approach was adopted using MobileNetV2 as the base model. MobileNetV2 is a lightweight and efficient convolutional neural network architecture pretrained on ImageNet, known for its performance in real-time applications. This model was fine-tuned and extended with an attention mechanism using Squeeze-and-Excitation (SE) blocks, enabling it to focus more accurately on lesion-specific patterns within the skin. The model was specifically designed to capture both global and localized features of eczema, while maintaining computational efficiency suitable for deployment in mobile or edge devices.

The SE block recalibrates channel-wise feature responses by learning which channels are more relevant for classification. This allows the model to emphasize more informative skin texture features and suppress irrelevant patterns. It works by first applying global average pooling to compress spatial information into a single vector per channel, capturing the overall importance of each feature map. This is followed by a small bottleneck network that learns non-linear interactions and outputs channel-wise weights through a sigmoid activation. These weights are then used to rescale the original feature maps, effectively enabling the model to "attend" to more meaningful features relevant to eczema detection while reducing noise from less significant patterns[28].

```python
def se_block(input_tensor, reduction_ratio=16):
    """
    Implements a Squeeze-and-Excitation (SE) block.
    Args:
        input_tensor: The input tensor from the previous layer.
        reduction_ratio: Reduction ratio for the bottleneck.
    Returns:
        Tensor after applying SE block.
    """
    filters = input_tensor.shape[-1]
    se = layers.GlobalAveragePooling2D()(input_tensor)
    se = layers.Dense(filters // reduction_ratio, activation='relu')(se)
    se = layers.Dense(filters, activation='sigmoid')(se)
    se = layers.Multiply()([input_tensor, se])
    return se
```

Figure 32: Implementation of the Squeeze-and-Excitation (SE) block to reweight important feature channels

The base MobileNetV2 model was loaded with pretrained ImageNet weights, and its convolutional layers were frozen to retain general image features. A global average pooling layer was added, followed by dropout for regularization. The SE block was then inserted to enhance feature representation. A fully connected layer and softmax activation concluded the model for binary classification.

```python
# Build MobileNetV2 model with SE block
def build_model_with_attention(input_shape=(224, 224, 3), num_classes=2):
    base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=input_shape)
    base_model.trainable = False  # Freeze the base model

    x = base_model.output
    x = layers.GlobalAveragePooling2D()(x)
    x = layers.Dropout(0.5)(x)  # Add dropout after pooling

    # Add SE block
    x = layers.Reshape((1, 1, x.shape[-1]))(x)
    x = se_block(x)
    x = layers.Flatten()(x)

    x = layers.Dense(32, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.05))(x)
    x = layers.Dropout(0.5)(x)  # Moderate dropout rate
    output = layers.Dense(num_classes, activation='softmax')(x)

    model = models.Model(inputs=base_model.input, outputs=output)
    model.compile(optimizer=Adam(learning_rate=1e-5), loss='sparse_categorical_crossentropy', metrics=['accuracy'])

    return model
```

Figure 33: Building a MobileNetV2-based model with integrated SE attention block for improved skin condition focus

The model was trained using the augmented dataset, with callbacks for early stopping (to prevent overfitting), dynamic learning rate reduction, and best-model checkpointing. The use of class weights ensured balanced learning despite class imbalance.

```python
# Callbacks for training
early_stopping = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.3, patience=2, min_lr=1e-7)
checkpoint = ModelCheckpoint('eczema_model_with_attention_best.keras', monitor='val_loss', save_best_only=True, mode='min')

# Train the model
history_with_attention = model_with_attention.fit(
    datagen.flow(X_train, y_train, batch_size=batch_size),
    epochs=50,
    validation_data=(X_val, y_val),
    class_weight=class_weight_dict,
    callbacks=[early_stopping, reduce_lr, checkpoint]
)
```

Figure 34: Model training with class weights and early stopping, learning rate scheduling, and checkpointing

The model's performance was evaluated using a confusion matrix and classification report, giving detailed insights into precision, recall, and F1-scores for both eczema and normal classes.

2.5.3.2 CNN Model

A custom CNN with SE attention blocks was implemented for eczema detection, consisting of three convolutional blocks with ReLU activation, max-pooling, and batch normalization. SE blocks enhanced feature focus by recalibrating feature maps through attention mechanisms. The model used sigmoid activation for binary classification, with the Adam optimizer and binary cross-entropy loss. Training over 50 epochs with augmented data, supported by callbacks like early stopping and ReduceLROnPlateau, demonstrated the effectiveness of attention mechanisms despite not outperforming the MobileNetV2-based model.

```python
# Define the model with SE blocks
def create_model_with_attention(input_shape):
    inputs = layers.Input(shape=input_shape)

    # Convolutional layer block 1
    # Extracts important features from the images
    x = layers.Conv2D(32, (3, 3), activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.0005))(inputs)
    x = layers.MaxPooling2D(pool_size=(2, 2))(x)
    x = layers.BatchNormalization()(x)
    x = se_block(x)

    # Convolutional layer block 2
    x = layers.Conv2D(64, (3, 3), activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.0005))(x)
    x = layers.MaxPooling2D(pool_size=(2, 2))(x)
    x = layers.BatchNormalization()(x)
    x = se_block(x)

    # Convolutional layer block 3
    x = layers.Conv2D(128, (3, 3), activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.0005))(x)
    x = layers.MaxPooling2D(pool_size=(2, 2))(x)
    x = layers.BatchNormalization()(x)
    x = se_block(x)

    # Flatten the output before the fully connected layers
    # Converts the 2D feature maps into a 1D vector for the fully connected layers
    x = layers.Flatten()(x)

    # Fully connected layer
    # Performs final classification
    x = layers.Dense(128, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.0005))(x)
    x = layers.Dropout(0.4)(x)
    outputs = layers.Dense(1, activation='sigmoid')(x)

    # Create the model
    model = Model(inputs, outputs)
    return model

# Create the model
model_with_attention = create_model_with_attention((224, 224, 3))
```

Figure 35: Custom CNN model architecture with SE attention blocks embedded after each convolutional layer block

2.5.3.3 ResNet50 Model

A ResNet50-based transfer learning approach was implemented for eczema detection, leveraging its depth and skipping connections to enhance feature extraction. Pretrained on ImageNet, the model was fine-tuned with a custom attention mechanism to emphasize relevant skin regions while downplaying irrelevant areas. The attention layer generated weights via learned transformations applied to feature maps, enhancing sensitivity to eczema-specific patterns. Fine-tuning involved unfreezing deeper layers to adapt pretrained weights for eczema detection. The model was trained using the Adam optimizer with a low learning

rate to facilitate gradual improvement during fine-tuning. To prevent overfitting and optimize learning, EarlyStopping and ReduceLROnPlateau callbacks were implemented.

```python
class Attention(layers.Layer):
    def __init__(self):
        super(Attention, self).__init__()

    def build(self, input_shape):
        self.W = self.add_weight(shape=(input_shape[-1], input_shape[-1]), initializer='random_normal', trainable=True)
        self.b = self.add_weight(shape=(input_shape[-1],), initializer='zeros', trainable=True)

    def call(self, inputs):
        q = tf.matmul(inputs, self.W) + self.b
        attention_weights = tf.nn.softmax(q, axis=-1)
        return attention_weights * inputs
```

Figure 36: Implementation of the attention mechanism as a custom layer

```python
# Build the Model using ResNet50 as the base model and an attention mechanism
def build_model():
    base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(IMG_HEIGHT, IMG_WIDTH, 3))
    x = base_model.output
    x = layers.GlobalAveragePooling2D()(x)

    # Attention Layer
    attention = Attention()(x)

    # Fully Connected Layers
    x = layers.Dense(1024, activation='relu')(attention)
    x = layers.Dropout(0.5)(x)  # Adding dropout for regularization
    x = layers.Dense(1, activation='sigmoid')(x)  # Output layer for binary classification (eczema vs normal)

    model = Model(inputs=base_model.input, outputs=x)

    # Unfreeze more layers of the base model for fine-tuning
    for layer in base_model.layers[:100]:  # Unfreeze more layers than before
        layer.trainable = False
    for layer in base_model.layers[100:]:
        layer.trainable = True

    model.compile(optimizer=Adam(learning_rate=1e-5), loss='binary_crossentropy', metrics=['accuracy'])
    return model
```

Figure 37: ResNet50 model with custom attention mechanism and fine-tuning for eczema detection

2.5.4 Eczema Severity Classification Model Architecture

2.5.4.1 EfficientNetB0 Model (Best Model)

The most effective model identified for eczema severity classification was based on EfficientNetB0, enhanced with a SE attention mechanism. EfficientNetB0 is known for its efficient scaling of depth, width, and resolution, providing a balanced architecture that offers high accuracy with relatively low computational costs. To

further enhance the model's ability to detect subtle variations across different severity levels, a custom SE block was integrated to improve the network's sensitivity towards relevant features.

To address the challenge of class imbalance, Focal Loss and class weight assignment were employed during model training. Focal Loss was used as a customized loss function to ensure that the model focuses more on hard-to-classify samples, which are often from the minority classes. Unlike standard cross-entropy loss, focal loss applies a modulating factor that down-weights easy-to-classify examples, thus effectively reducing their impact on the overall loss. This is achieved by applying a scaling factor $(1-p_t)^\gamma$ to the cross-entropy loss, where p_t is the predicted probability for the correct class. The hyperparameter γ adjusts the rate at which easy examples are down-weighted. Additionally, an α parameter is used to further balance the importance of each class, particularly benefiting underrepresented categories such as moderate and severe eczema.

```
# Focal Loss Function (handle one-hot encoded labels)
def focal_loss(alpha=0.25, gamma=2.0):
    def focal_loss_fixed(y_true, y_pred):
        y_true = tf.cast(y_true, tf.float32)
        y_pred = tf.clip_by_value(y_pred, tf.keras.backend.epsilon(), 1 - tf.keras.backend.epsilon())
        cross_entropy = -y_true * tf.math.log(y_pred)
        focal_loss = alpha * tf.math.pow(1 - y_pred, gamma) * cross_entropy
        return tf.reduce_sum(focal_loss, axis=-1)
    return focal_loss_fixed
```

Figure 38: Implementation of Focal Loss to enhance model sensitivity towards minority classes by applying dynamic scaling to hard-to-classify samples

In addition to using focal loss, class weights were computed and applied during training to further mitigate the effects of class imbalance. By assigning higher weights to minority classes, the model is encouraged to pay more attention to those classes during the learning process. The class weights were calculated using the compute_class_weight function from sklearn, which assigns weights inversely proportional to the class frequencies. This approach ensures that the training process does not disproportionately favor the majority class.

```python
# Compute class weights
class_weights = compute_class_weight('balanced', classes=np.unique(y_train), y=y_train)
class_weight_dict = {i: class_weights[i] for i in range(len(class_weights))}
```

Figure 39: Calculating class weights to assign higher importance to underrepresented classes and promote balanced learning during training

The SE block was implemented to enhance the model's attention mechanism by recalibrating feature maps at the channel level. This improves the network's ability to focus on essential features, particularly those that differentiate between mild, moderate, and severe eczema cases.

```python
# Squeeze-and-Excitation block
def se_block(input_tensor, ratio=16):
    filters = input_tensor.shape[-1]
    se = tf.keras.layers.GlobalAveragePooling2D()(input_tensor)
    se = tf.keras.layers.Dense(filters // ratio, activation='relu')(se)
    se = tf.keras.layers.Dense(filters, activation='sigmoid')(se)
    se = tf.keras.layers.Reshape([1, 1, filters])(se)
    return tf.keras.layers.Multiply()([input_tensor, se])
```

Figure 40: Implementation of the Squeeze-and-Excitation (SE) block to enhance feature relevance through channel-wise recalibration

EfficientNetB0 was used as the base model, pretrained on ImageNet. The model was modified by adding a SE block, followed by fully connected layers to enhance the learning capacity for severity classification. A softmax activation function was used for the final layer to support multi-class classification.

```python
# Build Enhanced EfficientNetB0 Model with Attention Mechanism
def build_efficientnet_model_with_attention(input_shape=(224, 224, 3), num_classes=3):

    base_model = EfficientNetB0(weights="imagenet", include_top=False, input_shape=input_shape)
    base_model.trainable = True

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

    x = se_block(x)

    x = GlobalAveragePooling2D()(x)
    x = BatchNormalization()(x)

    x = Dense(512, activation="relu", kernel_regularizer=l2(0.01))(x)
    x = Dropout(0.4)(x)

    x = Dense(256, activation="relu", kernel_regularizer=l2(0.01))(x)
    x = Dropout(0.3)(x)

    outputs = Dense(num_classes, activation="softmax")(x)

    model = Model(inputs, outputs)
    return model

# Compile the Model
model = build_efficientnet_model_with_attention()
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
    loss=focal_loss(alpha=0.25, gamma=2.0),
    metrics=["accuracy"]
)
```

Figure 41: Building the EfficientNetB0 model integrated with SE block and additional dense layers for classification

The model was compiled using Focal Loss to address class imbalance and trained using the Adam optimizer. To improve training efficiency, a cosine decay learning rate scheduler was applied along with early stopping and model checkpointing.

2.5.4.2 CNN Model

A custom model was developed to classify eczema severity into three categories, mild, moderate, and severe, using Residual Blocks integrated with SE attention mechanisms. SE attention blocks within these Residual Blocks recalibrate feature maps by prioritizing relevant channels, improving sensitivity to varying severity levels. The final architecture includes Global Average Pooling, Dense layers, and

Dropout for regularization, aiming to enhance robustness and accuracy in distinguishing eczema severity.

```python
# Squeeze-and-Excitation Block
def se_block(input_tensor, ratio=16):
    filters = input_tensor.shape[-1]
    se = GlobalAveragePooling2D()(input_tensor)
    se = Dense(filters // ratio, activation='relu')(se)
    se = Dense(filters, activation='sigmoid')(se)
    se = Reshape((1, 1, filters))(se)
    return Multiply()([input_tensor, se])

# Residual Block with Attention
def residual_block(x, filters, kernel_size=3, se_ratio=16):
    shortcut = Conv2D(filters, (1, 1), padding='same')(x)
    x = Conv2D(filters, kernel_size, padding='same', activation='relu')(x)
    x = BatchNormalization()(x)
    x = Conv2D(filters, kernel_size, padding='same', activation='relu')(x)
    x = BatchNormalization()(x)
    x = se_block(x, ratio=se_ratio)
    x = Add()([x, shortcut])
    return x
```

```python
# Build Enhanced Model
def build_enhanced_model(input_shape=(224, 224, 3), num_classes=3):
    inputs = Input(shape=input_shape)

    # Initial Convolutional Block
    x = Conv2D(64, (3, 3), padding='same', activation='relu')(inputs)
    x = BatchNormalization()(x)
    x = MaxPooling2D((2, 2))(x)
    x = Dropout(0.3)(x)

    # Residual Blocks with SE Attention
    x = residual_block(x, 128)
    x = MaxPooling2D((2, 2))(x)
    x = Dropout(0.4)(x)

    x = residual_block(x, 256)
    x = MaxPooling2D((2, 2))(x)
    x = Dropout(0.4)(x)

    x = residual_block(x, 512)
    x = MaxPooling2D((2, 2))(x)
    x = Dropout(0.5)(x)

    # Classification Block
    x = GlobalAveragePooling2D()(x)
    x = Dense(512, activation='relu')(x)
    x = Dropout(0.5)(x)
    outputs = Dense(num_classes, activation='softmax')(x)

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

Figure 42: Building a custom CNN model with Residual Blocks, SE blocks, and fully connected layers for classification

2.5.4.3 ResNet50 Model

A ResNet50-based model enhanced with the Convolutional Block Attention Module (CBAM) was developed to classify eczema severity into mild, moderate, and severe categories. The CBAM module combines channel and spatial attention

mechanisms to improve the model's ability to focus on relevant features while suppressing background noise. Pretrained on ImageNet, the ResNet50 architecture was fine-tuned for eczema severity classification, with CBAM blocks integrated to enhance feature extraction by adaptively prioritizing important channels and spatial regions. This approach aimed to improve accuracy and sensitivity in distinguishing varying eczema severity levels.

```
# Define CBAM (Channel and Spatial Attention Block)
def cbam_block(input_tensor, reduction_ratio=8):
    """
    Convolutional Block Attention Module (CBAM)
    Combines Channel and Spatial Attention.
    """
    channels = input_tensor.shape[-1]

    # Channel Attention
    channel_avg = GlobalAveragePooling2D()(input_tensor)
    channel_max = GlobalMaxPooling2D()(input_tensor)
    channel = Concatenate()([channel_avg, channel_max])
    channel = Dense(channels // reduction_ratio, activation='relu')(channel)
    channel = Dense(channels, activation='sigmoid')(channel)
    channel = Reshape((1, 1, channels))(channel)
    channel_attention = Multiply()([input_tensor, channel])

    # Spatial Attention
    def spatial_attention(inputs):
        spatial_avg = tf.reduce_mean(inputs, axis=-1, keepdims=True)
        spatial_max = tf.reduce_max(inputs, axis=-1, keepdims=True)
        return tf.concat([spatial_avg, spatial_max], axis=-1)

    spatial = Lambda(spatial_attention)(channel_attention)
    spatial = Conv2D(1, kernel_size=7, activation='sigmoid', padding='same')(spatial)
    spatial_attention = Multiply()([channel_attention, spatial])

    return spatial_attention
```

Figure 43: Implementation of the CBAM block combining channel and spatial attention mechanisms to enhance feature relevance

The CBAM block, integrated with a ResNet50-based model, enhances eczema severity classification by refining feature maps through Channel and Spatial Attention mechanisms. Channel attention aggregates spatial information using global average and max pooling to produce a channel-wise attention map, while spatial attention applies convolution over concatenated average-pooled and max-pooled maps to generate spatial attention weights. The ResNet50 backbone's lower layers were initially frozen to retain pretrained weights, with fine-tuning applied to

deeper layers for task-specific learning. The CBAM block was added post-ResNet50 output to improve feature relevance before classification, enhancing sensitivity to eczema severity levels.

```
# Load ResNet50 base model
base_model = applications.ResNet50(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Fine-tuning: Freeze some layers
for layer in base_model.layers[:100]:
    layer.trainable = False
for layer in base_model.layers[100:]:
    layer.trainable = True

# Add custom layers with CBAM and Dropblock
x = base_model.output
x = cbam_block(x)  # Add CBAM Attention
x = Dropout(0.3)(x)  # Apply Dropout to convolutional features
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu', kernel_regularizer=l2(0.01))(x)
x = Dropout(0.5)(x)
x = Dense(512, activation='relu', kernel_regularizer=l2(0.01))(x)
x = Dropout(0.3)(x)
output = Dense(3, activation='softmax')(x)  # 3 classes: mild, moderate, severe

# Compile the model
model = tf.keras.models.Model(inputs=base_model.input, outputs=output)
model.compile(
    optimizer=tf.keras.optimizers.AdamW(learning_rate=1e-4, weight_decay=1e-6),
    loss=tf.keras.losses.CategoricalCrossentropy(),
    metrics=['accuracy']
)
```

Figure 44: Building the ResNet50-based model with integrated CBAM block and custom dense layers

2.5.5 Psoriasis Detection Model Architecture

2.5.5.1 YOLO Model (Best Model)

The YOLOv8 object detection model was adopted for psoriasis lesion detection, selected for its exceptional performance, speed, and real-time inference capabilities. The model was initialized with pre-trained weights from the COCO dataset and fine-tuned on a custom psoriasis dataset, where lesion regions were manually annotated with bounding boxes using the CVAT (Computer Vision Annotation Tool). Model configuration and training parameters were specified via a YAML configuration file, with training conducted over 100 epochs using an input resolution of 640×640 pixels and a batch size of 16.

It lightweight and modular architecture, YOLOv8 enabled efficient training and deployment in GPU-accelerated environments while delivering high accuracy in detecting lesion boundaries. The model demonstrated strong generalization and precise localization of psoriasis-affected areas, even with a limited dataset. These advantages made YOLOv8 an optimal choice for seamless integration into the DermaScope AI system, enabling reliable and real-time lesion detection in clinical and teledermatology applications.

```
# 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]
erriding 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]
```

Figure 45: Train the YOLO Model

2.5.5.2 CNN Model

A custom Convolutional Neural Network (CNN) was designed for binary classification of skin images as either psoriasis-affected or normal. The model included three convolutional layers with increasing filters (32, 64, 128), each followed by max pooling and dropout to reduce overfitting. The output was flattened and passed through a dense layer with ReLU activation and a dropout layer, followed by a softmax layer for final prediction. This lightweight architecture served as an effective baseline for psoriasis detection from image data.

```
# 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 46: Custom CNN architecture for classifying psoriasis vs. normal skin

2.5.5.3 EfficientNetB0 Model

An advanced psoriasis classification model was built using the EfficientNetV2B0 architecture as the backbone. Through transfer learning, all layers were unfrozen to fine-tune the model for psoriasis detection. The extracted features were passed through a Flatten layer, followed by dense layers with ReLU activation, dropout, and L2 regularization to reduce overfitting. A softmax layer was used for final prediction of normal or psoriasis-affected skin. The model offered an effective balance of accuracy and efficiency, making it suitable for real-time clinical use.

```
[ ] # 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 47: EfficientNetV2B0 model architecture for binary classification of normal and psoriasis affected skin

2.5.5.4 ResNet50 Model

For psoriasis detection, a model was developed using ResNet50 as the feature extractor, utilizing its deep residual connections for improved image representation. The pre-trained model was used without the top layers, followed by a Flatten layer, a dense ReLU layer, dropout for regularization, and a softmax output for classifying normal vs. psoriasis-affected skin. ResNet50's rich feature hierarchy made it effective for capturing complex dermatological patterns in lesion images.

```
# 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
```

```
# 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 48: ResNet50-based model for binary skin classification of normal and psoriasis-affected skin

2.5.5.5 InceptionResNetV2 Model

For psoriasis detection, InceptionResNetV2 was used as the base model, combining Inception modules with residual connections for efficient feature extraction. Initialized with ImageNet weights, the model was fine-tuned using transfer learning. Custom layers including global average pooling, dropout, and dense layers were added for classifying normal and psoriasis-affected skin. This architecture effectively captures both global and local features, making it suitable 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 49: InceptionResNetV2 model for binary skin classification of normal and psoriasis-affected skin

2.5.6 Psoriasis Severity Classification Model Architecture

2.5.6.1 MobileNetV2 (Best Model)

For the psoriasis severity classification task, a customized MobileNetV2-based model was implemented, enhanced with Ghost Modules and a MixConv block for efficient and multi-scale feature extraction.

```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 50: GhostModule implementation

The base MobileNetV2, pre-trained on ImageNet, was partially frozen to retain general features while fine-tuning the top layers for psoriasis-specific patterns. A MixConv block enabled better lesion texture and shape recognition through different kernel sizes.

```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 51: MixConv block implementation

This was followed by Ghost Modules that reduced redundancy and computational cost while preserving critical features. The architecture concluded with global average pooling, batch normalization, dropout, and a softmax layer for multi-class classification into mild, moderate, and severe psoriasis categories. This lightweight and scalable design is well-suited for real-time clinical deployment.

```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 52: Full MobileNetV2 model with integrated blocks

2.5.6.2 CNN Model

A baseline CNN model was developed using a simple sequential architecture to serve as a reference for psoriasis severity classification. It included two convolutional layers with ReLU activation and max pooling to extract features and reduce spatial dimensions. After flattening, the model had a dense layer, dropout for regularization, and a softmax layer to classify images into mild, moderate, and severe categories. This straightforward design provided a benchmark for evaluating more advanced models.

```
[ ] # 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 53: Baseline CNN model used for psoriasis severity classification

2.5.6.2 EfficientNetB3 Model

A customized psoriasis severity classification model was built using EfficientNetB3, enhanced with a MixConv block and Ghost Modules for improved feature extraction and efficiency. The model leverages pre-trained weights with fine-tuned top layers to capture psoriasis-specific patterns. Ghost Modules help generate lightweight yet informative feature maps, while the MixConv block supports multi-scale detection of lesion characteristics. This architecture is tailored for high-resolution 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 54: Customized EfficientNetB3-based model architecture for psoriasis severity classification

2.6 Commercialization Aspects of the Product

The commercialization of the proposed system, DermaScope AI, aims to establish a cutting-edge solution for the accurate detection, severity assessment, and management of acne, eczema, and psoriasis using advanced machine learning techniques. The primary focus is to deliver a robust, user-friendly, and clinically meaningful tool that can be adopted by dermatologists, healthcare institutions, telemedicine platforms, and individual patients for real-time monitoring and diagnosis.

2.6.1 Market Potential

- Dermatology Clinics and Hospitals: Offering enhanced diagnostic accuracy and consistency in assessing acne, eczema, and psoriasis, improving patient care and optimizing treatment plans.
- Telemedicine Platforms: Facilitating remote diagnosis and management of skin diseases through AI-based assessments, making dermatological expertise accessible to patients in remote or underserved areas.
- Pharmaceutical Companies: Supporting clinical trials and drug efficacy studies by providing reliable and standardized assessment of skin disease severity over time.
- Healthcare Applications & Wearables: Integrating the solution with mobile health applications or smart devices to allow continuous monitoring and management of skin conditions.
- Cosmetic and Skincare Industry: Assisting product development by providing insights into skin condition improvements based on severity analysis and user feedback.

2.6.2 Role of Skin Disease Detection and Severity Assessment

- Personalized Treatment Plans: Enabling dermatologists to tailor treatment strategies based on precise severity assessment, enhancing treatment efficacy and patient satisfaction.
- Patient Monitoring: Allowing patients to track their progress over time and providing alerts or recommendations when conditions worsen or improve.
- Data-Driven Decisions: Leveraging collected data to enhance treatment protocols, refine AI models, and optimize patient outcomes.

2.6.3 Commercialization Strategy

- Subscription-Based Model: Offering a tiered subscription service for dermatology clinics, telemedicine platforms, and individual users. Plans may include basic diagnostic features, advanced severity assessment, and comprehensive analytics tools.
- Software as a Service (SaaS): Providing a cloud-based solution accessible through web or mobile applications, ensuring ease of deployment and maintenance for clients.
- Licensing Agreements: Partnering with healthcare institutions, telemedicine providers, and pharmaceutical companies for exclusive or co-branded solutions.
- Customization Services: Offering customized solutions tailored to the needs of specific clients, including unique dashboards, analytics modules, and integration with existing health management systems.

2.6.4 Competitive Advantage

- Comprehensive Solution: Combining image analysis, severity assessment, and patient monitoring in a unified platform that supports acne, eczema, and psoriasis.
- Explainable AI (XAI): Providing interpretable results that enhance clinician trust and improve diagnostic transparency.
- Scalability: The system is designed to be scalable, allowing integration with various healthcare platforms, accommodating large datasets, and evolving to include more skin diseases.
- Accessibility: Making dermatological assessment available remotely through telemedicine applications, benefiting underserved regions and enhancing patient convenience.

2.6.5 Scalability and Future Development

- Scalability: The application will be scalable with respect to the number of users, integration with healthcare platforms, and adaptation for additional skin diseases beyond acne, eczema, and psoriasis.
- Continuous Improvement: Leveraging ongoing research and client feedback to enhance AI models, improve diagnostic accuracy, and expand the scope of the system.
- Integration with Wearables: Exploring partnerships with wearable technology companies to provide continuous monitoring and real-time feedback for chronic skin conditions.
- Expanding Market Reach: Developing marketing strategies targeting healthcare providers, pharmaceutical companies, and telemedicine platforms worldwide.

A well-defined commercialization strategy will position DermaScope AI as a flagship solution in dermatology, bridging the gap between cutting-edge AI technology and practical clinical applications. Through strategic partnerships, scalable solutions, and continuous innovation, the product aims to establish itself as a leader in dermatological diagnostics and management.

2.7 Testing and Implementation

2.7.1 Functional Requirements

- **Acne Detection and Severity Classification:**
  - The system enables users to upload high-resolution images of affected skin areas via a web-based interface.
  - Multiple models were tested, including CNN, VGG16, VGG19, ResNet50, EfficientNetB0, and Siamese Networks. Among these, CNN provided the best balance between accuracy and computational efficiency.
  - For severity grading, the system uses the best-performing CNN architecture after extensive hyperparameter tuning, achieving high accuracy on test data.
  - Outputs include binary classification (acne vs non-acne) and severity level (mild, moderate, severe, very severe), with corresponding confidence scores.
  - All results are presented in a structured report format with timestamped entries.

- **Eczema Detection and Severity Classification:**
  - The detection module distinguishes between normal and eczema-affected skin using ResNet50 and EfficientNetB0 models.
  - A dedicated severity assessment model classifies eczema into mild, moderate, and severe, trained using dermatologist-labeled data.

- Data augmentation techniques and class balancing methods were applied to counteract dataset imbalance.
- Outputs include binary classification (eczema vs non-eczema) and severity level (mild, moderate, severe), with corresponding confidence scores.
- All results are presented in a structured report format with timestamped entries.

- **Psoriasis Detection and Severity Classification:**
  - YOLOv8 was selected for lesion detection due to its superior object detection accuracy and speed.
  - Severity classification relies on MobileNetV2 enhanced with MixConv and GhostModule blocks.
  - Data augmentation techniques were applied solely to improve generalization and prevent overfitting.
  - Outputs include binary classification (psoriasis vs non-psoriasis) and severity level (mild, moderate, severe), with corresponding confidence scores.
  - All results are presented in a structured report format with timestamped entries.
- **User Interaction and Dashboard:**
  - Patients and doctors access a dashboard to manage profiles, view image history, and analyze results.
  - Each image analysis includes the disease type, detection status, severity level, and prediction timestamp.
  - A progress tracking system shows historical severity trends for recurring diseases.

2.7.2 Non-Functional Requirements

- **Performance:**
  - Real-time model inference completed within 2–4 seconds on GPU-enabled devices.
  - Backend endpoints are optimized with caching strategies for faster response.
- **Scalability:**
  - Cloud-based infrastructure supports concurrent users and growing datasets.
- **Security:**
  - End-to-end encryption is applied to all image transmissions.
  - Authentication layers are in place to secure user and clinician access.
- **Usability:**
  - Tooltips, guides, and visual cues assist users in uploading images and interpreting results.
  - Designed to support both technical and non-technical users.
- **Accessibility:**
  - The frontend supports screen readers, keyboard navigation, and high-contrast display options.
- **Extensibility:**
  - The backend is modular to facilitate easy addition of other dermatological disease models.

- **Robustness:**
  - Model outputs include fallback predictions when confidence is low.
- **Maintainability:**
  - Codebase adheres to modular and reusable design practices with documentation and version control.

2.7.3 Backend Implementation

- Flask-based backend handles routing, model inference, and database communication.

```
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"]
    }
})

bcrypt = Bcrypt(app)

db_pool = pool.SimpleConnectionPool(
    1, 20,
    user=os.getenv('DB_USER'),
    password=os.getenv('DB_PASSWORD'),
    host=os.getenv('DB_HOST'),
    port=os.getenv('DB_PORT'),
    database=os.getenv('DB_NAME')
)
```

Figure 55: Flask app setup with CORS and PostgreSQL connection pool initialization

- Models are loaded for detection and severity classification tasks. And YOLOv8 is loaded for real-time psoriasis lesion detection.

```
#eczema_detection_model = tf.keras.models.load_model('./models/eczema/Eczema_Detection_Model_with_Attention.h5', compile=False)
eczema_detection_model = tf.keras.models.load_model('./models/eczema/Eczema_Detection_Model_CNN.h5',compile=False)
acne_severity_model = tf.keras.models.load_model('./models/acne/best_acne_severe.h5',compile=False)
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)
eczema_severity_model = tf.keras.models.load_model('./models/eczema/Eczema_Severity_Assessment_Model_with_Attention.h5', compile=False)
#psoriasis_severity_model = tf.keras.models.load_model('./models/psoriasis/psoriasis_severity_cnn_model.h5', compile=False)
#psoriasis_severity_model = tf.keras.models.load_model('./models/psoriasis/psoriasis_severity_model_mobilevnet_mix_79_.h5',compile=False)
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 56: Model loading setup for acne, eczema, and psoriasis detection and severity classification using TensorFlow and YOLOv8

- Cloudinary is integrated for secure cloud storage of user-submitted images.

```python
# Save image to cloud storage
image_handler = CloudinaryImageHandler()
image_url = image_handler.save_image(img_file, user_id)

if not image_url:
    return jsonify({'error': 'Failed to save image'}), 500
```

Figure 57: Uploading images to secure cloud storage using the Cloudinary API integration

- PostgreSQL database is used for storing results and patient history.

- API endpoints include:

  - /api/detect/acne, /api/detect/eczema, /api/detect/psoriasis

```python
@app.route('/api/detect/acne', methods=['POST'])
def detect_acne():
    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

        img = Image.open(img_file.stream)
        img = img.convert('RGB')

@app.route('/api/detect/eczema', methods=['POST'])
def detect_eczema():
    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

        img = Image.open(img_file.stream)
        img = img.convert('RGB')

@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)
```

Figure 58: API for acne, eczema, and psoriasis detection

- /api/assess/acne_severity, /api/assess/eczema_severity, /api/assess/psoriasis_severity

```
@app.route('/api/assess/acne_severity', methods=['POST'])
def assess_acne_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')  # Get the result ID from detecti

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

        # Process image for severity assessment
        img = Image.open(img_file.stream)
```

```
@app.route('/api/assess/eczema_severity', methods=['POST'])
def assess_eczema_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')
```

```
@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')
```

Figure 59: API for acne, eczema, and psoriasis severity assessment

- Backend tested using Postman for request validation, response verification, and exception handling.

2.7.4 Frontend Implementation

- React-based frontend handles file uploads, displays detection overlays and classification results.
- API integration with Flask backend ensures real-time communication and data retrieval.

```
useEffect(() => {
  const fetchData = async () => {
    try {
      setError(null);
      const response = await fetch(`http://localhost:5000/api/patient-results/${patientId}`);
      if (!response.ok) throw new Error('Failed to fetch patient data');

      const data = await response.json();
      setPatientData(data.patient);
      setTestResults(data.testResults);
    } catch (error) {
      console.error(error);
      setError('Failed to load patient data. Please try again later.');
    } finally {
      setLoading(false);
    }
  };
  fetchData();
}, [patientId]);
```

Figure 60: React frontend fetching data from Flask backend using dynamic API integration

- Pages include:
  - Image upload interface for each disease

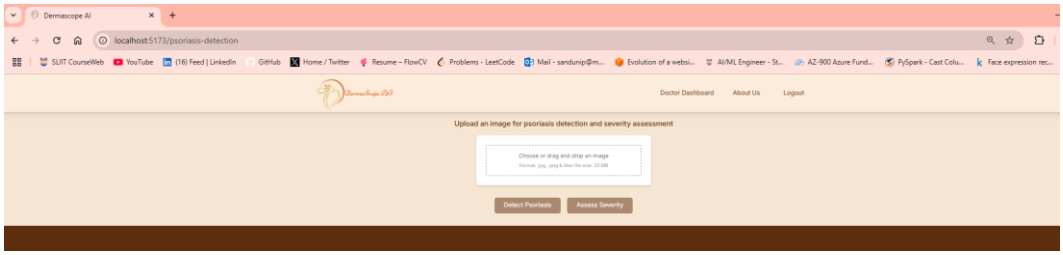


Figure 61: Image Uploading Interface

- Real-time response view with detection outputs

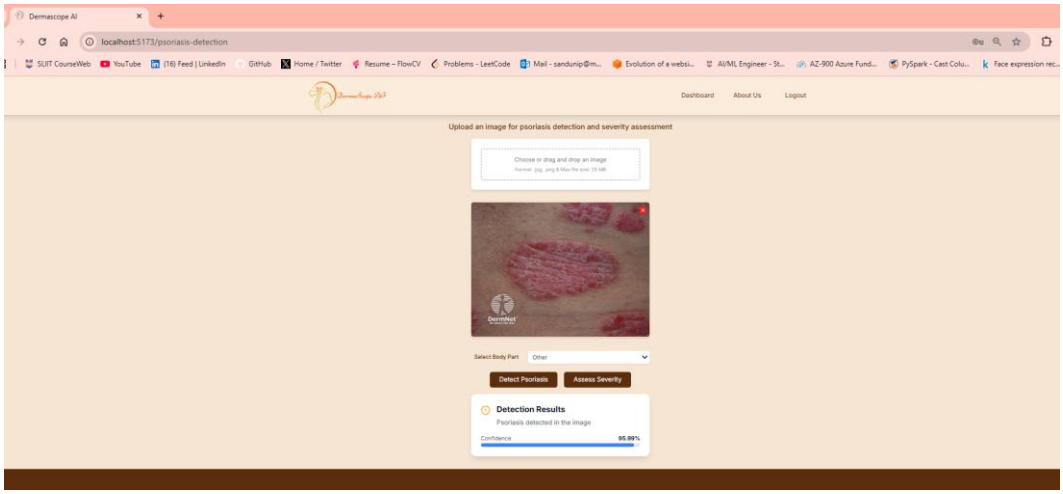


Figure 62: Response view with detection outputs

- Severity interpretation card with confidence levels

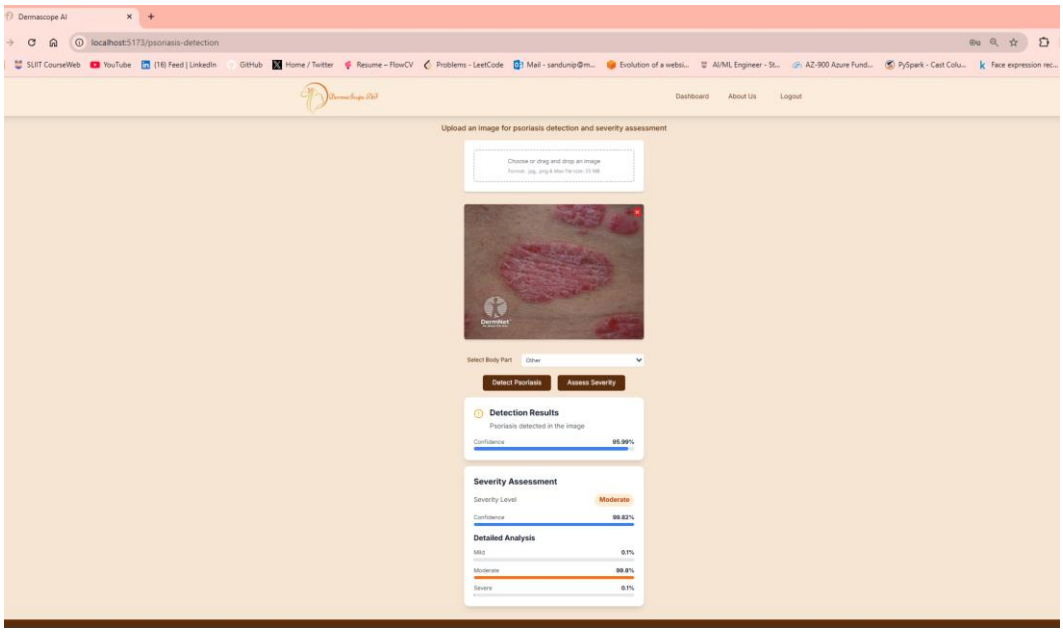


Figure 63: Severity interpretation card with confidence levels

- Doctor dashboard for reviewing patient history and visual feedback

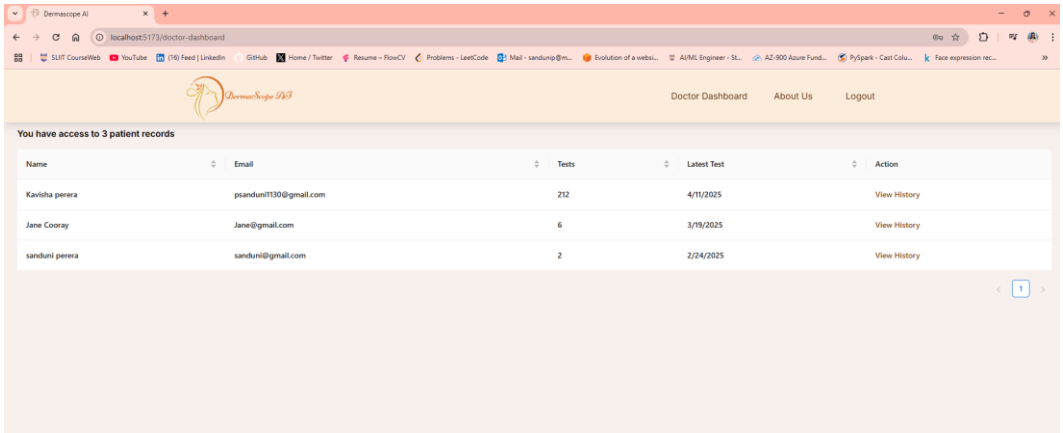


Figure 64: Doctor dashboard

- Patient history and progress tracking page

Figure 65: Patient history and progress tracking page

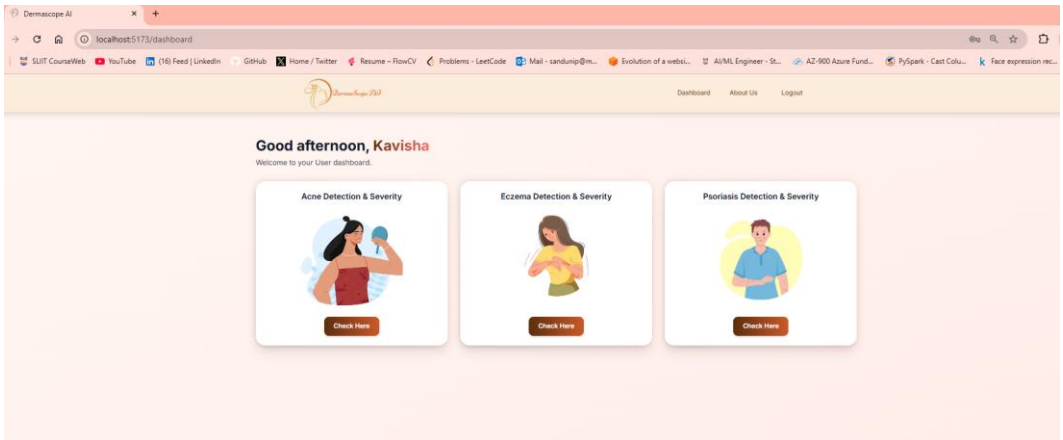


Figure 66: Dashboard Page

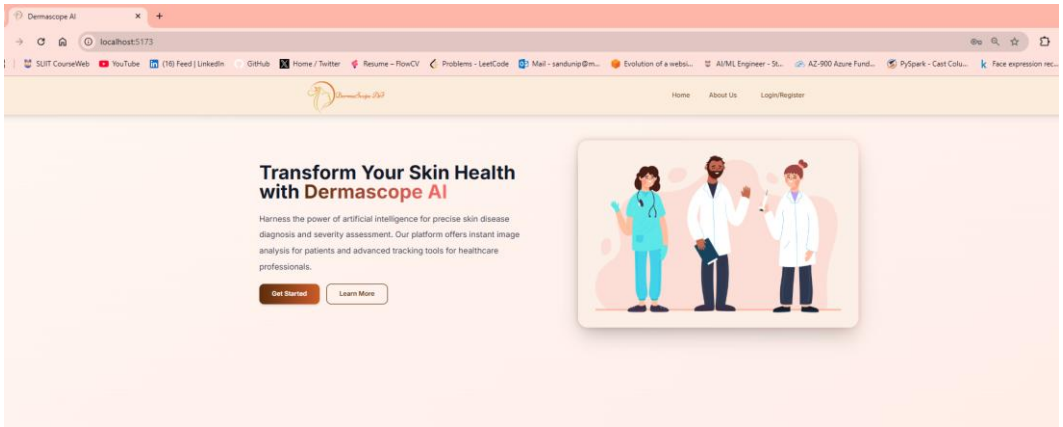


Figure 67: Home Page

2.8 System Diagram

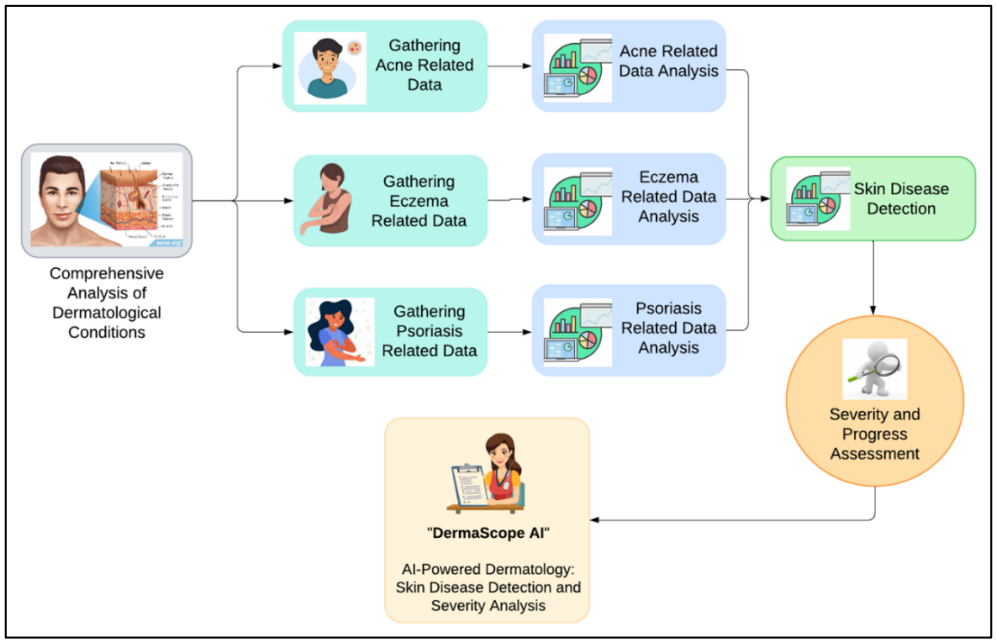


Figure 68: Overall System Architecture

The system architecture begins with the collection of acne, eczema, and psoriasis data from patients. Each dataset undergoes specific analysis tailored to its respective skin condition. Following this, the analyzed data is fed into a unified detection module that identifies the presence of skin diseases. Once detection is complete, a severity and progress assessment are performed. All these components are integrated into the DermaScope AI platform, which offers a complete solution for automated dermatological evaluation and monitoring.

2.8.1 Component Diagram

2.8.1.1 Acne Component Diagram

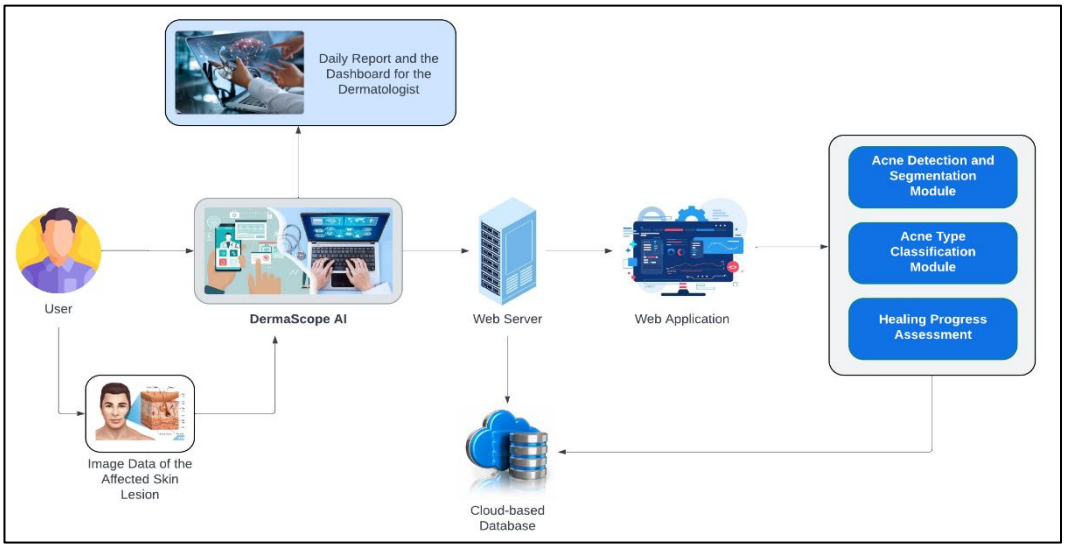


Figure 69: Acne Component Diagram

2.8.1.2 Eczema Component Diagram

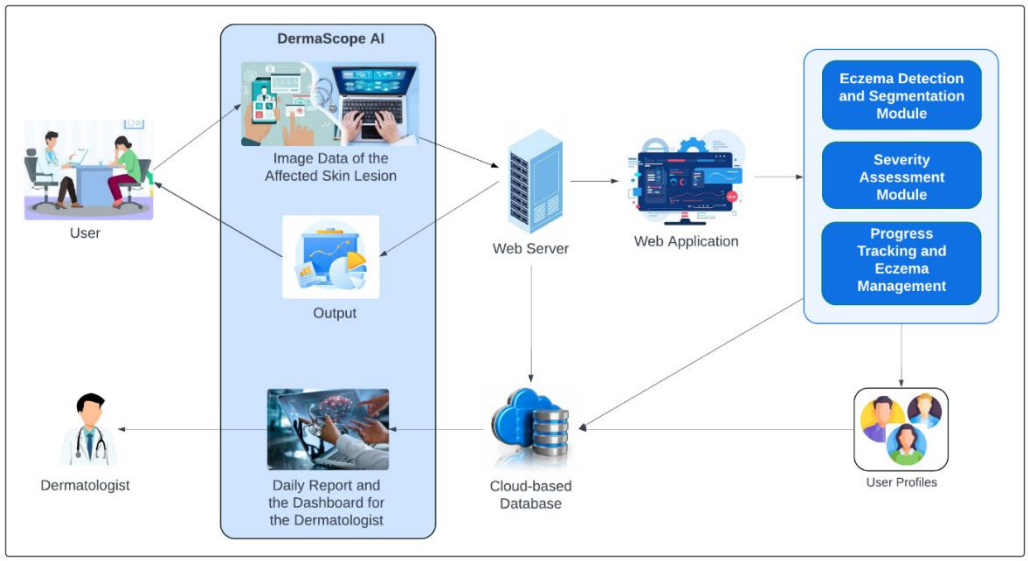


Figure 70: Eczema Component Diagram

2.8.1.3 Psoriasis Component Diagram

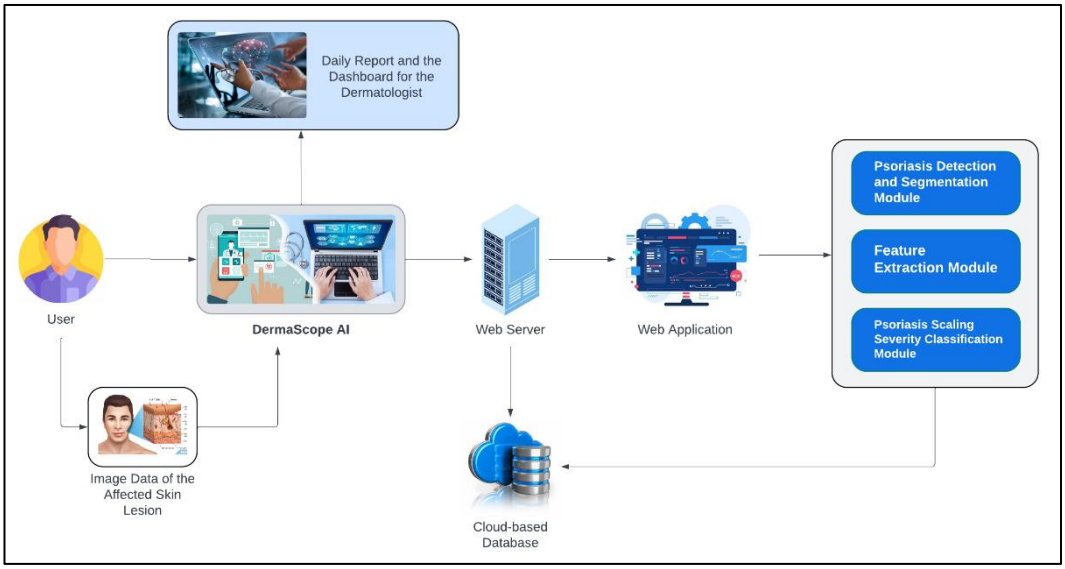


Figure 71: Psoriasis Component Diagram

3. RESULTS AND DISCUSSIONS

3.1 Acne

3.1.1 Acne Detection

Model	Accuracy	Error Rate
CNN	93.25%	6.75%
VGG16	99.28%	0.72%
VGG19	98.07%	1.93%
ResNet	83.02%	16.98%
EfficientNetB0	48.67%	51.33%
Siamese	91.07%	8.93%

3.1.1.1 CNN

The CNN model achieved an accuracy of 93.25% with an error rate of 6.75%, placing it among the top-performing models evaluated. The confusion matrix, as shown in Figure X, demonstrates that the model correctly classified 197 true negatives and 193 true positives, with 15 false positives and 10 false negatives. This relatively low misclassification rate highlights the model's effectiveness in binary classification for acne detection.

The training and validation curves for accuracy and loss, illustrated in Figure Y, further support the model's stability and generalization ability. The training accuracy steadily improves over epochs and surpasses 97%, while the validation accuracy maintains consistency around 92–94%, suggesting that the model is learning meaningful features without overfitting.

The loss curves for both training and validation also converge smoothly toward zero, indicating efficient optimization and minimal performance gap between training and testing datasets. These results reinforce the CNN model’s potential as

a lightweight and accurate architecture for deployment in real-time or resource-constrained environments.

Although it does not outperform VGG16, the CNN model remains a strong candidate, especially for scenarios where computational efficiency is a critical factor.

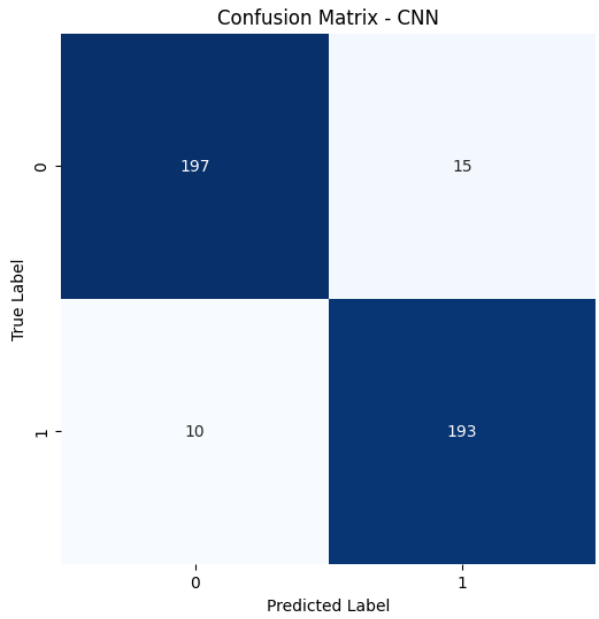


Figure 72: Confusion Matrix of CNN

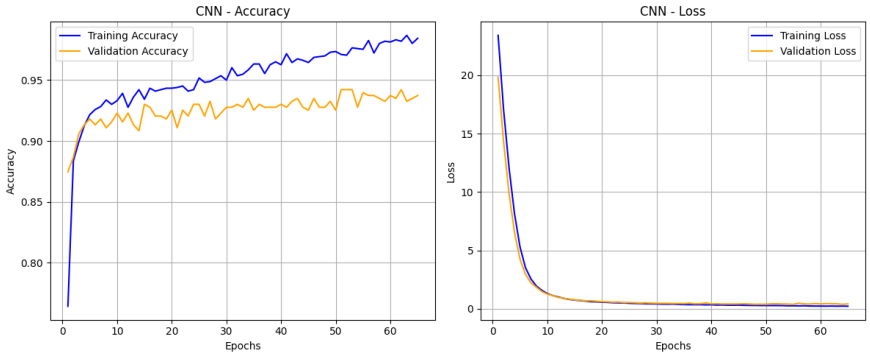


Figure 73: Accuracy and loss graphs of CNN

3.1.1.2 VGG16

Among all the models tested, the VGG16 (Custom) architecture demonstrated the highest performance for acne detection, achieving a remarkable accuracy of 99.28% and a minimal error rate of 0.72%. The confusion matrix in Figure X clearly shows this model's precision, with 212 true negatives and 200 true positives, and only 3 false negatives importantly, there were no false positives, highlighting the model's near-perfect classification ability.

The model's training and validation curves, shown in Figure Y, reflect excellent convergence behavior. The training accuracy consistently reaches ~100%, and the validation accuracy stabilizes around 98-99%, indicating high generalization capacity without signs of overfitting. Similarly, the loss curves rapidly decrease and remain close to zero across the epochs, with negligible variance between training and validation losses.

These results reinforce the effectiveness of transfer learning when using a well-established architecture like VGG16, especially in medical image analysis where fine-grained visual details matter. The robustness of VGG16 makes it an ideal choice for the first phase of the acne analysis pipeline, accurately distinguishing acne-prone skin from non-acne cases with clinical-grade reliability.

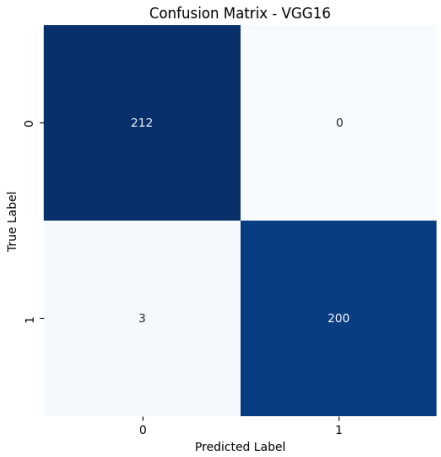


Figure 74: Confusion Matrix of VGG16

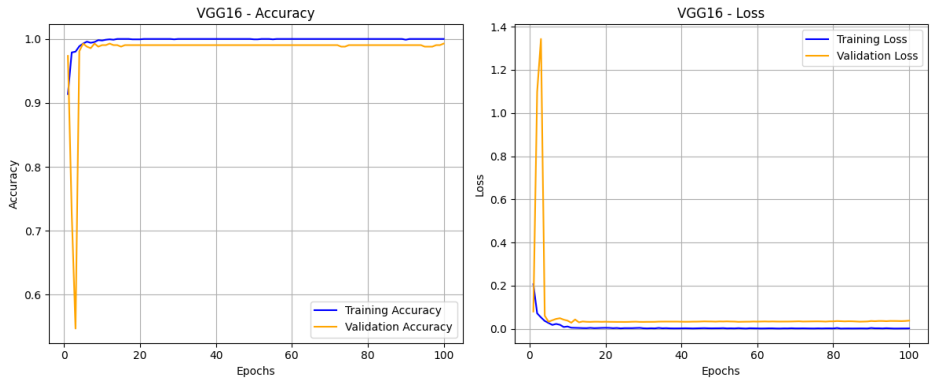


Figure 75: Accuracy and Loss graphs of VGG16

3.1.1.3 VGG19

The VGG19 model also performed remarkably well in the task of acne detection, achieving an accuracy of 98.07% with an error rate of 1.93%. As shown in the confusion matrix (Figure X), the model correctly predicted 206 true negatives and 201 true positives, while misclassifying 6 false positives and 2 false negatives. Although the misclassification is slightly higher compared to VGG16, the model still maintained excellent overall performance.

The training and validation curves (Figure Y) show consistent and rapid convergence. Training accuracy approaches 100%, while validation accuracy stabilizes around 98%, indicating that the model generalizes well to unseen data. The training and validation loss curves remain very close to zero after just a few epochs, suggesting that the model effectively learns without overfitting.

VGG19's deeper architecture, compared to VGG16, allows for capturing more intricate visual features. However, this added complexity does not significantly outperform VGG16 in this case, and may introduce slightly more variability in predictions. Nonetheless, the VGG19 model proves to be a strong and reliable performer, capable of delivering high accuracy in acne detection scenarios.

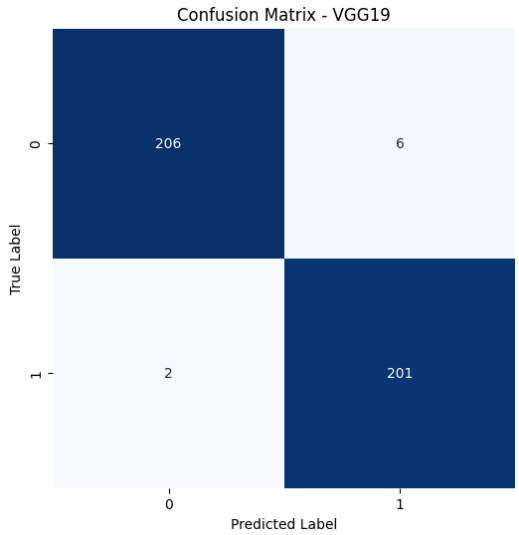


Figure 76: Confusion Matrix of VGG19

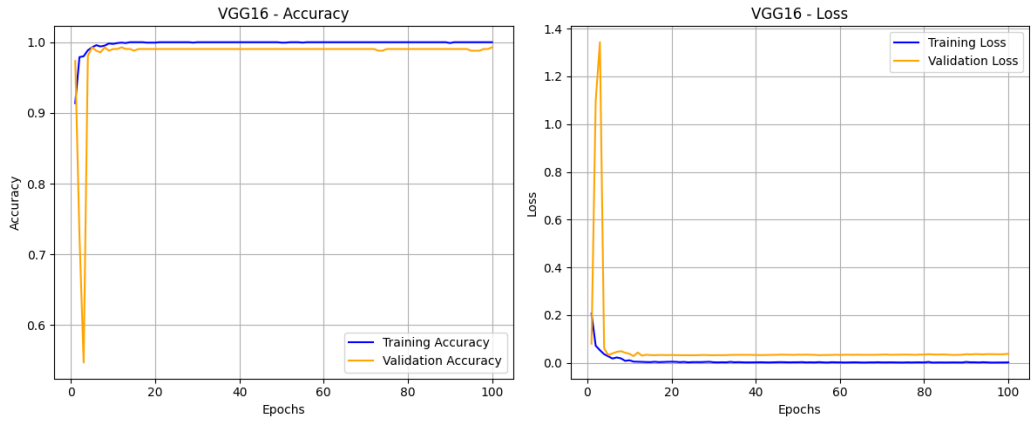


Figure 77: Accuracy and Loss graphs of VGG19

3.1.1.4 ResNet50

Confusion matrix of ResNet50 classification results on the acne dataset. The ResNet50 model achieved 204 true negatives and 191 true positives, with only 8 false positives and 12 false negatives. This outcome corresponds to an overall classification accuracy of 95.18%, reflecting excellent performance. In practical terms, the model correctly identifies acne in most cases (high true positives) while rarely mislabeling normal skin as acne (low false positives) or missing acne cases (low false negatives). The precision of the model's predictions is about 96% and

the recall (sensitivity) about 94%, indicating both high specificity (few false alarms) and high sensitivity in detecting acne. Such strong performance on the validation/test set demonstrates the model's robust ability to generalize to unseen images of skin, a crucial requirement for medical image classification.

Training and validation accuracy over epochs for ResNet50. Throughout training, the ResNet50's training accuracy steadily increased from initial values and eventually plateaued above 99%. This indicates that the model fit the training data extremely well, nearing perfect classification on the training set by the final epochs. The validation accuracy (orange curve) fluctuated in the 93–95% range across epochs, rather than rising monotonically. These minor oscillations in validation accuracy suggest that each training epoch slightly influenced performance on the hold-out data (some epochs saw a bit higher or lower accuracy). Overall, the validation accuracy remained high (~94%) and closely aligned with the training accuracy, which implies good generalization – the model maintains high performance on data it was not trained on, with only a small gap between training and validation accuracy. Training and validation loss over epochs for ResNet50. The training loss (blue curve) exhibits a smooth downward convergence, approaching a very low value as epochs progress. This smooth decline signifies that the model was learning effectively, reducing its error on the training set consistently. In contrast, the validation loss (orange curve) shows some inconsistency with occasional spikes (increases) during training before eventually stabilizing. Early on, as the model improved, validation loss did not decrease uniformly each epoch – at certain points it spiked, indicating that the model's performance on validation data temporarily worsened (likely when the model overfitted some patterns or encountered difficult validation samples). However, these fluctuations diminished in later epochs, and the validation loss trend leveled off, roughly paralleling the training loss at a slightly higher value. By the end of training, the relatively low and stable validation loss suggests the model's

performance on unseen data had stabilized, albeit with a slightly higher error than on the training set (a normal outcome due to generalization to new data).

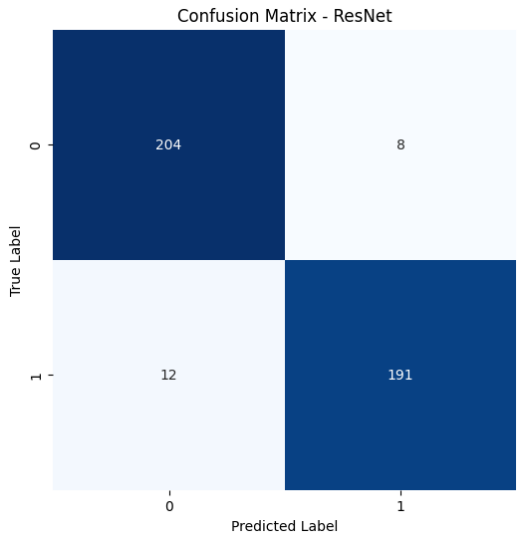


Figure 78: Confusion Matrix ResNet

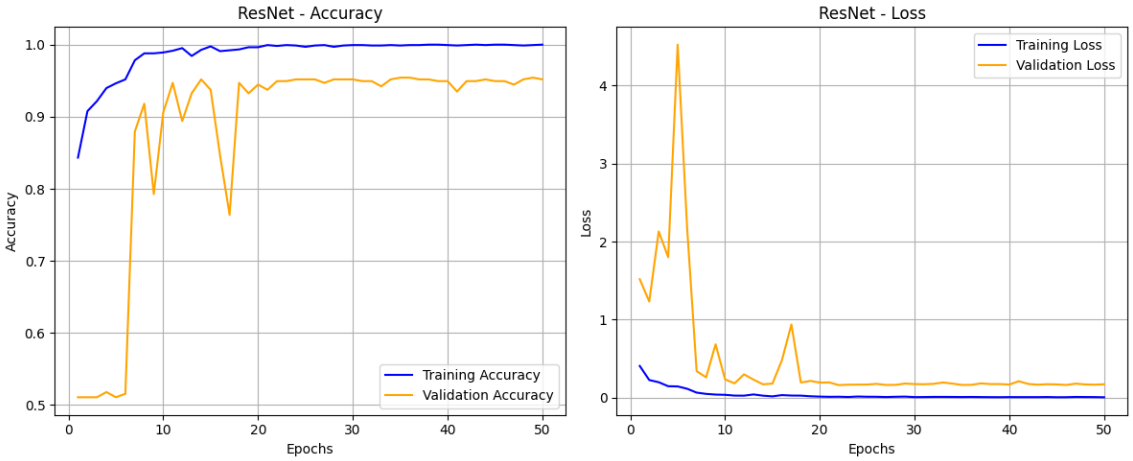


Figure 79: Accuracy and Loss graphs of ResNet

3.1.1.5 EfficientNetB0

The performance of the EfficientNetB0 model in acne detection was significantly lower compared to other tested models. The model achieved an accuracy of only

48.67%, with an error rate of 51.33%, indicating that the model struggled to differentiate between acne and non-acne images.

The confusion matrix (Figure X) illustrates a clear failure in classification. The model misclassified all instances by labeling all inputs as acne-positive, resulting in 0 true negatives, 212 false positives, and 203 true positives, with 0 false negatives. This shows a complete class imbalance in predictions, where the model heavily favors one class and completely ignores the other. Such behavior suggests issues with training convergence or data misinterpretation.

Looking at the accuracy and loss curves (Figure Y), it is evident that the model failed to learn meaningful patterns. Both training and validation accuracy remained around 50% throughout the training, barely above random chance for binary classification. Moreover, the training loss initially spiked and quickly dropped, but without a corresponding improvement in validation performance. The validation loss curve remained flat, further suggesting that the model could not generalize or adapt to the validation data.

In comparison to other models such as VGG16, VGG19, or ResNet50, EfficientNetB0 was unable to learn discriminative features for acne detection in this study. While EfficientNet architectures have shown promise in many vision tasks due to their compound scaling, their performance can degrade significantly without careful hyperparameter tuning and sufficient training data.

Given these findings, EfficientNetB0 is not a suitable candidate for this specific acne detection task under the current training setup. Future attempts to improve its performance could involve more tailored preprocessing, adjustment of training strategy (e.g., freezing fewer layers, modifying learning rate), or using a larger version of EfficientNet (e.g., B3 or B4) to enhance its feature extraction capabilities.

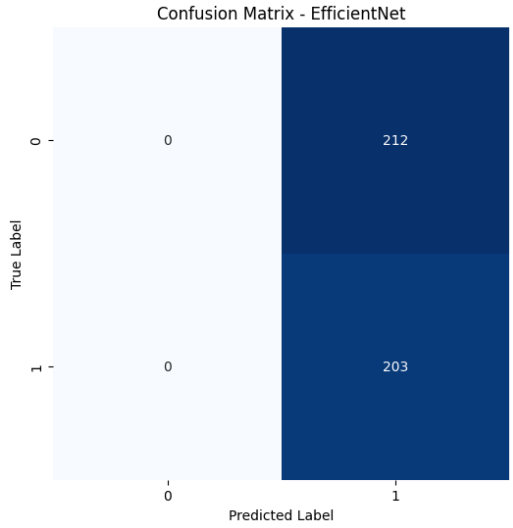


Figure 80: Confusion Matrix of EfficientNet

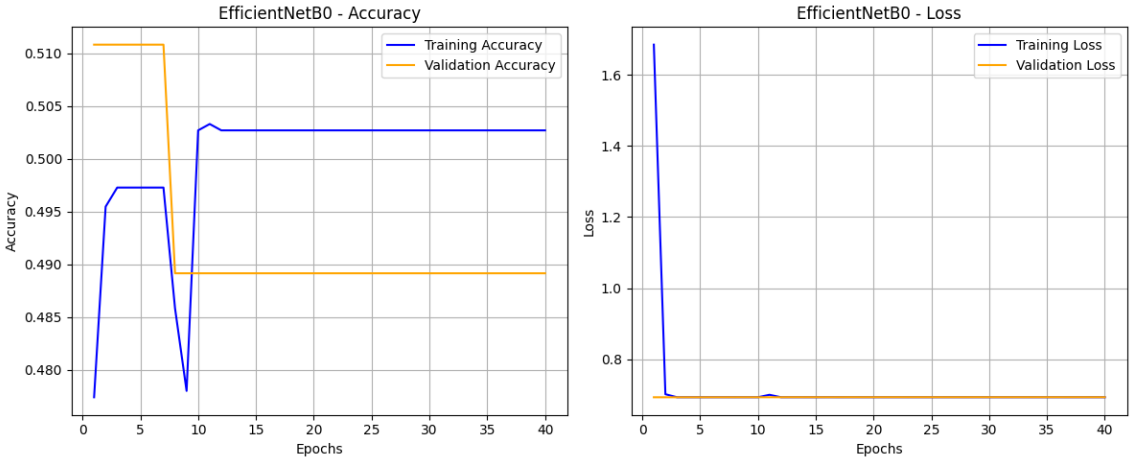


Figure 81: Accuracy and Loss graphs of EfficientNet

3.1.1.6 Siamese Neural Networks

The Siamese model demonstrated commendable performance for acne detection, achieving an accuracy of approximately 91.30%. As depicted in the confusion matrix (Figure X), the model correctly classified 467 true positives (acne cases) and 445 true negatives (non-acne cases). However, there were 33 false negatives

and 55 false positives, showing a slight imbalance in misclassifications, with a slightly higher tendency to mislabel non-acne images as acne.

The training and validation accuracy curves (Figure Y) show that the model converged effectively, with training accuracy nearing 100% and validation accuracy stabilizing around 91–93%. This reflects the model's strong learning capability, while still generalizing well to unseen data. The loss curves further illustrate this trend: the training loss steadily decreased, while the validation loss exhibited some fluctuations and an increasing trend toward later epochs. This could indicate the beginning of overfitting, where the model continues to improve on training data but begins to degrade slightly on validation performance.

The Siamese architecture, which focuses on measuring similarity between image pairs, is particularly well-suited for binary classification tasks with subtle visual differences, such as differentiating between acne and non-acne skin. Its comparative learning approach allows it to generalize better in low-data scenarios or where class definitions are fine-grained.

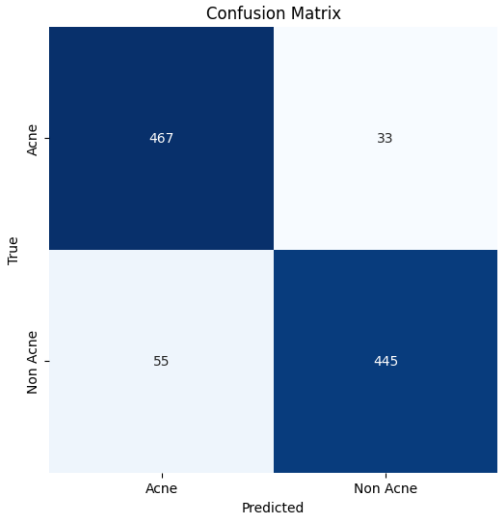


Figure 82: Confusion Matrix of Siamese neural Networks

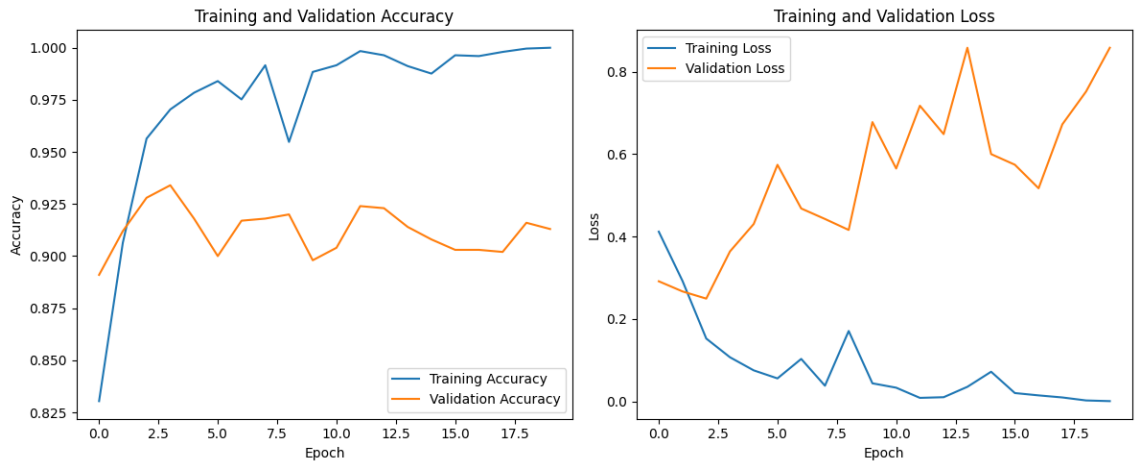


Figure 83: Accuracy and Loss graphs of Siamese Neural Networks

3.1.2 Acne Severity Detection

Model	**Accuracy**	**Loss Rate**
EfficientNetB0	38.35%	61.65%
VGG16	73.34%	26.34%
ViT	76.53%	23.47%
CNN	71.74%	28.26%
Siamese with CNN	40.00%	60.00%

3.1.2.1 EfficientNetB0

In the context of acne severity classification, the EfficientNetB0 model yielded the lowest performance among all the tested architectures. With an overall accuracy of only 38.35% and a loss rate of 61.65%, the model clearly struggled to learn meaningful features for distinguishing between the four severity levels.

As illustrated in the training vs validation accuracy plot (Figure X), the training accuracy steadily improved, reaching over 80% by the 50th epoch. However, the validation accuracy remained highly unstable, fluctuating between 20% to 60%, and never showing consistent improvement. This discrepancy between training and

validation performance indicates significant overfitting, where the model fits the training data well but fails to generalize to unseen samples.

The loss curves (Figure Y) reinforce this conclusion. While the training loss converged to a minimal value, the validation loss exhibited extreme volatility, spiking to values exceeding 7000 in later epochs. Such erratic behavior is a clear indicator that the model was unable to adapt effectively to the validation dataset.

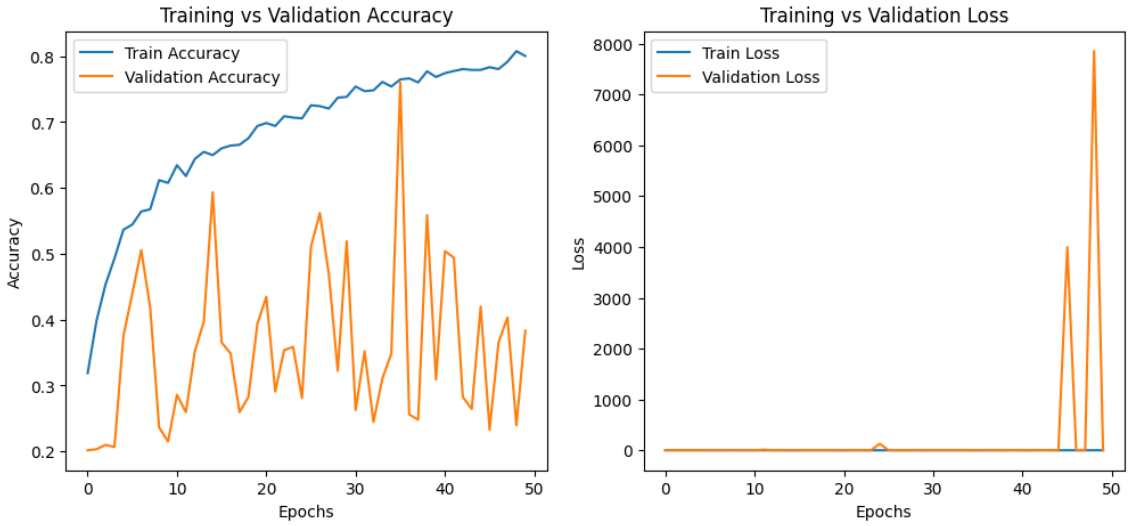


Figure 84: Accuracy and Loss graphs of EfficientNet

3.1.2.2 CNN

The custom CNN model demonstrated strong performance in classifying acne severity across four levels (Level 0 to Level 3). It achieved an overall accuracy of 71.74%, with a corresponding loss rate of 28.26%, positioning it among the top-performing models in this task.

The confusion matrix (Figure X) reflects the model's ability to distinguish between severity classes with reasonable consistency. The true positive counts were relatively balanced:

- Level 0: 105 correct predictions

- Level 1: 112 correct predictions
- Level 2: 112 correct predictions
- Level 3: 105 correct predictions

Although some misclassifications occurred such as confusion between Level 0 and Level 1, and between Level 2 and Level 3 the model showed no severe class imbalance, and most predictions were clustered around the correct class or neighboring severity levels, which is common due to the subjective and gradual nature of acne severity.

The training vs validation accuracy plot (Figure Y) shows that the training accuracy steadily improved over 50 epochs, reaching approximately 90%, while the validation accuracy stabilized around 72%. This performance trend indicates that the model was able to generalize well without overfitting significantly.

On the other hand, the loss curves also demonstrate favorable training behavior. After a high initial training loss, the model converged rapidly, and both training and validation losses remained low throughout, with no major spikes suggesting stable optimization and effective learning.

These findings suggest that the CNN model is capable of identifying the subtle visual distinctions between different acne severity levels. Its relatively low complexity, quick convergence, and competitive accuracy make it a strong candidate for lightweight clinical or mobile deployment, especially when real-time analysis is required.

To further improve performance, techniques such as data augmentation, class-specific weighting, or deeper CNN variants could be explored. However, the current results already showcase the CNN model's solid generalization and robustness across multi-class acne severity detection.

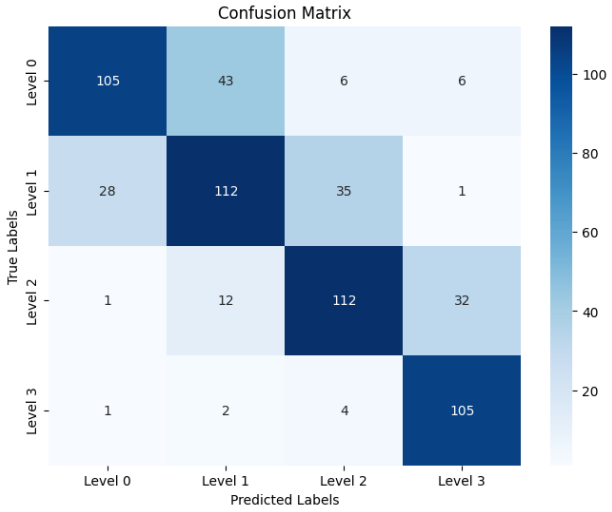


Figure 85: Confusion Matrix of CNN

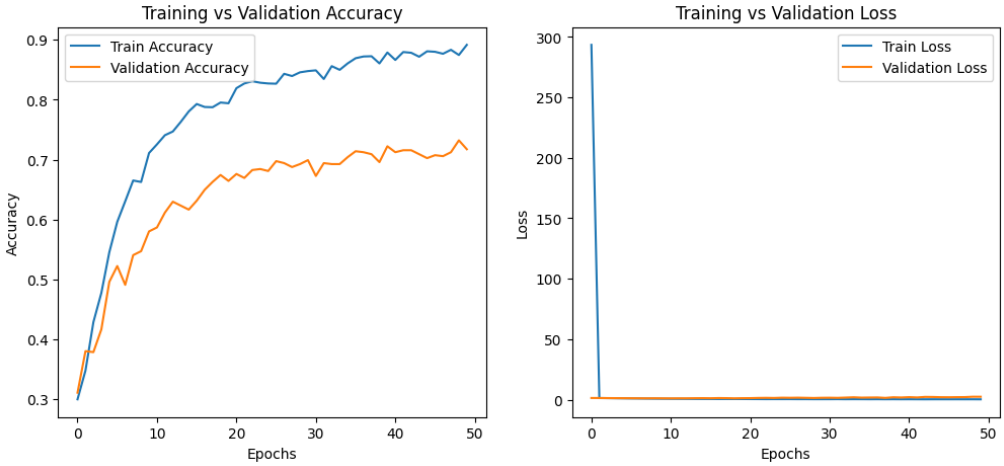


Figure 86: Accuracy and Loss graphs of CNN

3.1.2.3 VGG16

The VGG16 model yielded the highest accuracy in the acne severity classification task, achieving 73.34% accuracy with a 26.34% loss rate. This result highlights VGG16 as the most effective architecture tested for distinguishing between the four severity levels: Level 0, Level 1, Level 2, and Level 3.

The confusion matrix (Figure X) demonstrates the model's balanced performance across all severity classes. Notably:

- Level 1 was the most accurately predicted class, with 134 true positives.
- Level 0 showed greater difficulty, with many of its true instances (96) misclassified as Level 1.
- Level 2 was moderately well-predicted (79 correct), though it was confused with both Levels 1 and 3.
- Level 3 had 78 correct predictions, but also some misclassification into lower severity levels.

Despite a few overlaps between adjacent classes (which is expected given the gradual and visually subtle nature of acne severity progression), the confusion matrix reflects that the model has learned to distinguish between the severity levels fairly well.

The training and validation accuracy curves (Figure Y) show that the model learned effectively, with training accuracy reaching close to 90%, and validation accuracy stabilizing between 70–75%. These values indicate good generalization to unseen data, with a manageable performance gap between training and validation.

The loss curves reinforce this conclusion: the training loss steadily declined, while the validation loss remained relatively flat with some fluctuations. These spikes in validation loss suggest the presence of occasional overfitting, especially in later epochs, where the model may begin to memorize more specific patterns from the training set that don't translate well to the validation set.

Nonetheless, compared to other models, VGG16 proved to be the most balanced and stable performer, effectively handling the multiclass nature of acne severity. Its pre-trained feature extraction capabilities likely contributed to recognizing key dermatological cues associated with severity gradations.

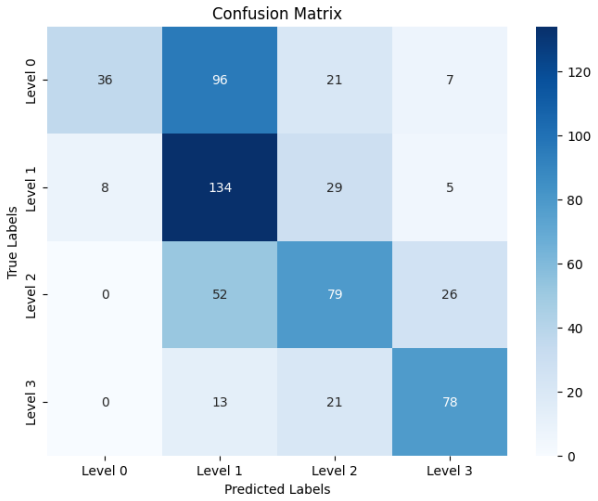


Figure 87: Confusion Matrix of VGG16

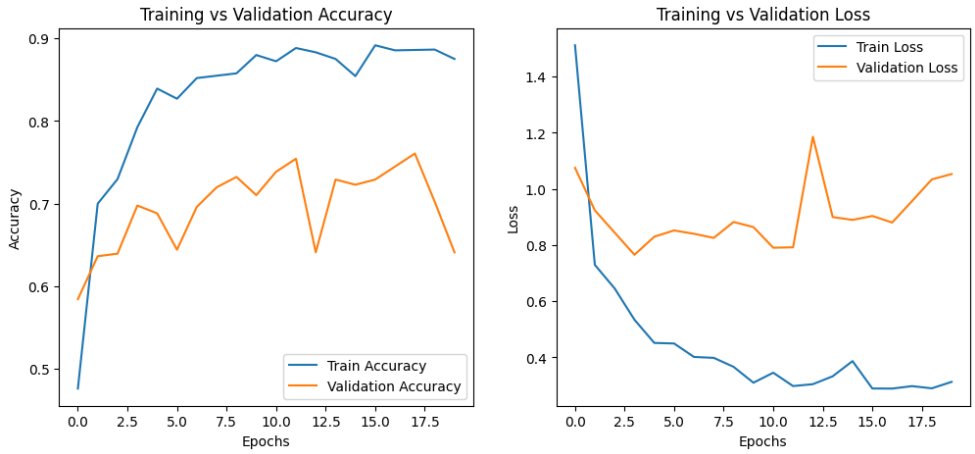


Figure 88: Accuracy and Loss graphs of VGG16

3.1.2.4 ViT

The Vision Transformer (ViT) model demonstrated promising performance in the multi-class acne severity detection task, achieving competitive accuracy and stable predictions across all four severity levels. With consistent improvements in training accuracy and a relatively well-distributed confusion matrix, ViT proves to be a viable alternative to traditional convolutional networks for this problem.

The confusion matrix (Figure X) indicates that the ViT model correctly predicted a large number of samples in each class:

Level 0: 96 true positives

Level 1: 118 true positives

Level 2: 128 true positives

Level 3: 121 true positives

There were some misclassifications, particularly among adjacent severity levels, which is expected due to the subjective visual similarity between mild and moderate cases. For instance, Level 0 was often misclassified as Level 1 (38 cases), and Level 2 was confused with Level 3 (29 cases). However, this reflects a pattern of intelligent generalization, where the model tends to confuse neighboring severity levels rather than distant ones.

The accuracy graph (Figure Y) shows a steady increase in both training and validation accuracy across 100 epochs. Training accuracy eventually reached over 90%, while validation accuracy stabilized between 74–76%, indicating strong generalization and reliable performance on unseen data.

On the other hand, the loss curves revealed an important insight: although training loss steadily decreased, validation loss began to diverge, especially after epoch 50. This suggests some degree of overfitting, likely due to the high capacity of transformer models and their sensitivity to training data distribution. The increasing validation loss, despite stable accuracy, may also point to misaligned confidence levels in prediction probabilities, rather than hard misclassifications.

Despite these challenges, the ViT model proved highly capable of capturing spatial and contextual relationships in skin images. Its attention-based mechanism likely contributed to its strength in identifying fine-grained visual cues essential for severity grading.

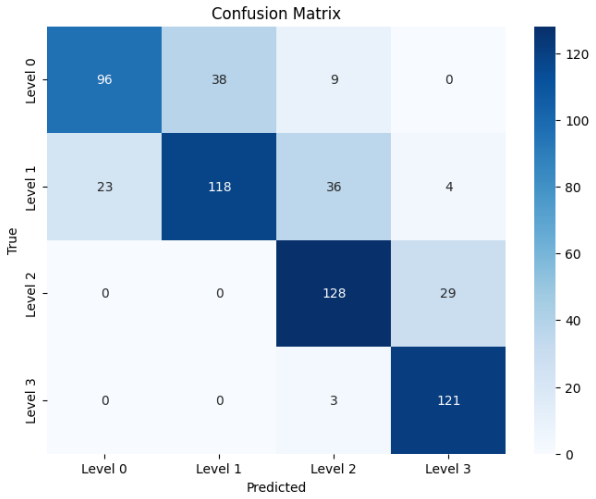


Figure 89: Confusion Matrix of ViT

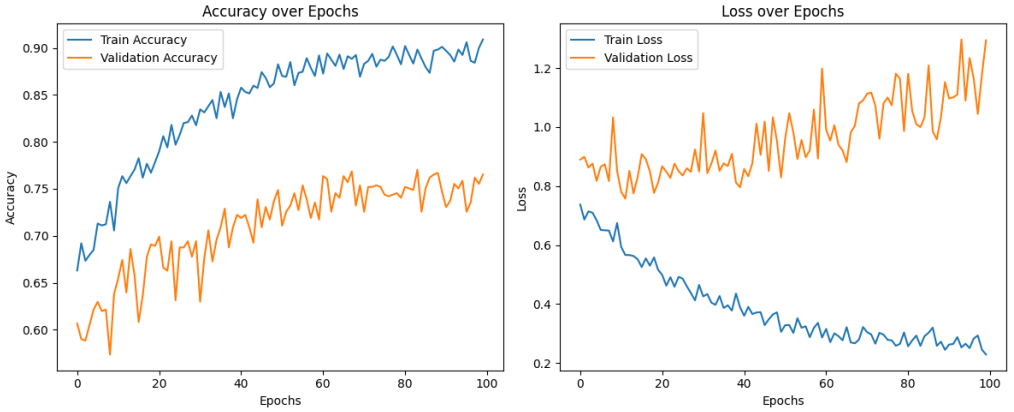


Figure 90: Accuracy and Loss graphs of ViT

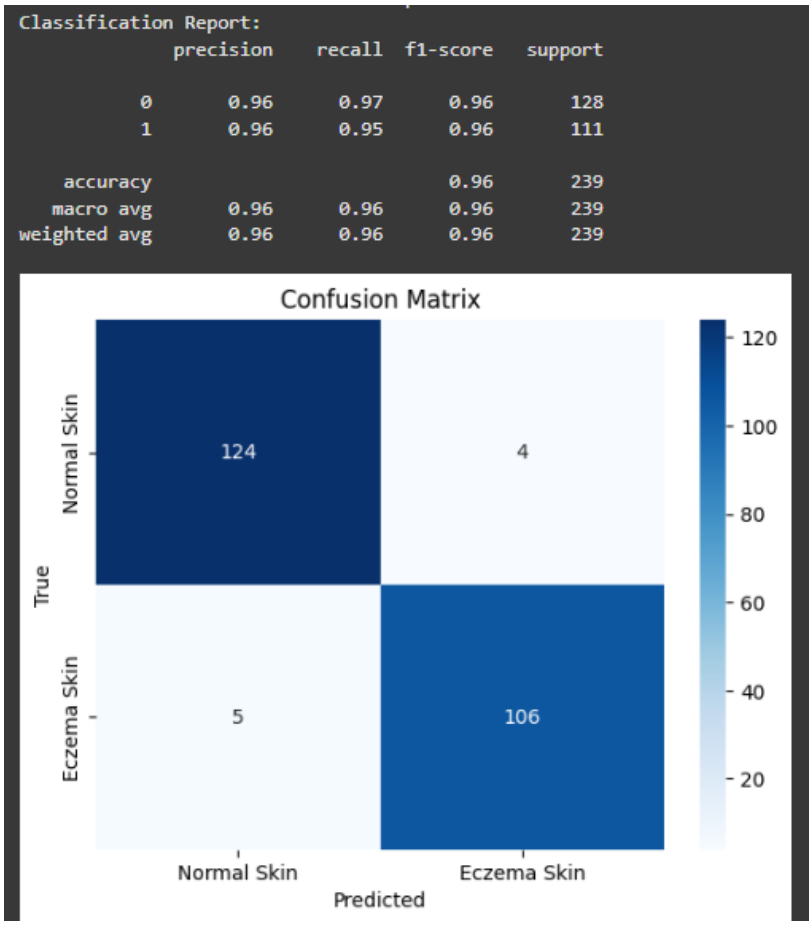
3.2 Eczema

3.2.1 Eczema Detection

3.2.1.1 MobileNetV2 Model (Best Model)

The model's performance was evaluated using classification report and confusion matrix. The overall accuracy achieved by the model was 96%, with a precision of 96%, recall of 96%, and an F1-score of 96% for both classes. The confusion matrix visualizes the predictions, showing that the model correctly classified most samples

with only minimal misclassifications. The balanced performance across both classes indicates that the model is reliable and effective in distinguishing between normal and eczema-affected skin.



```
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.97      0.96       128
           1       0.96      0.95      0.96       111

    accuracy                           0.96       239
   macro avg       0.96      0.96      0.96       239
weighted avg       0.96      0.96      0.96       239
```


Figure 91: Classification report and the confusion matrix for MobileNetV2

The training and validation accuracy and loss curves provide essential insights into the model's learning progress over the 50 training epochs. The accuracy plot shows a steady increase in both training and validation accuracy, with the validation accuracy consistently outperforming the training accuracy. This trend suggests that the model is effectively learning from the data and generalizing well to the validation set. The validation accuracy reaching above 90% indicates strong predictive capability.

The loss plot displays a continuous decline in both training and validation loss, demonstrating that the model is converging effectively without signs of overfitting. The simultaneous reduction in training and validation loss over time indicates that the model is learning useful features without excessively memorizing the training data. The stability and consistency of these curves confirm that the model's architecture and training process are well-designed and optimized for eczema detection.

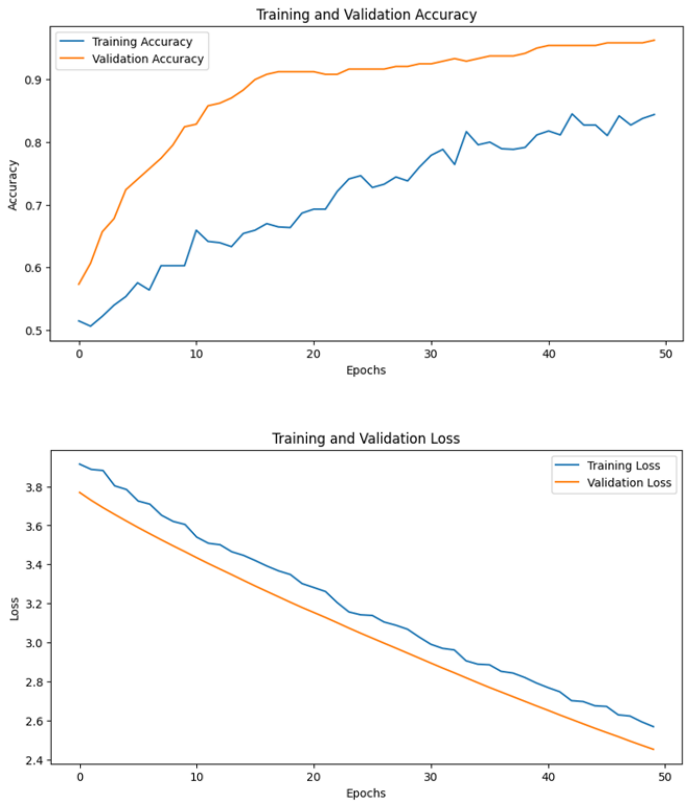


Figure 92: Training and validation accuracy/loss visualization for MobileNetV2

3.2.1.2 CNN Model

The model's performance on the validation set was evaluated using a confusion matrix and classification report. The confusion matrix indicates that the model achieved a high number of correct predictions for both classes, particularly for eczema cases where 108 out of 111 samples were correctly classified. However, a

noticeable weakness is observed in the normal class, where 33 samples were incorrectly predicted as eczema.

The classification report reveals a precision of 97% for normal skin and 77% for eczema, with corresponding recall values of 74% and 97%, respectively. While the model is effective at identifying eczema (high recall), it struggles with precision for eczema cases, leading to false positives. The overall accuracy is 85%, which, although reasonable, suggests room for improvement. Improving the model's precision for eczema detection would enhance its diagnostic reliability and clinical applicability.

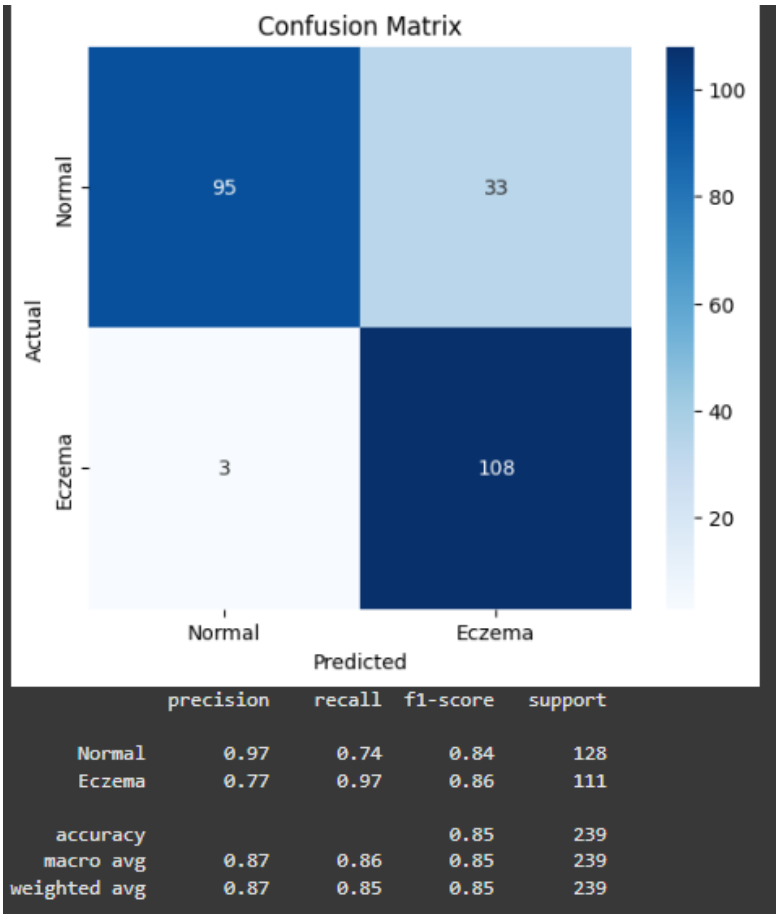


Figure 93: Confusion matrix and the classification report for CNN

The training and validation accuracy and loss curves for this model reveal a significant issue with overfitting. The training accuracy quickly reaches nearly 100% within a few epochs, indicating that the model is learning the training data very effectively. However, the validation accuracy drops sharply after the first epoch and remains low throughout the training process, suggesting that the model is failing to generalize to unseen data.

The loss curves further highlight this problem. While the training loss decreases and stabilizes, the validation loss increases rapidly after the first epoch and continues to rise, indicating severe overfitting. The increasing validation loss is a strong indicator that the model is memorizing the training data rather than learning generalizable patterns.

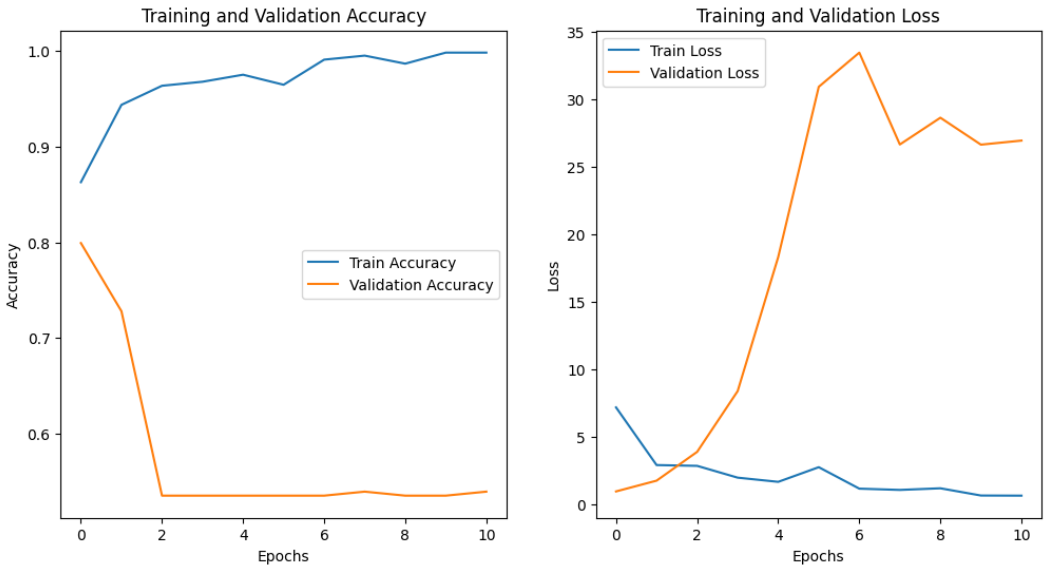
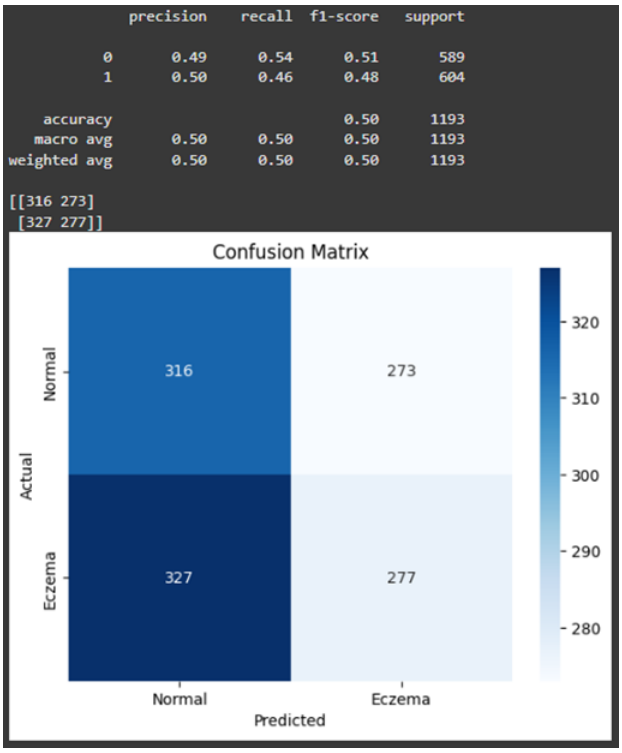


Figure 94: Training and validation accuracy/loss visualization for CNN

3.2.1.3 ResNet50 Model

The model's performance, reflected by an overall accuracy of 50%, indicates poor discriminatory power between Normal and Eczema classes, with nearly identical precision, recall, and F1-scores of around 0.50. The confusion matrix shows a significant misclassification.



```
              precision    recall  f1-score   support

           0       0.49      0.54      0.51       589
           1       0.50      0.46      0.48       604

    accuracy                           0.50      1193
   macro avg       0.50      0.50      0.50      1193
weighted avg       0.50      0.50      0.50      1193

[[316 273]
 [327 277]]
```


Figure 95: Classification report and confusion matrix for ResNet50

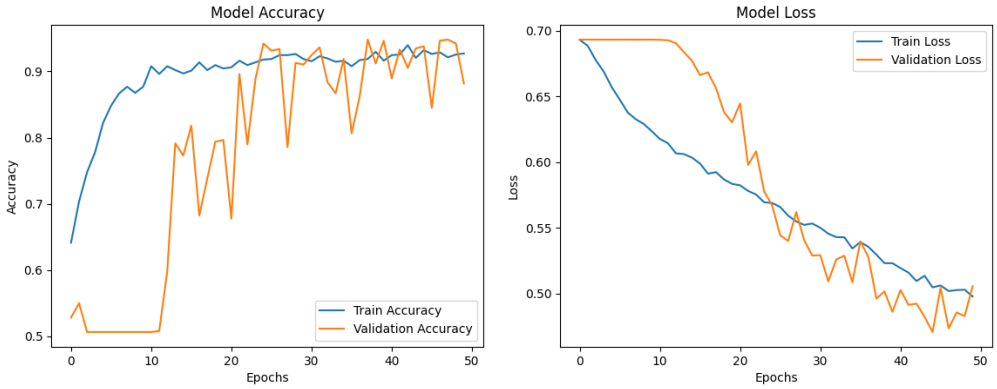
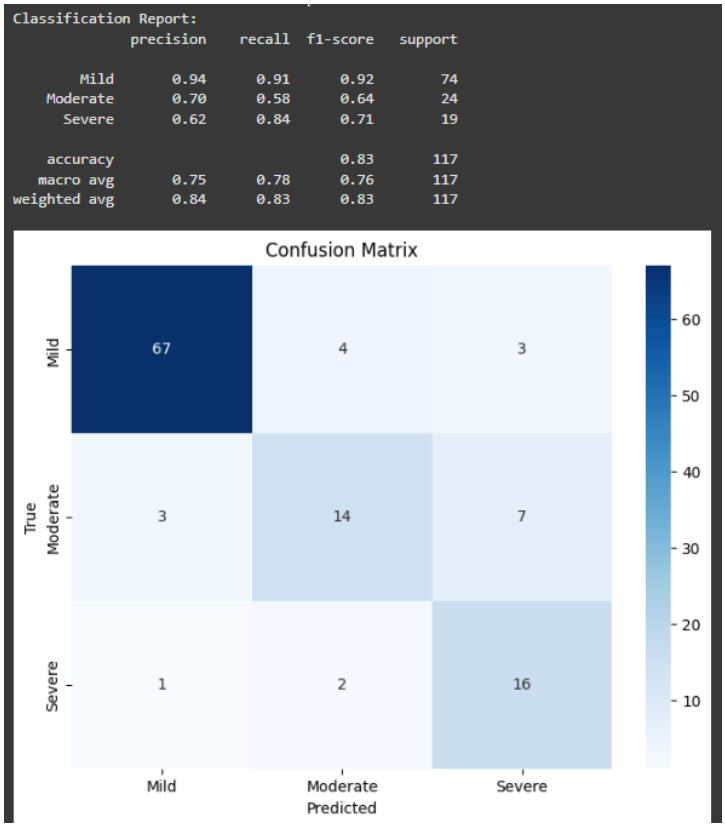


Figure 96: Training and validation accuracy/loss visualization for ResNet50

3.2.2 Eczema Severity Classification

3.2.2.1 EfficientNetB0 Model (Best Model)

The model achieved a commendable accuracy of 83%, with precision, recall, and F1-scores indicating balanced performance. The Mild class shows a particularly high precision of 0.94, a recall of 0.91, and F1-score of 0.92, indicating strong reliability in detecting mild cases. The Severe class also displays good performance, with a recall of 0.84 and F1-score of 0.71, highlighting its ability to accurately identify most severe cases. The Moderate class achieves reasonable precision (0.70) and recall (0.58), indicating that the model is effectively capable of detecting intermediate severity cases. Overall, the model demonstrates reliable and consistent performance across all severity levels, making it a robust and effective solution for eczema severity classification.



```
Classification Report:
              precision    recall  f1-score   support

        Mild       0.94      0.91      0.92        74
    Moderate       0.70      0.58      0.64        24
      Severe       0.62      0.84      0.71        19

    accuracy                           0.83       117
   macro avg       0.75      0.78      0.76       117
weighted avg       0.84      0.83      0.83       117
```


Figure 97: Classification report and confusion matrix for EfficientNetB0

The training and validation accuracy and loss curves illustrate a stable and consistent learning process throughout the training epochs. Both the training and validation accuracy steadily increase over time, eventually converging around 0.75 to 0.80, indicating effective learning and generalization. The training and validation loss curves demonstrate a continuous decline, stabilizing near the end, which confirms that the model is effectively minimizing the loss function without signs of overfitting. The alignment of training and validation curves suggests a well-balanced model that generalizes well to unseen data.

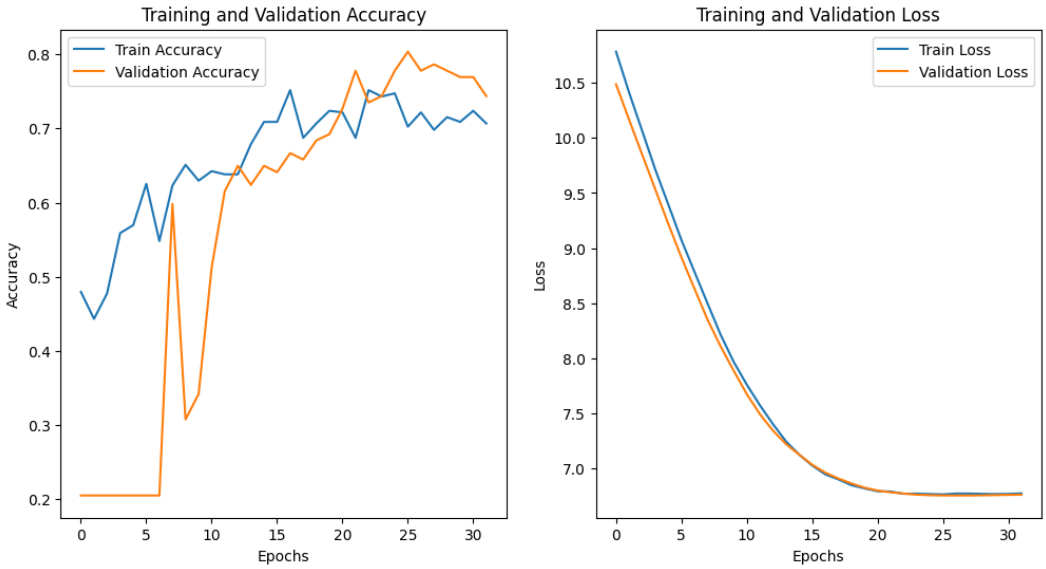
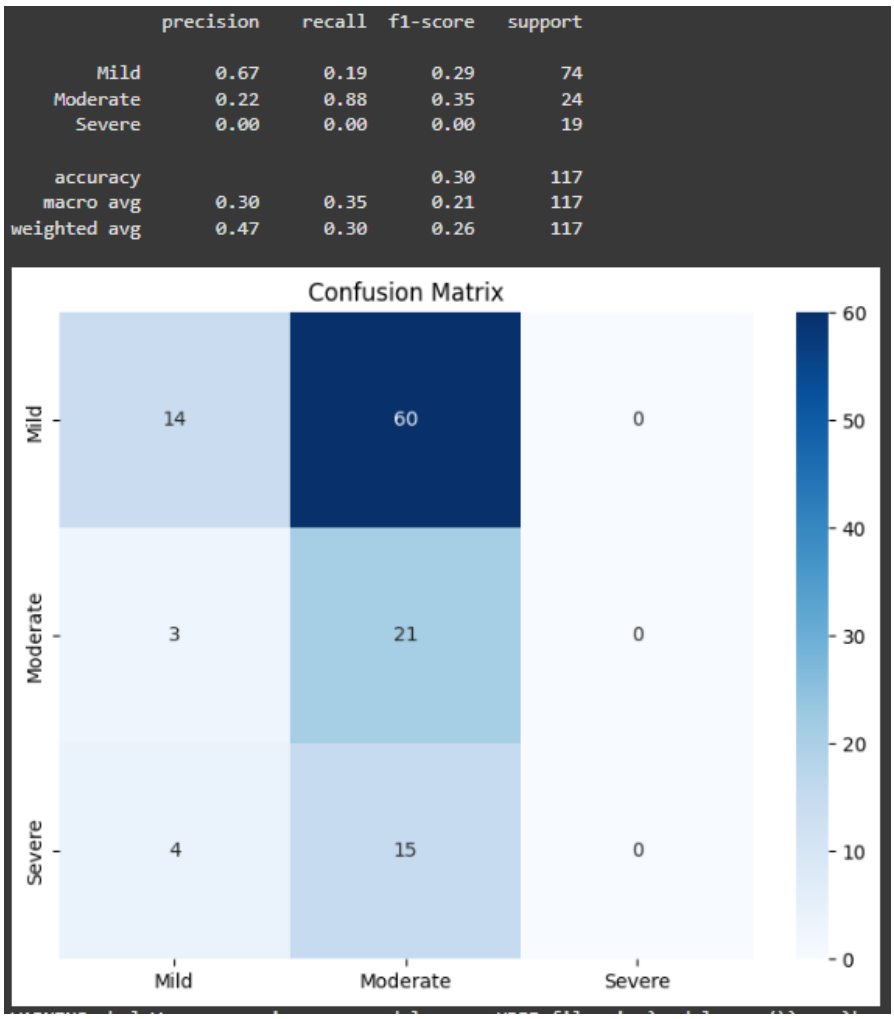


Figure 98: Training and validation accuracy/loss visualization for EfficientNetB0

3.2.2.2 CNN Model

This model did not perform well, achieving a low overall accuracy of 30% with particularly poor performance in detecting Severe cases (0% recall and precision). The confusion matrix shows significant misclassifications, especially with Mild cases being frequently predicted as Moderate. Due to its inadequate performance across all severity levels, this model was ultimately disregarded from further consideration.

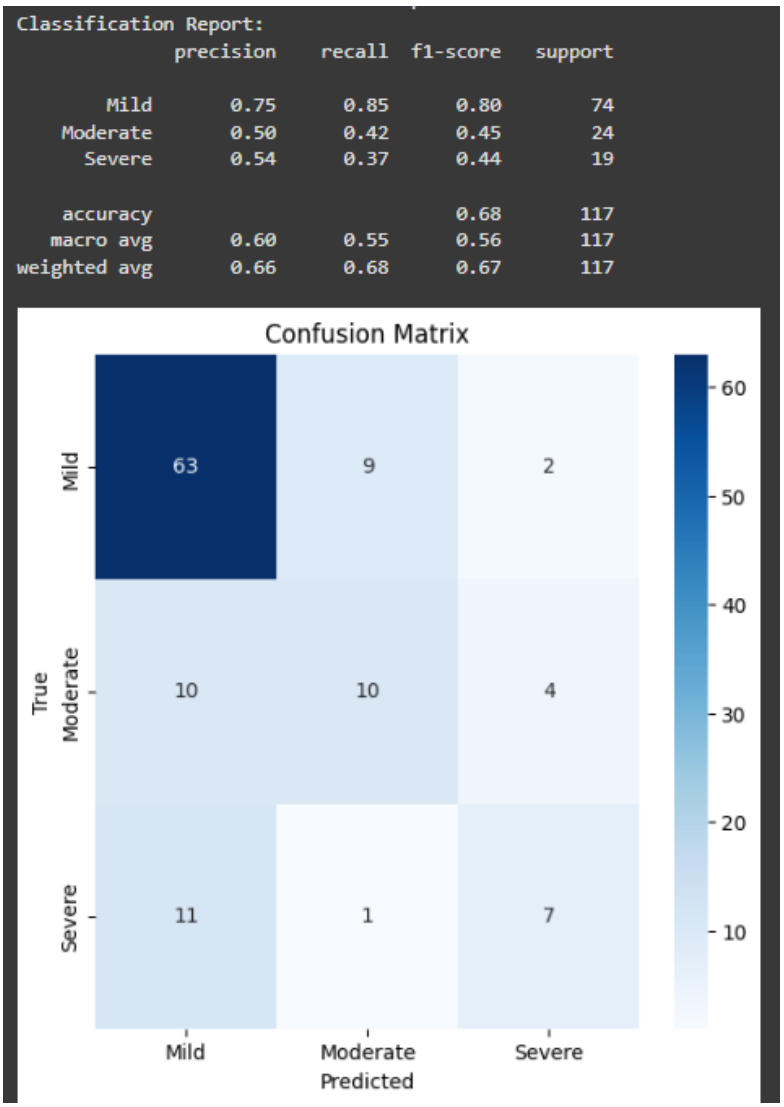


	precision	recall	f1-score	support
Mild	0.67	0.19	0.29	74
Moderate	0.22	0.88	0.35	24
Severe	0.00	0.00	0.00	19
accuracy			0.30	117
macro avg	0.30	0.35	0.21	117
weighted avg	0.47	0.30	0.26	117

Figure 99: Classification report and confusion matrix for CNN

3.2.2.3 ResNet50 Model

This model achieved a moderate accuracy of 68%, with the Mild class performing well at 75% precision, 85% recall, and 80% F1-score. However, the Moderate and Severe classes show reduced performance, with F1-scores of 0.45 and 0.44, respectively. The confusion matrix reveals that most misclassifications occur between Moderate and Severe cases, indicating that the model struggles to distinguish between these categories. Although the performance is reasonable, it is not sufficient for reliable eczema severity classification.



Classification Report:

	precision	recall	f1-score	support
Mild	0.75	0.85	0.80	74
Moderate	0.50	0.42	0.45	24
Severe	0.54	0.37	0.44	19
accuracy			0.68	117
macro avg	0.60	0.55	0.56	117
weighted avg	0.66	0.68	0.67	117

Figure 100:Classification report and confusion matrix for ResNet50

The training and validation curves for this model reveal significant fluctuations in validation accuracy and loss, indicating instability and overfitting. While the training loss steadily decreases over epochs, the validation loss shows erratic spikes, particularly around the 10th epoch, where it sharply increases above 50. Similarly, validation accuracy fluctuates considerably, with several sudden drops and peaks, suggesting that the model struggles to generalize well to unseen data. These inconsistencies indicate that the model's learning process is not well-optimized, making it unreliable for accurate classification.

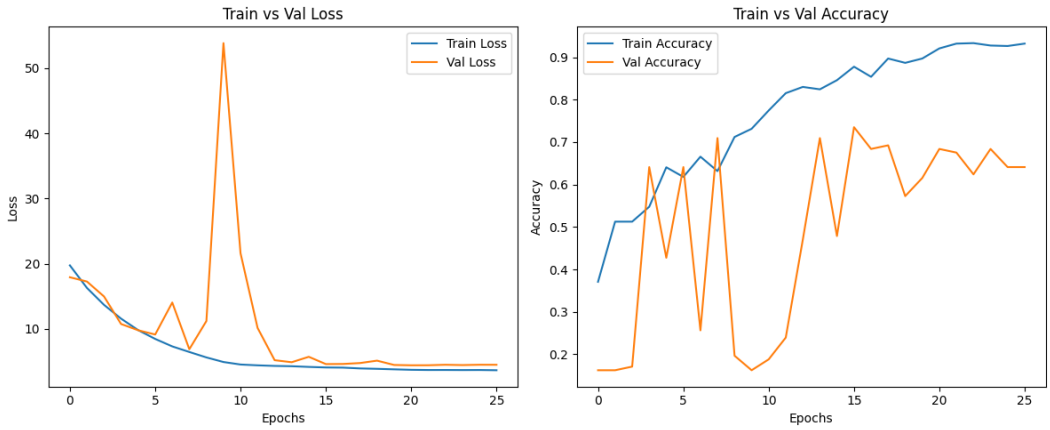


Figure 101: Training and validation accuracy/loss visualization for ResNet50

3.3 Psoriasis

3.3.1 Psoriasis Detection

3.3.1.1 YOLOv8 (Best Model)

The YOLOv8-based psoriasis detection model demonstrated outstanding performance across both classification categories. As depicted in the confusion matrix, the model accurately predicted 46 out of 47 normal skin cases and 60 out of 61 psoriasis-affected cases, achieving an overall precision of 0.977, recall of 0.981, and a remarkable mAP@0.5 of 0.980. These metrics highlight the model's reliability in localizing and classifying lesions with minimal false positives and false negatives, even in complex dermatological images.

```
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 102: YOLOv8 Model Performance Metrics

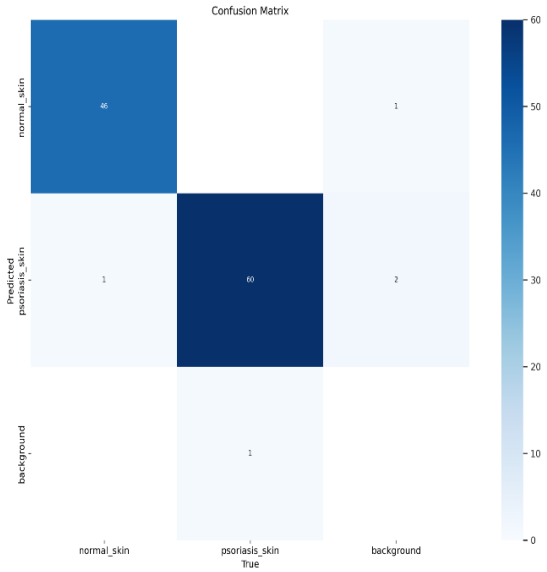


Figure 103: YOLOv8 Model Confusion Matrix

Further evaluation using confidence-based performance curves confirmed the robustness of the YOLOv8 model. The F1-Confidence Curve peaked at 0.96 at a threshold of 0.607, indicating an optimal balance between precision and recall. Precision and Recall Confidence Curves remained above 0.95 across various thresholds, while the Precision-Recall Curve achieved a strong mAP@0.5 of 0.966, highlighting the model's generalization capability.

These results demonstrate YOLOv8's suitability for clinical and real-time applications, offering high accuracy and fast inference. Its consistent performance across diverse conditions—such as varying lighting, skin tones, and lesion types—along with its lightweight design, makes it ideal for deployment in mobile or low-power diagnostic tools. Overall, the YOLOv8 model is a reliable and scalable solution for automated psoriasis lesion detection.

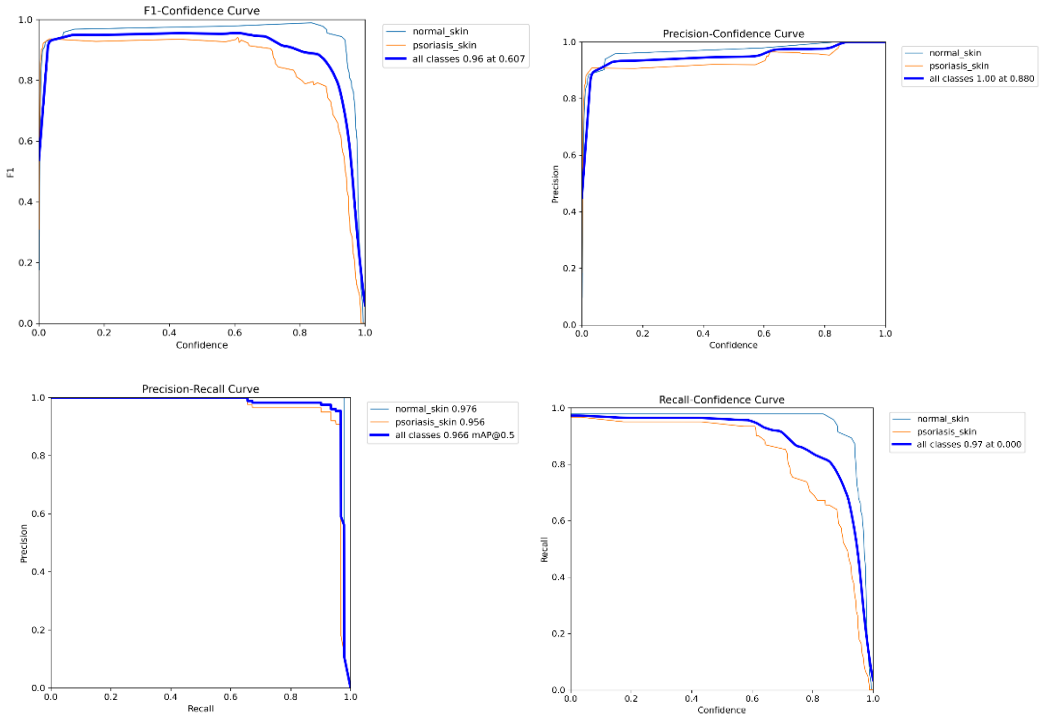


Figure 104: YOLOv8 Model Metric Curves

3.3.1.2 CNN Model

A basic Convolutional Neural Network (CNN) was employed 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 the psoriasis class. While the model successfully identified all psoriasis cases (zero false negatives), it struggled with normal skin classification, misclassifying 24 out of 59 normal 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 105: CNN Classification Report

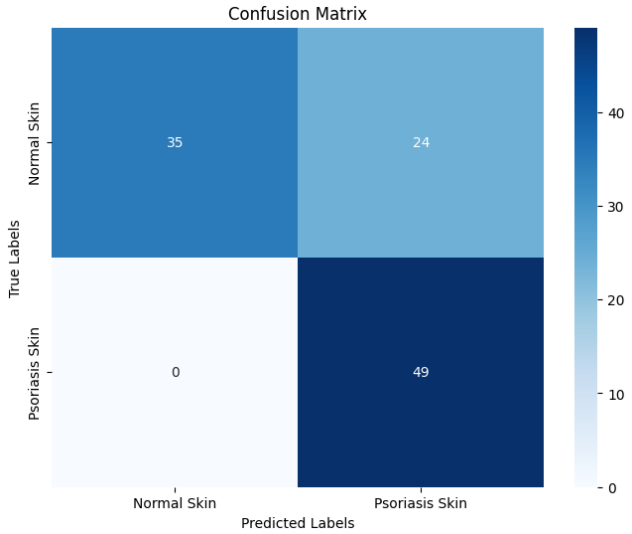


Figure 106: CNN Confusion Matrix

The training and validation accuracy curves revealed noticeable fluctuations, suggesting training instability and potential overfitting. Although the baseline CNN showed promise in detecting psoriasis, its limited generalization ability and

imbalance in class-wise performance underscored the need for a more refined and robust architecture.

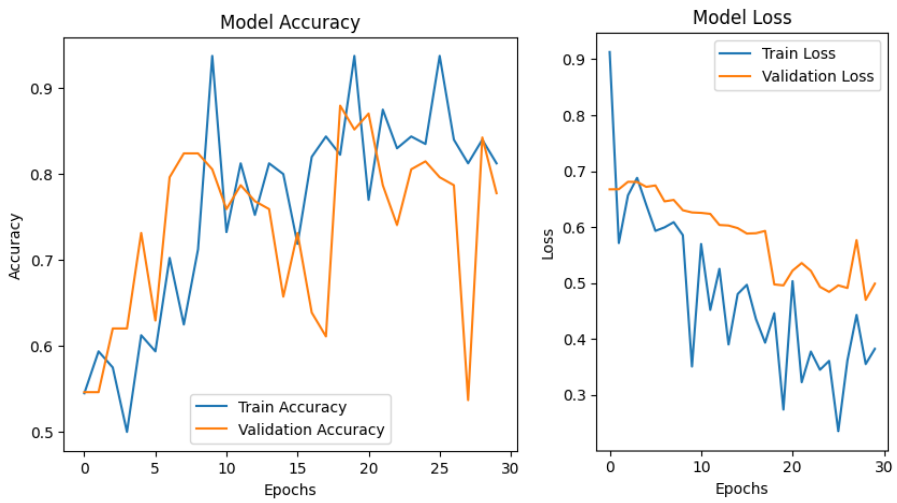


Figure 107: CNN Training and Validation Accuracy/Loss Curves

3.3.1.3 EfficientNetB0 Model

The EfficientNetB0 model achieved perfect classification performance on the validation dataset. The classification report shows precision, recall, and F1-score of 1.00 for both classes—Normal Skin and Psoriasis Skin—indicating zero false positives and false negatives. This is further confirmed by the confusion matrix, where all 54 instances of each class were correctly classified, totaling 108 accurate predictions with no misclassifications.

```
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 108: EfficientNetB0 Classification Report

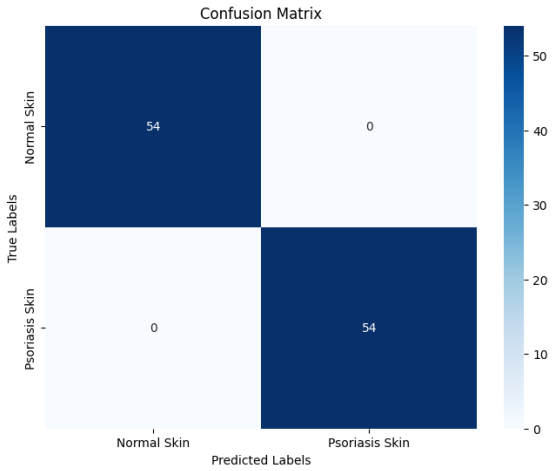


Figure 109: EfficientNetB0 Confusion Matrix

The training and validation accuracy and loss curves illustrate a smooth and consistent learning process. Training accuracy reached 1.0 within a few epochs and remained stable, while validation accuracy mirrored this trend, demonstrating excellent generalization. Additionally, both training and validation loss steadily decreased without signs of divergence, indicating effective learning without overfitting. These results affirm the robustness, reliability, and clinical applicability of the EfficientNetB0-based psoriasis classification model.

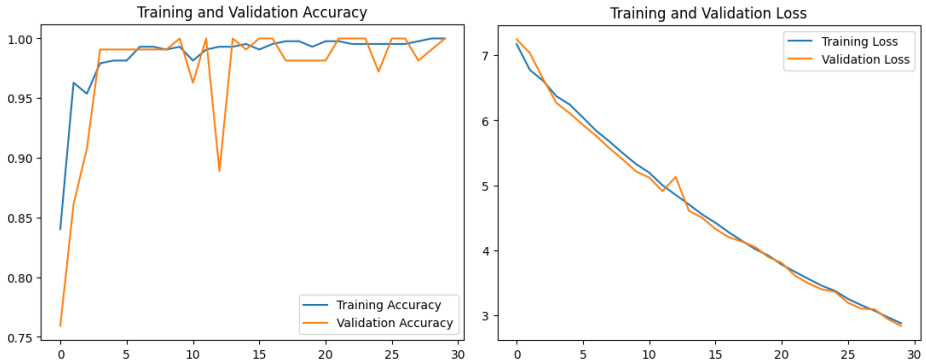


Figure 110: EfficienNetB0 Training and Validation Accuracy/Loss Curves

3.3.1.4 ResNet50 Model

The ResNet50 model achieved a 93% validation accuracy in classifying psoriasis and normal skin. It demonstrated high precision and recall, particularly for normal skin. The confusion matrix reveals 1 false positive and 7 false negatives, indicating strong overall performance with minor 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 111: ResNet50 Classification Report

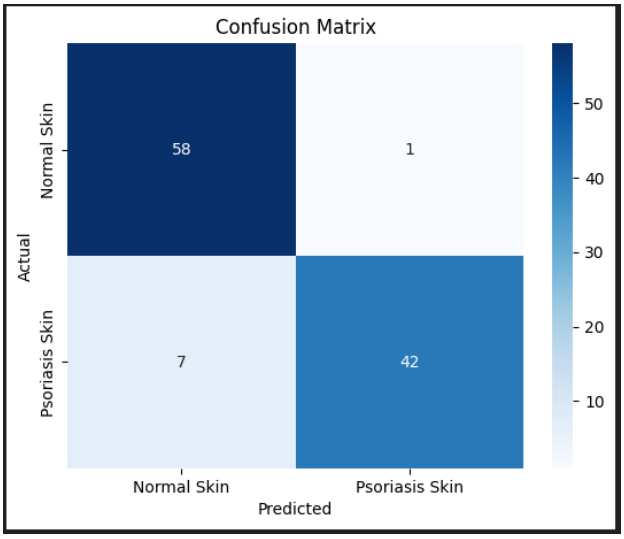


Figure 112: ResNet50 Confusion Matrix

Training accuracy remained consistently high across epochs; however, validation accuracy showed fluctuations, and the validation loss exhibited noticeable spikes, suggesting signs of overfitting. Despite this, the model retained solid classification capabilities and performed reliably on the dataset.

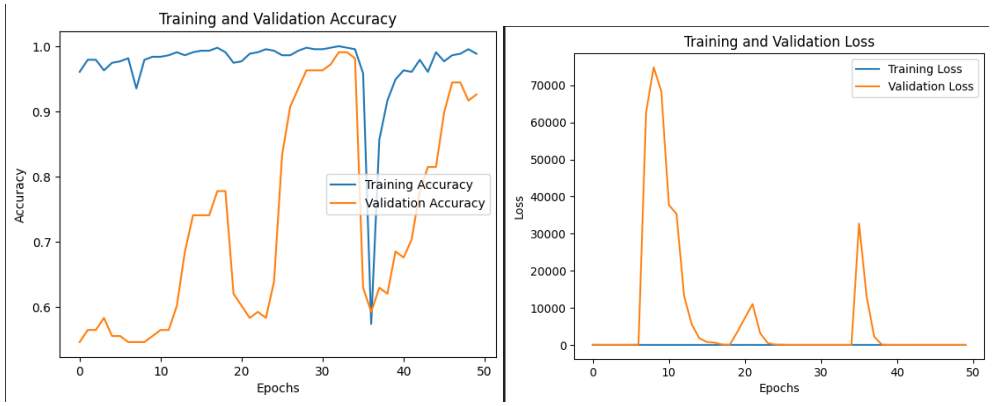


Figure 113: ResNet50 Training and Validation Accuracy/Loss Curves

3.3.1.5 InceptionResNetV2 Model

The InceptionResNetV2 model delivered perfect classification performance, achieving 100% validation accuracy with precision, recall, and F1-score of 1.00 for both normal and psoriasis classes. The confusion matrix confirms this flawless result, with all 59 normal and 49 psoriasis images correctly classified, totaling 108 accurate predictions and zero misclassifications.

```
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 114: InceptionResNetV2 Classification Report

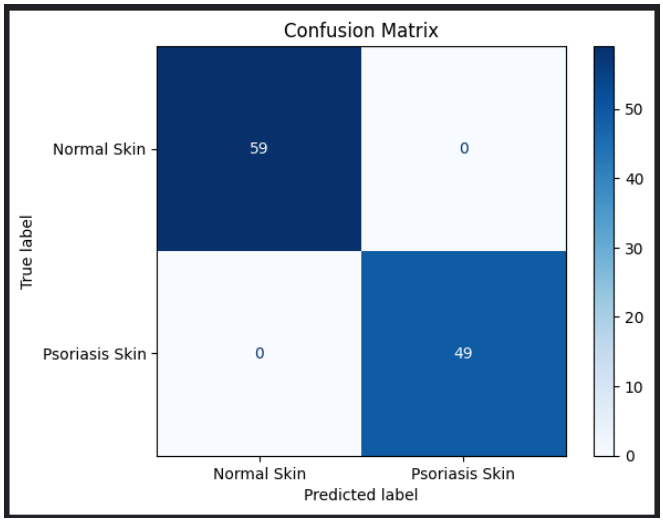


Figure 115: InceptionResNetV2 Confusion Matrix

Training and validation curves supported the model's effectiveness, with training accuracy rapidly converging to 100% and validation accuracy remaining consistently high. Both training and validation loss steadily declined without signs of overfitting, confirming the model's ability to learn efficiently and generalize well to unseen data.

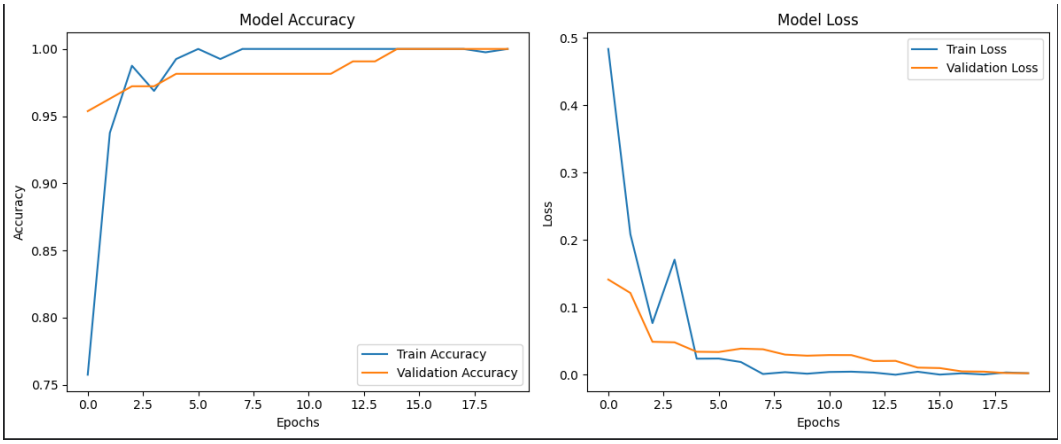


Figure 116: InceptionResNetV2 Training and Validation Accuracy/Loss Curves

3.3.2 Psoriasis Severity Classification

3.3.2.1 MobileNetV2 (Best Model)

The MobileNetV2 model delivered the best performance, achieving 80% overall accuracy in classifying severity levels. The classification report indicates high performance in the severe category (F1-score: 0.88), followed by mild (0.81) and moderate (0.70). The confusion matrix confirms that most predictions were accurate, though moderate cases were occasionally confused with mild or severe, indicating some overlap in mid-range lesion features.

```
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 117: MobileNetV2 Severity Classification Report

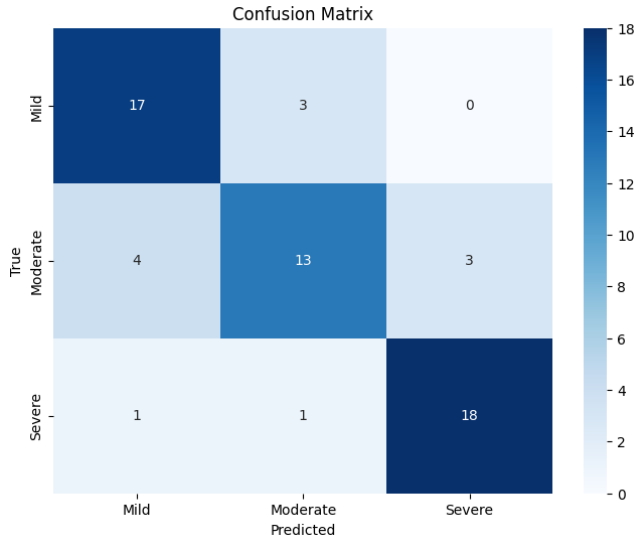


Figure 118: MobileNetV2 Severity Confusion Matrix

Training and validation curves displayed a steady improvement in accuracy and loss over 30 epochs, suggesting stable and effective learning without signs of overfitting. Additionally, random prediction results highlight the model's ability to generalize across different skin tones and lighting conditions, validating its practical usability in real-world scenarios.

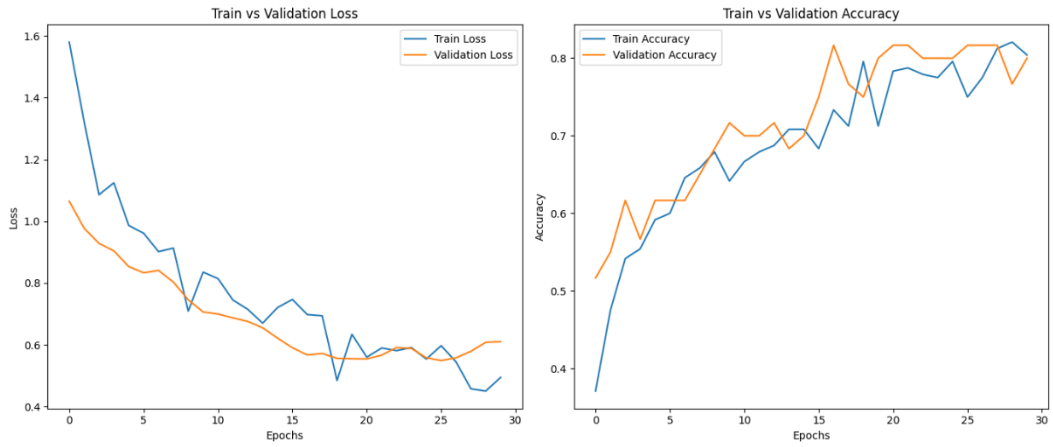


Figure 119: MobileNetV2 Severity Training and Validation Accuracy/Loss Curves

Figure 120: Random Predictions from Best Model

3.3.2.2 CNN Model

The baseline CNN model achieved an overall accuracy of 63% in classifying psoriasis severity into mild, moderate, and severe levels. According to the classification report, severe cases (level_3) showed the best performance, with a precision of 0.72 and recall of 0.90, indicating they were easier to identify. In contrast, moderate cases (level_2) were the most challenging, showing the lowest scores across metrics. The confusion matrix highlights frequent misclassifications

between mild and moderate categories, with a few errors in severe predictions. This indicates a need for improved class separation, particularly between early-stage symptoms.

```
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 121: CNN Severity Classification Report

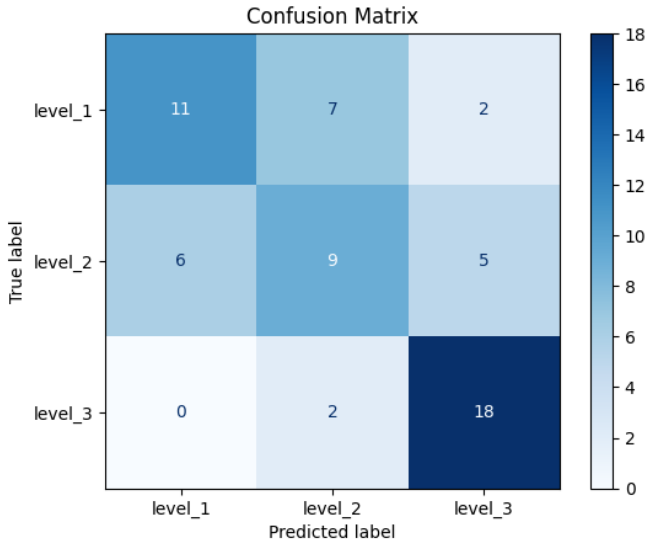


Figure 122: CNN Severity Confusion Matrix

The training and validation curves reveal fluctuations in both accuracy and loss over 30 epochs. Unstable accuracy and inconsistent loss reduction suggest the model faced difficulties in learning and generalizing. These patterns point to potential overfitting and highlight the need for enhancements in model architecture, regularization, or data augmentation strategies.

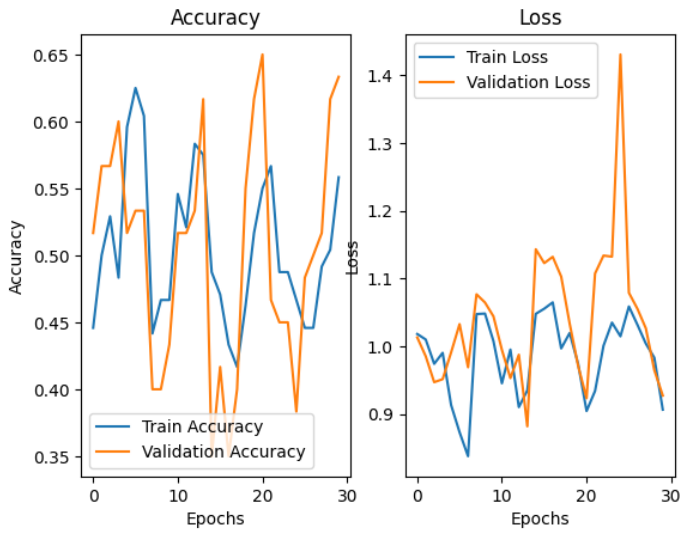


Figure 123: CNN Severity Training and Validation Accuracy/Loss Curves

3.3.2.3 EfficientNetB3 Model

The EfficientNetB3-based model also achieved 63% accuracy in psoriasis severity classification. While performance varied across classes, it showed strong results for severe cases, with an F1-score of 0.88 and precision of 0.95. However, the model struggled with mild cases, which were frequently misclassified as moderate suggesting challenges in detecting subtle early-stage lesions and possible class imbalance or feature overlap.

```
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 124: EfficientNetB3 Classification Report

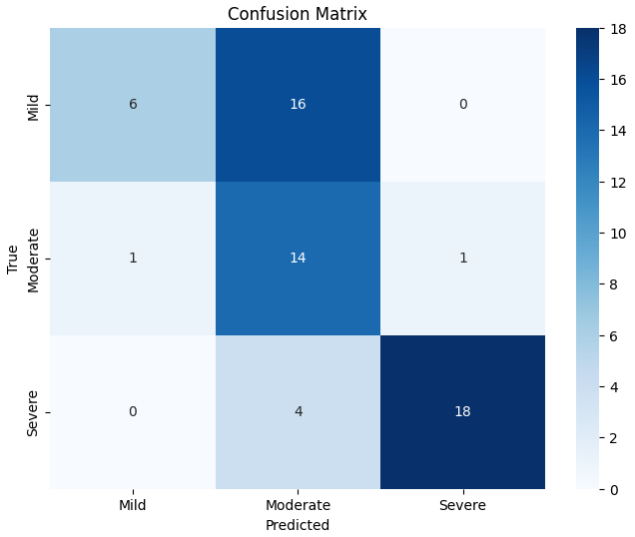


Figure 125: EfficientNetB3 Confusion Matrix

raining and validation metrics reveal signs of underfitting. Training accuracy remained low, and training loss showed noisy patterns across epochs, while validation performance improved steadily. The gap between training and validation accuracy suggests the model failed to fit the training data effectively. Improvements such as architectural tuning, longer training duration, or expanding the dataset could enhance performance.

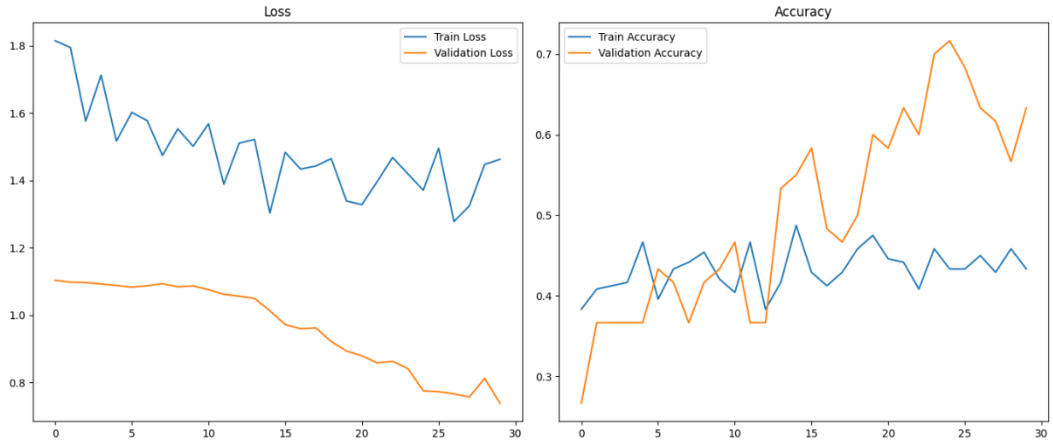


Figure 126: EfficentNetB3 Severity Training and Validation Accuracy/Loss Curves

4. SUMMARY OF EACH STUDENT'S CONTRIBUTION

Student Name	Student ID	Component
Gimmana M.R.M	IT21176456	Automated advanced acne lesion analysis and monitoring with AI
Lakshani D.M.W.S	IT21262104	Utilizing advanced ML technologies for detailed eczema diagnosis and severity assessment
Perera W.A.S.K	IT21261732	Automated AI-augmented approach to psoriasis lesion detection and accurate severity evaluation

5. CONCLUSION

This research developed an AI-driven system for acne, eczema, and psoriasis detection and severity grading with robust lesion localization and precise severity estimation. Explainability techniques enhanced model interpretability, fostering clinical trust and use. The system demonstrated outstanding generalization over diverse skin tones and conditions, allowing it to be used in real-world dermatology. Although the system is robust, misclassifications in boundary cases and dataset imbalance remain, pointing to potential areas of improvement.

Future work will include enhancing the accuracy of classification using ResNet50 tuning and investigating ensemble models, particularly for mild cases where misclassification happens more often. Generalizability will be increased by increasing diversity in the dataset by having more varied skin tones, variations of lesions, and severity variation. Enhancing post-processing functions such as Non-Maximum Suppression (NMS) and anchor box optimization can also suppress false detection for intricate dermatology cases. The second core direction is real-time deployment in clinics to incorporate AI-enabled dermatology solutions as a seamless workflow augmentation. The third and last is the introduction of longitudinal data for disease-evolution prediction that will provide individualized patient feedback to enable optimized monitoring and long-term management of skin conditions. These advances will further cement the relevance of the framework in clinic settings, yielding improved patient treatment outcomes and increasing the validity of AI-based dermatological diagnosis.

6. REFERENCES

[1] B. L. ,. Y. J. ,. L. D. ,. P. S. ,. X. L. ,. X. Y. ,. H. C. Ting Li, "Application of multimodal deep learning in the auxiliary diagnosis and treatment of dermatological diseases," in 2025 Published by Elsevier B.V. on behalf of Chinese Medical Association., 1 March 2025.

[2] †. P. H. N. 1. H. X. L. 1. L. T. N. 1. N.-T. T. 1. M. T.-T. T. 1. H. T. N. 1. N. T. V. 1. A. T. N. 1. K. S. 6. K. T. 7. T. I. 6. T. X. N. 6. H. Quan Thanh Huynh 1, "Automatic Acne Object Detection and Acne Severity Grading Using Smartphone Images and Artificial Intelligence," Diagnostics, 2022 Aug 3.

[3] A. j. S. S. Rohit yadav , "Acne Detection Care System using Deep Learning ," IEEE, 2024.

[4] O. K. ,. S. M. Krithika Naidu, "Dermato: A Deep Learning based Application for," IEEE, 2023.

[5] A. S. M. ,. S. B. B. ,. N. K. Jawad Humayun, "Localization of Acne Lesion through Template Matching," IEEE, 2012.

[6] K. M. K. ,. U. V. Kusum Meda Ravi, "Acne Classification using Deep Learning Models," IEEE, 2024.

[7] M. G. Y. J. K. K. H. G. N. R. S. S. and G. W. , "Human Skin Diseases Identification and Treatment Suggestion by Sri Lankan Ayurveda Medicine using Machine Learning," in 2023 5th International Conference on Advancements in Computing (ICAC) | 979-8-3503-5813-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICAC60630.2023.10417632, 2023.

[8] H. Nisar, Y. K. Ch'ng and Y. K. Ho, "Automatic Segmentation and Classification of Eczema Skin Lesions Using Supervised Learning," 2020 IEEE 16th International Colloquium on Signal Processing & Its Applications (CSPA), Langkawi, Malaysia, 2020, pp. 151–156. doi:10.1109/CSPA48992.2020.9068701.

[9] S. G. S. Jardeleza, J. C. Jose, J. F. Villaverde and M. A. Latina, "Detection of Common Types of Eczema Using Gray Level Co-occurrence Matrix and

Support Vector Machine," 2023 15th International Conference on Computer and Automation Engineering (ICCAE), Tokyo, Japan, 2023, pp. 231–235. doi: 10.1109/ICCAE56788.2023.10111261.

[10] M. S. Junayed, A. N. M. Sakib, N. Anjum, M. B. Islam and A. A. Jeny, "EczemaNet: A Deep CNN-based Eczema Diseases Classification," 2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS), Istanbul, Turkey, 2020, pp. 174–179. doi: 10.1109/IPAS50080.2020.9334929.

[11] R. Attar et al., "Reliable Detection of Eczema Areas for Fully Automated Assessment of Eczema Severity from Digital Camera Images," JID Innovations, vol. 3, 2023, Art no. 100213. doi: 10.1016/j.xjidi.2023.100213.

[12] H. Nisar, Y. R. Tan and Y. K. Ho, "Segmentation of Eczema Skin Lesions Using U-Net," 2020 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES), Langkawi, Malaysia, 2020, pp. 362–366. doi: 10.1109/IECBES48179.2021.9398739.

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

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] A. Sharma, “Eczema2 Dataset,” Kaggle, [Online]. Available: https://www.kaggle.com/datasets/adityush/eczema2.

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

[27] N. Chaure, "Acne Dataset", Kaggle, 2022. [Online]. Available: https://www.kaggle.com/datasets/nayanchaure/acne-dataset.

[28] Y. Wang, Z. Zhang, M. Yuan, J. Liao, and L. Zhang, "Lightweight CNN for HRRP Recognition Based on Attention Mechanism Structured Pruning," in Proc. 2023 IEEE International Radar Conference (RADAR), 2023, pp. 1–6. doi: 10.1109/RADAR54928.2023.10371061.

7. GLOSSARY

- AD (Atopic Dermatitis): A chronic inflammatory skin condition characterized by itchy, red, and swollen skin, commonly referred to as eczema.

- AI (Artificial Intelligence): The simulation of human intelligence processes by machines, particularly computer systems, to perform tasks typically requiring human intelligence.

- CBAM (Convolutional Block Attention Module): A deep learning module that enhances feature extraction by combining channel and spatial attention mechanisms, improving model focus on relevant image regions.

- CNN (Convolutional Neural Network): A class of deep learning models designed specifically for processing data with grid-like topology, such as images, commonly used for image classification and recognition.

- CSS (Cascading Style Sheets): A stylesheet language used for describing the presentation of a document written in HTML or XML, used to style and layout web pages.

- DL (Deep Learning): A subset of machine learning that uses neural networks with multiple layers to model complex patterns in data.

- EASI (Eczema Area and Severity Index): A standardized tool used to measure the severity of eczema based on the extent of affected body area and the intensity of lesions.

- GLCM (Gray Level Co-occurrence Matrix): A statistical method of examining texture that considers the spatial relationship of pixels, often used in image processing.

- HTML (HyperText Markup Language): The standard language for creating and structuring content on the web.

- IDE (Integrated Development Environment): A software application that provides comprehensive tools for software development, including code editing, debugging, and compilation.

- ML (Machine Learning): A subset of AI focused on the development of algorithms that allow computers to learn from and make decisions based on data.

- REST API (Representative State Transfer Application Programming Interface): A set of rules and conventions for building and interacting with web services, using standard HTTP methods.

- SCORAD (SCORing Atopic Dermatitis): A clinical tool used to assess the extent and severity of atopic dermatitis symptoms for standardized diagnosis and monitoring.

- SE (Squeeze-and-Excitation): A neural network block that improves feature extraction by recalibrating channel-wise feature responses, enhancing the representational power of deep learning models.

- SVM (Support Vector Machine): A supervised machine learning algorithm commonly used for classification and regression tasks by finding the hyperplane that best separates data points in feature space.

- UI (User Interface): The visual and interactive components of a digital application that facilitate user interaction and experience.

- UX (User Experience): The overall experience of a user when interacting with a digital product, including usability, accessibility, and efficiency.

- XAI (Explainable Artificial Intelligence): A subfield of AI focused on developing models that provide human-interpretable explanations for their predictions and decisions.

8. APPENDICES

Figure 127: Application logo

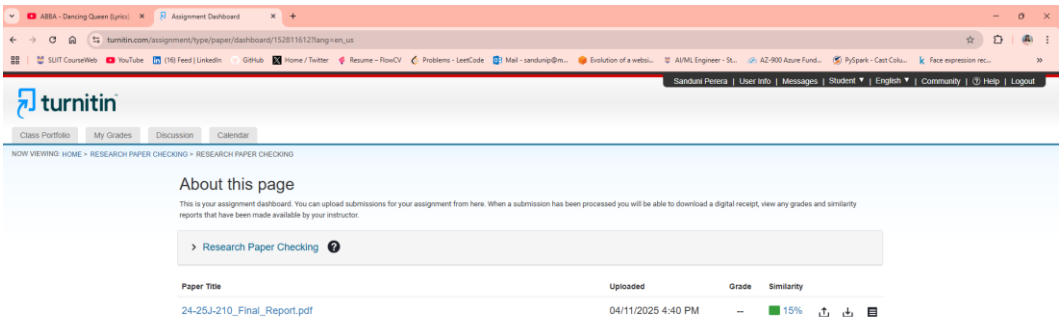


Figure 128: Turnitin Similarity

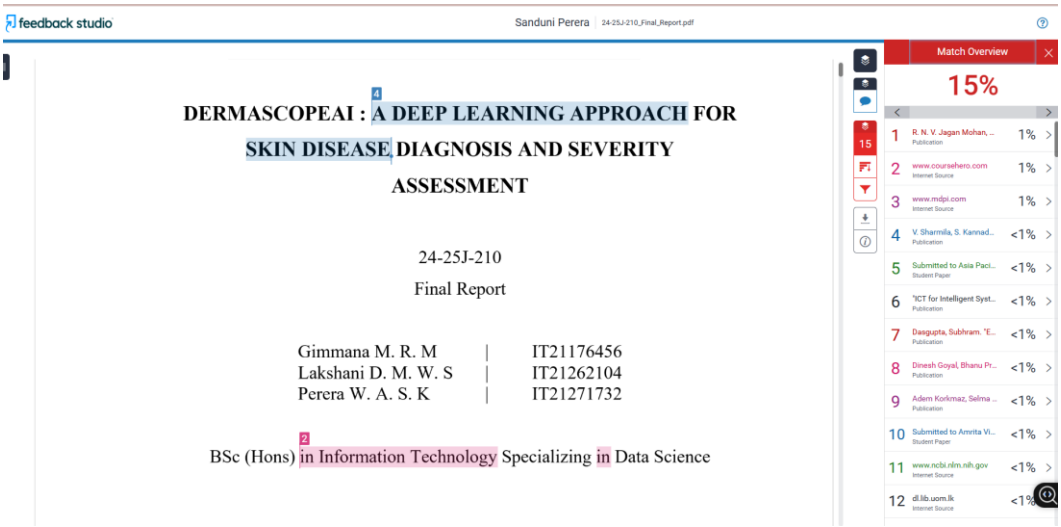


Figure 129:Turnitin Report