Psoriasis Support System Based on Semantic Segmentation AI Models

Universidade Fernando Pessoa

Toni Lopes

Faculdade de Ciência e Tecnologia

Universidade Fernando Pessoa

Thesis submitted for the obtaining of the

Master's degree in Computer Engineering - Major in Mobile Computing.

2024

Abstract

Psoriasis is a chronic inflammatory skin disease characterized by the presence of lesions that vary in shape, size, and color. Accurate segmentation of these lesions from clinical images is crucial for effective diagnosis and treatment monitoring. However, the variability in image dimensions, lighting conditions, and the presence of noise in real-world datasets complicates the segmentation task. Additionally, available datasets for psoriasis lesion segmentation are often small, making it challenging to train deep learning models effectively without overfitting.

The project aims to enhance the segmentation accuracy of psoriasis lesions by employing a range of AI models and optimizing their performance through the application of advanced pre-processing techniques and data augmentation strategies. The AI models tested include Mask R-CNN, U-Net, YOLOv8n, FCN, DeepLabV3+, BiSeNet, HRNet, PSPNet, and SegNet. Pre-processing techniques, such as bilateral smooth filters and adaptive median filters, were applied to enhance image quality and reduce noise. Furthermore, data augmentations, including random adjustments in brightness, contrast, cropping, rotation, flipping, and scaling, were used to simulate real-world variations and increase the robustness of the models. These augmentations were carefully replicated from related works to mimic their experimental setups.

The evaluation of multiple AI models for psoriasis lesion segmentation demonstrated that applying effective pre-processing and data augmentation techniques significantly improves model performance. Among the tested models, FCN achieved the highest F1 score of 0.889, using a bilateral smooth filter and translation augmentations. U-Net and SegNet followed closely, with U-Net reaching an F1 score of 0.885 and SegNet 0.880, both benefiting from similar pre-processing and augmentation strategies. Mask R-CNN and BiseNet also showed competitive results, underscoring the importance of carefully selected augmentations that mimic real-world variability. These findings highlight the value of pre-processing and augmentation in enhancing model generalization, particularly in small, diverse datasets.

In addition to the AI-based segmentation work, the project also incorporates a mobile application designed to assist patients in tracking their psoriasis lesions over time. This application enables users to monitor whether lesions

are improving or worsening, providing valuable insights for deciding when to consult a dermatologist. By helping patients recognize early signs of deterioration, the application plays a crucial role in supporting more proactive and informed healthcare decisions.

Resumo

A psoríase é uma doença crónica inflamatória da pele, caracterizada pela presença de lesões que variam em forma, tamanho e cor. A segmentação precisa dessas lesões a partir de imagens clínicas é essencial para um diagnóstico eficaz e monitorização do tratamento. No entanto, a variabilidade nas dimensões das imagens, nas condições de iluminação e a presença de ruído em conjuntos de dados do mundo real complicam a tarefa de segmentação. Além disso, os conjuntos de dados disponíveis para a segmentação de lesões de psoríase são frequentemente pequenos, o que dificulta o treino eficaz de modelos de aprendizagem profunda sem ocorrer sobreajuste.

O objetivo do projeto é melhorar a precisão da segmentação das lesões de psoríase através do uso de uma variedade de modelos de IA e otimizar o seu desempenho por meio de técnicas avançadas de pré-processamento e estratégias de aumento de dados. Os modelos de IA testados incluem Mask R-CNN, U-Net, YOLOv8n, FCN, DeepLabV3+, BiSeNet, HRNet, PSPNet e SegNet. Técnicas de pré-processamento, como filtros de suavização bilateral e filtros medianos adaptativos, foram aplicadas para melhorar a qualidade da imagem e reduzir o ruído. Além disso, foram utilizados aumentos de dados, incluindo ajustes aleatórios de brilho, contraste, recorte, rotação, espelhamento e escalamento, para simular variações reais e aumentar a robustez dos modelos. Estes aumentos foram cuidadosamente replicados de estudos relacionados para imitar as suas configurações experimentais.

A avaliação de múltiplos modelos de IA para segmentação de lesões de psoríase demonstrou que a aplicação de técnicas eficazes de pré-processamento e aumento de dados melhora significativamente o desempenho dos modelos. Entre os modelos testados, o FCN alcançou a pontuação F1 mais alta de 0,889, usando um filtro de suavização bilateral e aumentos de translação. O U-Net e o SegNet seguiram de perto, com o U-Net a atingir uma pontuação F1 de 0,885 e o SegNet 0,880, ambos beneficiando de estratégias semelhantes de pré-processamento e aumento de dados. O Mask R-CNN e o BiseNet também apresentaram resultados competitivos, sublinhando a importância de aumentos cuidadosamente selecionados que mimetizam a variabilidade do mundo real. Estes resultados destacam o valor do pré-processamento e do aumento

de dados na melhoria da generalização dos modelos, particularmente em conjuntos de dados pequenos e diversificados.

Além do trabalho de segmentação baseado em IA, o projeto também incorpora uma aplicação móvel destinada a ajudar os pacientes a acompanhar a evolução das suas lesões de psoríase ao longo do tempo. Esta aplicação permite que os utilizadores monitorizem se as lesões estão a melhorar ou a piorar, fornecendo informações valiosas para decidir quando consultar um dermatologista. Ao ajudar os pacientes a reconhecer sinais precoces de deterioração, a aplicação desempenha um papel crucial no apoio a decisões de saúde mais proativas e informadas.

Resumé

La psoriasis est une maladie de la peau inflammatoire chronique, caractérisée par la présence de lésions dont la forme, la taille et la couleur varient. La segmentation précise de ces lésions à partir d'images cliniques est essentielle pour un diagnostic efficace et un suivi adéquat du traitement. Cependant, la variabilité des dimensions des images, des conditions d'éclairage et la présence de bruit dans les ensembles de données réels compliquent cette tâche de segmentation. De plus, les ensembles de données disponibles pour la segmentation des lésions de psoriasis sont souvent de petite taille, ce qui rend difficile l'entraînement efficace des modèles d'apprentissage profond sans surajustement.

Le projet vise à améliorer la précision de la segmentation des lésions de psoriasis en employant une gamme de modèles d'IA et en optimisant leur performance à travers l'application de techniques avancées de pré-traitement et de stratégies d'augmentation des données. Les modèles d'IA testés incluent Mask R-CNN, U-Net, YOLOv8n, FCN, DeepLabV3+, BiSeNet, HRNet, PSPNet et SegNet. Des techniques de pré-traitement, telles que les filtres de lissage bilatéral et les filtres médians adaptatifs, ont été appliquées pour améliorer la qualité des images et réduire le bruit. Par ailleurs, des augmentations de données, comprenant des ajustements aléatoires de luminosité, de contraste, de recadrage, de rotation, de symétrie et de mise à l'échelle, ont été utilisées pour simuler des variations réelles et augmenter la robustesse des modèles. Ces augmentations ont été minutieusement reproduites à partir de travaux similaires afin d'imiter leurs configurations expérimentales.

L'évaluation de plusieurs modèles d'IA pour la segmentation des lésions de psoriasis a montré que l'application de techniques de pré-traitement et d'augmentation de données améliore de manière significative la performance des modèles. Parmi les modèles testés, le FCN a obtenu le score F1 le plus élevé de 0,889 en utilisant un filtre de lissage bilatéral et des augmentations de translation. U-Net et SegNet ont suivi de près, avec des scores F1 respectifs de 0,885 et 0,880, bénéficiant de stratégies similaires de pré-traitement et d'augmentation. Mask R-CNN et BiseNet ont également montré des résultats compétitifs, soulignant l'importance d'augmentations soigneusement

sélectionnées qui reproduisent la variabilité réelle. Ces résultats mettent en évidence l'importance du pré-traitement et de l'augmentation des données dans l'amélioration de la généralisation des modèles, en particulier sur des ensembles de données petits et diversifiés.

En plus des travaux de segmentation basés sur l'IA, le projet intègre également une application mobile destinée à aider les patients à suivre l'évolution de leurs lésions de psoriasis dans le temps. Cette application permet aux utilisateurs de surveiller si les lésions s'améliorent ou se détériorent, fournissant ainsi des informations précieuses pour décider quand consulter un dermatologue. En aidant les patients à reconnaître les signes précoces de détérioration, l'application joue un rôle essentiel dans le soutien à des décisions de santé plus proactives et éclairées.

I would like to dedicate this work to my parents, whose unwavering support and investment in my education and further studies have been instrumental in my academic journey. Their resilience and willpower have been a constant source of inspiration.

Acknowledgments

I extend my deepest gratitude to my mentors, Professors Rui Moreira and Christophe Soares, for their invaluable guidance and knowledge shared throughout the course of this project. Their insights have greatly contributed to its successful completion.

To my friends and colleagues, Paulo Carvalho and Rui Catarino, I am grateful for the collaborative spirit we fostered when we initially embarked on this project during our bachelor's degree. What began as a simpler exploration of segmentation techniques has now evolved into this work.

I also acknowledge Universidade Fernando Pessoa for providing a nurturing academic environment. The affordability of the coffee may have kept me going, but it was the excellent teachers, who not only shared their expertise in a clear and accessible way but also imparted their professional experience, that truly helped me grow. Their patience, dedication, and genuine concern for the students' understanding have shaped my academic and professional development in meaningful ways.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and Problem

Psoriasis is a skin condition that causes scaly patches on the skin, commonly appearing on the elbows, knees, scalp, and lower back, but it can affect any part of the body (Boehncke, 2018). While some individuals experience only small patches, in certain cases, these lesions can cause itching or even pain. This condition significantly impacts the daily lives of patients, not only affecting their physical health but also their social and psychological well-being.

Artificial Intelligence (AI), particularly when combined with image processing, is an emerging method that has gained widespread use in various fields, especially healthcare (Miller and Brown, 2017). AI tools have been developed for triaging different types of cancer (Zheng et al., 2023), and these technologies have been used by medical institutions for several years with highly positive and reliable results. These tools save lives by diagnosing diseases in their early or treatable stages (Sg et al., 2017), demonstrating the effectiveness of AI-based methods. Building on these advancements, this project aims to develop an AI-based solution to assist in the monitoring of psoriasis lesions through images.

Many psoriasis patients lack regular access to specialists due to the high cost of dermatology consultations, time constraints, or long waiting lists for appointments. This project seeks to create an accessible solution that allows patients to monitor their psoriasis lesions using a mobile device, capturing images of the lesion at any time without disrupting their daily lives.

Despite progress in AI for other medical conditions, some key challenges remain unresolved, particularly in classifying the exact location of psoriasis lesions for accurate area calculation. The similarity between skin diseases, such as eczema and psoriasis, creates a significant barrier for models to achieve high classification accuracy. These issues can be mitigated with larger, high-quality datasets, which help train AI models more effectively. Unfortunately, psoriasis-specific datasets remain limited, unlike skin cancer datasets, which are larger and have driven more significant technological advancements. Image noise, such as hair and moles, along with low image quality, also contribute to misclassification. Preprocessing and advanced image processing techniques can address these challenges, improving the model's accuracy.

1.2 Objectives

The aim of this project is to develop an Artificial Intelligence (AI)-based solution for detecting and monitoring psoriasis lesions through image processing. The system is designed to accurately detect and classify psoriasis lesions from images, tracking their size over time based on pixel count. While the project does not calculate real-world dimensions, it provides relative size measurements that help monitor lesion progression. The development also includes a mobile application to facilitate patient use, enabling easy capture and analysis of lesion images for ongoing monitoring.

To achieve these objectives, the project will involve the development of a sophisticated AI model for semantic segmentation that can handle the complexities of psoriasis detection. This model will be evaluated based on several performance metrics, including precision, recall, F1 score, and prediction time per image. Additionally, the robustness of the model will be tested under varying real-world conditions, such as different lighting scenarios and skin tones, to ensure its reliability.

A key component of the project is the creation of a mobile application that will serve as a practical tool for patients. The app will allow users to capture images of their lesions easily and will process these images to detect and measure the lesions in pixels. This will enable patients to monitor their condition conveniently and track changes over time, offering a valuable way to assess lesion progression.

Furthermore, the project aims to ensure that the system performs well in real-world situations, addressing challenges such as flash reflections and occlusions (e.g., hair). The development process will also incorporate user feedback to refine both the AI model and the mobile application, ensuring that the final solution effectively meets patient needs and expectations.

By integrating these features and evaluating the system thoroughly, the project seeks to provide a reliable and user-friendly solution for psoriasis detection and monitoring.

1.3 Document Structure

This document is divided into five chapters. Chapter 1 introduces the project's topic, outlining the problem and identifying the objectives to be achieved through the work, and describes the document structure. Chapter 2 reviews existing literature and technologies relevant to psoriasis detection and AI-based image processing, highlighting previous work and current advancements in the field. Chapter 3 details the system's architecture, functional requirements, and implementation processes. This chapter includes descriptions of the AI models used, data handling procedures, and system design. Chapter 4 corresponds to the methods and results of the evaluation phase, including performance metrics, experimental setup, and analysis of the results. Finally, Chapter 5 presents the conclusions drawn from this research and suggests potential developments for future work.

Chapter 2

State-of-Art

In recent years, the integration of artificial intelligence (AI), mobile applications, and server infrastructure has become foundational to the development of advanced digital systems. These systems enable solutions across a range of fields, from healthcare and environmental monitoring to retail and personal applications. In such projects, each component—mobile application, server, and AI—plays a critical and interconnected role.

Mobile applications provide a user-friendly interface, allowing users to interact with the system from their personal devices. These applications must be designed to not only display information but also facilitate tasks such as capturing and uploading data, particularly images, which are frequently central to AI-driven applications. Mobile apps, therefore, are a necessary point of entry for gathering and submitting data that AI models require for processing.

AI, on the other hand, provides the core analytical capability of these systems. For instance, an AI model can be trained to process images and recognize specific patterns or features within them, based on extensive datasets used during its training phase. This training is critical, as it enables the AI to understand and interpret new images with a level of accuracy that mimics human recognition skills. With robust training, the AI can identify and analyze specific features within images, enabling informed decisions based on complex visual data.

The server acts as the central processing hub, handling data transfers between the mobile application and the AI model. Servers support storage, secure access, and seamless communication within the system, ensuring that all parts work in unison. They are essential for processing and storing both image data and AI predictions, allowing results to be retrieved and updated as users interact with the mobile app.

Together, mobile applications, AI models, and server infrastructure create cohesive systems that can interpret complex data and deliver insights directly to users. Through efficient coordination between these elements, such systems demonstrate the potential of technology to provide accessible, data-driven solutions in a range of fields.

Mobile Application

A mobile application, commonly referred to as an app, is a software program designed to run on mobile devices such as smartphones and tablets. These applications provide users with various functionalities, from simple tasks like checking the weather to more complex processes like managing health records or conducting financial transactions. Mobile apps are typically downloaded from app stores and are optimized for touch interfaces, allowing users to interact with them intuitively. They enhance user experiences by offering tailored services and real-time capabilities, often using device features such as cameras, GPS, and sensors.

Flutter

Flutter is a notable programming language in mobile application development. It is an open-source UI software development kit (SDK) created by Google, allowing developers to build natively compiled applications for multiple platforms—mobile, web, and desktop—using a single codebase. This capability simplifies the development process by removing the need for separate code for different operating systems.

The primary aim of Flutter is to facilitate the creation of visually appealing applications that perform smoothly across devices. Its widget-based architecture enables high customization in user interface design, promoting flexibility. Developers can utilize prebuilt widgets that adhere to design guidelines, ensuring a consistent user experience.

A key advantage of Flutter (Bhagat, 2022) over traditional programming languages is its ability to compile to native code, enhancing speed and performance. Unlike web-based frameworks that may face latency, Flutter apps operate directly on the device's hardware, offering faster execution and responsiveness. Additionally, Flutter's hot reload feature allows developers to view code changes in real-time, making the development process more iterative and efficient.

In summary, Flutter stands out in mobile application development due to its cross-platform capabilities, focus on high-performance applications, and user-friendly development experience. This makes it an excellent choice for developers seeking to create sophisticated and engaging mobile applications in the current digital landscape.

Database Management

Database management refers to the structured organization and storage of data, enabling efficient retrieval and manipulation within an application. In mobile applications, effective database management is crucial for maintaining data integrity and enhancing user interactions.

SQLite is a prevalent choice for mobile database management. It is a lightweight, serverless SQL database engine designed for local storage on mobile devices. By operating directly within the application, SQLite simplifies data storage and management, avoiding the complexities of server-based systems. Its main goal is to provide a robust solution for efficient data handling, ensuring quick access and reliable performance.

The benefits of using SQLite include its simplicity and easy integration (Allen and Owens, 2006). Unlike traditional databases that require extensive setup and administration, SQLite is embedded within the application, facilitating rapid implementation without the need for a separate server. This characteristic makes it particularly suitable for resource-constrained mobile environments. Additionally, SQLite's support for standard SQL syntax allows developers to execute complex queries and data manipulations easily, enhancing overall data management capabilities.

By utilizing SQLite, developers can effectively manage user-generated data, such as images and preferences, ensuring a smooth user experience. This functionality is vital for tracking lesion data and synchronizing information with the server.

In summary, effective database management with SQLite is essential for efficient data handling in mobile applications. Its lightweight, serverless design enables quick implementation and a user-friendly experience while ensuring data integrity.

Network Communication

Network communication encompasses the exchange of data between a mobile application and external servers or devices over a network, playing a crucial role in functionalities such as data retrieval, image uploading, and real-time updates. Efficient communication ensures users have access to the latest information and can interact smoothly with the application.

To facilitate this communication, technologies like HTTP (Hypertext Transfer Protocol) are essential. HTTP establishes a standardized method for data transfer, enabling mobile applications to send and receive requests from servers, which is vital for operations like fetching images or submitting data.

Dio, a networking library for Flutter, enhances HTTP capabilities by offering features like interceptors and global configurations, simplifying the management of asynchronous operations. Its primary purpose is to streamline the network communication process, enabling developers to build responsive applications.

Additionally, HTTP Certificate Pinning serves as a security measure that validates SSL/TLS certificates, ensuring the mobile application only connects with trusted servers. This practice enhances application security and user trust.

In summary, effective network communication is fundamental for mobile applications, facilitating data exchange and enhancing user experience. By utilizing protocols like HTTP, libraries such as Dio, and security measures like HTTP Certificate Pinning, developers can create secure and efficient applications that meet user needs.

User Data Management

User data management involves the organized handling and protection of data related to individual users of a mobile application, including preferences, authentication credentials, and personal settings. Effective management is essential for enhancing user experience, ensuring security, and facilitating personalization.

Shared Preferences is a widely used mechanism for storing small amounts of key-value data in mobile applications. It allows for the efficient saving of user settings, such as login information or theme preferences, which enhances user convenience and personalization by maintaining state between sessions.

The primary purpose of Shared Preferences is to provide a lightweight and accessible method for managing user-specific data, especially when complex database structures are unnecessary. It allows developers to store and retrieve simple data types without imposing significant performance overhead.

In summary, effective user data management is vital for mobile application development, as it significantly impacts user satisfaction. By utilizing Shared Preferences, developers can enhance the functionality and personalization of the application, fostering a more enjoyable user experience.

Image Management

Image management in a mobile application encompasses the processes and techniques for capturing, storing, displaying, and manipulating images. Effective image management is essential for applications that rely on visual content, particularly those focused on photography, social media, or medical imaging. In this project, it is especially pertinent as users capture images of psoriasis lesions for analysis and classification.

The Camera functionality allows users to take photographs directly within the app, which is crucial for applications requiring real-time image capture. This feature streamlines the image capture process, enabling users to document images quickly without leaving the app. Integrating camera capabilities enhances user experience and supports various use cases, particularly in capturing lesion images for analysis, central to the app's purpose.

In addition to capturing images, the Photo View library facilitates displaying images in a gallery format, allowing users to browse and view their captured images easily. This tool enhances user engagement by providing a smooth and interactive viewing experience, including features like zooming, swiping between images, and other gestures that contribute to an intuitive user interface.

The primary purpose of effective image management is to enable users to capture, view, and interact with images seamlessly within the application. By utilizing both Camera and Photo View functionalities, developers can create a cohesive experience that aligns with the app's objective of monitoring and analyzing psoriasis lesions through images.

In summary, image management significantly impacts a mobile application's effectiveness, particularly in contexts where visual data is critical. By integrating robust image capture and viewing capabilities, developers enhance user experience, facilitate efficient data collection, and contribute to the app's success in achieving its intended purpose.

Permissions Management

Permissions management in a mobile application involves requesting and managing user permissions necessary for the app's functionalities. Mobile operating systems like Android and iOS enforce strict permission models to protect user privacy and data security, requiring developers to design permission requests carefully to ensure user comfort when granting access to features such as the camera, location, or storage.

In this project, the Permission Handler library simplifies the management of permissions by providing an interface for checking, requesting, and handling permissions, thereby streamlining the permission workflow. For example, before users can capture images using the camera functionality, the application must request permission to access the device's camera. Similarly, if the app needs to save images or retrieve data from storage, it must obtain the relevant permissions.

The purpose of permissions management is twofold: it ensures compliance with privacy regulations established by mobile platforms and enhances user trust by being transparent about the necessity of certain permissions and their contribution to the app's overall functionality. When users understand the reasons behind permission requests, they are more likely to grant the necessary access.

Effective permissions management also improves user experience. By utilizing the Permission Handler library, developers can proactively check for permissions before accessing features, allowing the app to guide users through the permission-granting process and reducing interruptions during critical tasks like capturing images of psoriasis lesions.

In summary, permissions management is vital in mobile application development, particularly when handling sensitive user data. By employing libraries like Permission Handler, developers can effectively manage permissions, foster user trust, and ensure the application functions as intended while respecting user privacy. This management is essential for providing a seamless user experience and enhancing the application's overall reliability.

User Interface

The user interface (UI) of a mobile application encompasses the visual elements and interactive features that allow users to engage with the app. A well-designed UI is crucial as it directly influences user experience (UX), determining how easily users navigate the app and perform desired tasks. A good UI is not only aesthetically pleasing but also intuitive, enabling users to achieve their goals with minimal friction.

In this project, the Body Part Selector is a key component of the UI, assisting users in selecting specific areas of the body where psoriasis lesions may be present. By offering a clear and user-friendly interface, the Body Part Selector facilitates accurate image capture and ensures users can easily identify and specify the locations of their lesions, which is essential for subsequent AI model analysis.

In conclusion, the user interface is a vital aspect of the mobile application that significantly impacts the overall user experience. By focusing on user-centered design principles, particularly with features like the Body Part Selector, developers can create an intuitive and engaging interface that facilitates effective interaction between users and the application. Thoughtful UI design enhances user satisfaction and supports the primary objectives of the application, such as accurately capturing images and monitoring the progression of psoriasis lesions.

Cloud and Data Synchronization

Cloud and data synchronization ensures consistent data updates and accessibility across devices through cloud services, enhancing user experience in mobile applications that handle user-generated content, such as images. In this project, a server-client architecture is utilized to synchronize images and data between the mobile app and backend server. This setup allows real-time updates, enabling users to access their information seamlessly across different devices.

The benefits of this synchronization include:

- **Consistent User Experience:** Users can access lesion images and data from any device, facilitating easier management and sharing with healthcare providers.
- **Data Integrity and Security:** Storing data in the cloud reduces the risk of data loss from device theft or failure, while robust security measures protect sensitive medical information.
- **Advanced Functionalities:** Cloud synchronization allows for AI-driven analytics, enabling timely feedback on conditions and remote monitoring by healthcare professionals.

In summary, cloud and data synchronization is essential for providing seamless access to user data, enhancing user experience, ensuring data integrity, and supporting advanced features in managing psoriasis lesions effectively.

Server

A server is a powerful computing system designed to manage network resources and facilitate communication between clients, such as mobile applications or web browsers, and databases. The primary purpose of a server is to process requests from clients, deliver resources, and manage data efficiently. In the context of mobile applications, servers play a crucial role in storing user-generated content, processing data, and providing functionalities like authentication and data synchronization. By acting as a centralized hub, servers ensure that users can access their information consistently and securely, regardless of the device they are using.

Python

Python is a high-level, interpreted programming language known for its readability and versatility. Its design philosophy emphasizes code simplicity and clarity, making it an excellent choice for both beginners and experienced developers. One of the main advantages of Python is its extensive standard library, which provides a wealth of modules and functions that streamline development tasks. Additionally, Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming, allowing developers to choose the approach that best suits their project requirements. This flexibility, combined with a vibrant community and a rich ecosystem of third-party libraries, has made Python a popular language for web development, data analysis, artificial intelligence, and more.

Framework

A framework is a set of tools, libraries, and best practices designed to facilitate software development by providing a structured foundation for building applications. Frameworks streamline the development process by offering pre-defined templates and conventions, allowing developers to focus on writing application-specific code rather than dealing with low-level details. They promote code reusability, maintainability, and scalability, enabling developers to create complex applications efficiently. By adhering to a framework's conventions, developers can collaborate more effectively and ensure that their codebase is organized and coherent.

Flask

Flask is a lightweight web framework for Python that enables developers to build web applications quickly and efficiently. It is designed to be simple and flexible, making it an ideal choice for small to medium-sized projects. One of the key advantages of Flask is its minimalistic approach, allowing developers to add only the components they need without unnecessary complexity. Flask also provides built-in support for various features, such as routing, request handling, and session management, while remaining highly extensible through the use of third-party libraries. This flexibility allows developers to tailor their applications according to specific requirements, making Flask a popular choice for creating RESTful APIs and web applications.

REST API and Flask RESTful

A REST API (Representational State Transfer Application Programming Interface) is a set of rules and conventions that allows different software systems to communicate with each other over the internet using standard HTTP methods, such as GET, POST, PUT, and DELETE. REST APIs are stateless, meaning that each request from a client contains all the information needed to understand and process it, enhancing scalability and reliability. Flask RESTful is an extension of Flask that simplifies the creation of REST APIs by providing tools and features specifically designed for building and managing API endpoints. With Flask RESTful, developers can easily define resources, manage routing, and handle serialization, making it a powerful tool for building robust and efficient APIs.

PostgreSQL

PostgreSQL is an advanced, open-source relational database management system that emphasizes extensibility and standards compliance. It is known for its ability to handle complex queries, large datasets, and high transaction volumes efficiently. One of the main advantages of PostgreSQL is its support for advanced data types and performance optimization features, which enable developers to create sophisticated applications with diverse data storage needs. Additionally, PostgreSQL offers robust security features, such as user authentication and role-based access control, ensuring that sensitive data remains protected. Its strong community support and a wealth of available extensions further enhance its capabilities, making PostgreSQL a popular choice for modern web applications.

Flask SQLAlchemy

Flask SQLAlchemy is an extension for Flask that integrates the SQLAlchemy ORM (Object Relational Mapper) into Flask applications. SQLAlchemy provides a powerful and flexible way to interact with relational databases by allowing developers to define database models as Python classes. One of the main advantages of Flask SQLAlchemy is that it abstracts the complexity of database interactions, enabling developers to perform CRUD (Create, Read, Update, Delete) operations with simple and intuitive Python code. This integration simplifies database management within Flask applications, allowing for seamless interactions with PostgreSQL or other supported databases. By using

Flask SQLAlchemy, developers can focus on building application logic while benefiting from the efficiency and reliability of SQLAlchemy's powerful ORM capabilities.

PBKDF2_SHA256

PBKDF2_SHA256 is a secure password hashing algorithm that transforms an original password into a fixed-size hash. It employs a configurable number of iterations, enhancing security by making it more computationally intensive for attackers to crack passwords through brute-force methods. Additionally, it incorporates a unique salt—random data added to the password before hashing—ensuring that identical passwords yield different hashes. This combination of salting and multiple iterations makes PBKDF_SHA256 a robust method for protecting passwords against unauthorized access.

JWT Tokens

JWT (JSON Web Tokens) are an open standard for securely transmitting information between parties as a JSON object. In the context of web applications, JWTs are commonly used for user authentication and authorization. The primary advantage of JWTs is that they enable stateless authentication, meaning that once a user is authenticated, their session information is encoded into a token and stored client-side, eliminating the need for server-side session storage. This approach enhances scalability and reduces server load. Additionally, JWTs are compact and can be easily transmitted in HTTP headers, making them suitable for modern web applications. The use of JWTs ensures that user sessions are secure and can be verified without requiring continuous communication with the database, providing a seamless experience for users.

Flask JWT

Flask JWT is an extension of the Flask web framework that enables the implementation of JSON Web Token (JWT) authentication. This mechanism allows for secure transmission of information between parties as a JSON object. By using JWT, Flask applications can authenticate users and ensure that sensitive data is transmitted securely. The extension simplifies the process of creating, validating, and refreshing tokens, thereby enhancing the security of web applications through stateless authentication.

Artificial Intelligence

Understanding Artificial Intelligence (AI)

Artificial Intelligence (AI) refers to the capability of a machine to imitate intelligent human behavior. It encompasses the development of algorithms and models that allow machines to perform tasks such as reasoning, problem-solving, and decision-making, which are typically associated with human intelligence. In various fields, including healthcare, AI has the potential to revolutionize how data is processed and analyzed, leading to enhanced decision-making processes.

An AI model is a specific implementation of an AI algorithm that learns from data to perform tasks. Models are trained on datasets to recognize patterns and make predictions based on the input data. Several AI models are particularly relevant to this project, including:

- **BiSeNet (Bilateral Segmentation Network) (Yu et al., 2020):** This model is designed for efficient real-time segmentation by utilizing both low-level and high-level features to achieve precise boundary detection.

- **DeepLabV3+ (Yu et al., 2022):** An advanced segmentation model that employs atrous convolution and pyramid pooling to capture multi-scale context information, improving segmentation performance.

- **FCN (Fully Convolutional Network) (Shelhamer et al., 2017):** This model adapts traditional convolutional neural networks to output spatial maps, enabling pixel-wise segmentation.

- **HRNet (High-Resolution Network) (Xu et al., 2020):** HRNet maintains high-resolution representations through parallel convolutions, ensuring accurate localization of features.

- **Mask R-CNN (Zhang et al., 2020):** An extension of Faster R-CNN, Mask R-CNN adds a branch for predicting segmentation masks, allowing for instance segmentation of objects.

- **PSPNet (Pyramid Scene Parsing Network) (Yang and Guo, 2022):** PSPNet utilizes a pyramid pooling module to gather contextual information from different regions, enhancing scene understanding.

- **SegNet (Badrinarayanan et al., 2015):** A model that employs an encoder-decoder architecture, SegNet is effective for pixel-wise segmentation and reconstructs high-resolution outputs from low-resolution inputs.

- **U-Net (Su et al., 2021):** This model is particularly popular in biomedical image segmentation, utilizing skip connections to retain spatial information during the learning process.

- **YOLOv8n (You Only Look Once) (Rasi et al., 2023):** A real-time object detection model that achieves high accuracy and speed by predicting bounding boxes and class probabilities from full images.

Image Classification by AI Models

AI models classify images by analyzing the patterns and features present in the input data. During the training process, models learn to associate specific features with corresponding labels, enabling them to predict the class of new, unseen images. Classification typically involves comparing the input image's characteristics against learned representations to determine the most likely label.

Semantic segmentation (Lateef and Ruichek, 2019) is an advanced image classification method that labels each pixel by the object or region it represents, providing a detailed, pixel-level understanding of an image's content. This approach is crucial in fields requiring precise boundaries, such as medical imaging and autonomous driving.

For binary or multi-label segmentation tasks, the sigmoid activation function is commonly used in the output layer. Unlike softmax, which assigns each pixel to only one class, sigmoid treats each class separately, giving each pixel a probability score that reflects how likely it is to belong to the target class.

Inputs and Outputs of AI Models

The inputs for an AI model (Bandi et al., 2023) are the features extracted from the dataset, which in this case consists of images of psoriasis lesions. The outputs are the predictions made by the model, indicating the classification of the lesions, such as whether they are present or absent and their severity. Image processing techniques, such as image segmentation and binary segmentation, are crucial for defining the outputs. Image segmentation refers to dividing an image into meaningful parts, while semantic segmentation categorizes each pixel into classes, such as lesion and non-lesion areas. The outputs of segmentation can vary based on the model employed, leading to different degrees of accuracy and precision in identifying lesions.

AI Model Configuration

AI model configuration (Bandi et al., 2023) involves setting the parameters that dictate how a model learns from data. This includes defining the learning rate, which determines the speed at which the model updates its parameters during training. An epoch represents one complete pass through the training dataset, and multiple epochs are often required to achieve optimal learning. Additionally, the training image size refers to the dimensions of input images; many models require images to be resized to consistent dimensions for effective training and accurate results.

Benchmarking AI Models

A benchmark is a standard or point of reference used to evaluate the performance of AI models. By establishing benchmarks, it becomes possible to systematically compare the performance of different models based on various criteria, such as accuracy, efficiency, and speed. This comparison is essential for determining which model performs best under specific conditions, guiding future improvements and optimizations in the training and evaluation processes.

Datasets in AI

A dataset is a structured collection of data used for training and evaluating AI models. In the context of this project, the dataset consists of images of psoriasis lesions, which provide the necessary examples for the model to learn from. Dataset masking or labeling refers to the process of annotating images with relevant information, such as defining the boundaries of lesions. This process is critical for supervised learning, where models are trained on labeled examples to accurately predict outcomes.

AI Training and GPU Utilization

AI training is the process of teaching a model to recognize patterns and make predictions by exposing it to a labeled dataset. During training, the model adjusts its internal parameters to minimize prediction errors. The use of Graphics Processing Units (GPUs) in this context provides significant advantages, as GPUs are designed to perform parallel processing, allowing for faster training times and the ability to handle larger datasets more efficiently.

AI Evaluation and Metrics

AI evaluation involves assessing the performance of a trained model on a separate dataset to determine its accuracy and effectiveness. This process is crucial for understanding how well the model generalizes to new data. Various AI metrics (Bandi et al., 2023), including accuracy, recall, and the F1 score, provide quantitative measures of model performance. The accuracy metric indicates the proportion of correct predictions, while recall measures the model's ability to identify all relevant instances. The F1 score combines precision and recall to offer a balanced evaluation of model performance. Scikit-learn (sklearn) is a widely used library in Python that provides tools for model evaluation and metrics calculation.

Pre-processing and Data Augmentation Techniques

Pre-processing techniques are applied to raw data to enhance its quality and prepare it for analysis. The goal of these techniques is to optimize the learning process and improve model accuracy (Ivan et al., 2023).

- **Adaptive Median Filter** The adaptive median filter aims to reduce image noise while preserving critical edge information. Higher smoothing levels result in more effective noise reduction but at the cost of potentially eliminating fine details, which are essential for accurate lesion segmentation.

- **Background Masking / Region of Interest** Although these terms may seem interchangeable, in this context, background masking and defining a region of interest (ROI) were employed to isolate lesion areas and exclude irrelevant background elements.

- **Bilateral Smoothing Filter** The bilateral smoothing filter is applied to the images to reduce noise while preserving edge details. By retaining the boundaries of the lesion areas, this filter facilitates more accurate delineation of lesion edges.

- **Binary Mask** The binary mask technique converts the input image into a binary format, simplifying the segmentation task by classifying each pixel as either part of the lesion or not. This reduction of complexity assists in the training of segmentation models, enhancing their focus on the essential task of lesion identification.

- **Color Histogram** By analyzing the distribution of colors within the images, the color histogram technique aids in distinguishing between lesion and non-lesion regions. Understanding the intensity variations in the color spectrum allows the system to more effectively identify potential lesion areas based on their color characteristics.

- **Contrast Enhancement** Contrast enhancement is utilized to improve the visibility of features within the images, particularly by highlighting lesions against varying skin tones.

- **Dilation and Erosion** Morphological operations such as dilation and erosion are applied to manipulate the shape and structure of the lesion regions. Dilation expands the boundaries of the lesions, while erosion reduces them. These operations were used to refine the segmentation results, particularly when handling lesion areas that were difficult to distinguish from surrounding skin.

- **Gaussian Filter** The Gaussian filter, a widely-used blurring technique, reduces noise and softens image details. While it blurs certain features, this technique is beneficial in eliminating minor noise artifacts that could negatively affect segmentation performance in specific cases.

- **Grayscale Conversion** Grayscale conversion prepares the images for techniques that operate on single-channel grayscale images, such as adaptive median filtering and texture analysis methods. This conversion simplifies the image structure, focusing on intensity values for subsequent processing.

- **Hair Removal** Hair removal is a vital pre-processing step in image analysis, especially in medical imaging, where hair can introduce noise that obscures relevant details. By removing hair artifacts, the image is clarified, enabling AI models to focus accurately on the target features, ultimately enhancing diagnostic precision.

- **Resize** Resizing images is essential for preparing data for AI models, as it standardizes input dimensions across the dataset. Uniform image sizes optimize model efficiency, reduce processing time, and improve consistency in analysis, allowing AI models to process and interpret images more effectively.

- **Sharpening** Sharpening filters are applied to enhance edge definition within images, which aids in delineating lesion boundaries with greater clarity. This process improves the contrast of edges, allowing finer details of the lesions to stand out, thereby facilitating more accurate analysis and segmentation in medical imaging.

Data augmentation techniques artificially expand the training dataset by creating modified versions of existing images. The objective of data augmentation is to improve the robustness of the model by exposing it to a wider variety of input conditions (Shorten and Khoshgoftaar, 2019). For instance, the crop technique involves randomly cropping sections of images, providing the model with diverse perspectives on the data.

- **Brightness** This technique modifies the brightness of the image within a controlled range, simulating different lighting conditions. Such adjustments improve the model's robustness, allowing it to better adapt to various brightness levels commonly encountered in real-world settings.

- **Contrast Enhancement** Contrast enhancement increases the distinction between light and dark areas in the image, thereby rendering critical features more pronounced. This process improves the model's capacity to differentiate between various structures or objects, particularly in images characterized by inherently low contrast.

- **Crop** Random cropping removes portions of the original image, exposing the model to varying segments of the input data. This technique encourages the model to focus on multiple areas, thereby reducing reliance on specific features located in fixed positions and enhancing generalization.

- **Flip** Flipping the image horizontally or vertically introduces variability in orientation, allowing the model to recognize objects from different perspectives. This augmentation aids in ensuring that the model does not rely solely on the object's original orientation for accurate identification.

- **Gaussian Blur** Gaussian blur introduces a softening effect that simulates slight defocus or noise, enhancing the model's ability to handle imperfect or blurry images, which are common in real-world applications.

- **Rotate** Images are rotated by random angles. This augmentation helps the model become less sensitive to the orientation of objects, enabling it to effectively learn from objects presented at various angles.

- **Scale** Scaling enlarges the image, allowing the model to learn from images of different resolutions and sizes. This augmentation enhances the model's capacity to detect features at multiple scales, improving recognition capabilities across various levels of detail.

- **Sharpness Enhancement** This technique enhances the clarity and definition of image edges and fine details, improving the model's capacity to detect small, intricate features, which are critical in tasks such as segmentation where precision is paramount.

- **Shear** Shearing introduces an angular distortion in the image, enhancing the model's ability to generalize across slight perspective changes or distortions present in the objects.

- **Translate** Translation shifts the image in a random direction. This technique ensures that the model does not become overly reliant on objects being situated in fixed regions of the image, thereby improving its generalization capabilities.

- **Zoom In** Zooming in focuses on the central region of the image. This technique aids the model in capturing finer details, thereby enhancing its ability to learn from both overall patterns and minute features of the objects.

Roboflow is an essential tool utilized for dataset masking, preprocessing, and applying data augmentation techniques. It simplifies the process of managing datasets and enhances the training pipeline. OpenCV is a library used for implementing preprocessing and data augmentation techniques, offering a robust set of tools for image manipulation.

TensorFlow is another library leveraged in this project, providing a framework for building and training machine learning models. Keras, built on top of TensorFlow, simplifies the creation and training of neural networks, making it easier to implement complex models with less code.

2.1 Technologies in Psoriasis Support using Semantic Segmentation AI Models

The project leveraged a variety of advanced technologies to develop an AI-based solution for psoriasis lesion detection and monitoring. For the back-end infrastructure, a Python Flask server was implemented, operating with HTTPS to ensure secure communications. This server handles the processing of lesion images, storing and retrieving data, and synchronizing user information across devices. The secure authentication system, built on JSON Web Tokens (JWT), provides functionalities for user registration, login, logout, and account recovery, ensuring data privacy and secure access.

On the front-end, the mobile application was developed using Flutter, a cross-platform framework. The app allows users to capture images of their lesions and track their progression. It also features a gallery for viewing lesion history and synchronizes images and lesion progress with the server when the user logs in from another device. The application was designed to provide a user-friendly experience, displaying contours of the previously captured lesion to help users maintain consistent camera-to-lesion distance, ensuring accurate size tracking.

While the app is robust in terms of synchronization and data tracking, it currently does not support offline use, requiring internet connectivity for data access and updates. The integration of these technologies forms a cohesive system aimed at empowering patients to monitor their psoriasis effectively, while the AI models provide detailed segmentation results based on real-world image variations such as lighting and skin tone.

2.1.1 Artificial Intelligence Technologies

Artificial Intelligence Models

In this project, several state-of-the-art AI models were employed for benchmarking, including BiSeNet (Bilateral Segmentation Network), DeepLabV3+, FCN (Fully Convolutional Network), HRNet (High-Resolution Network), Mask R-CNN, PSPNet, SegNet, U-Net, and YOLOv8n. These models were selected for their proven effectiveness in semantic segmentation tasks, particularly for medical image analysis. They were trained and evaluated using various pre-processing and data augmentation techniques to determine the optimal configuration for accurate lesion segmentation. The comparative analysis among these models aims to identify which configuration yields the best evaluation metrics for this specific application.

TensorFlow served as the primary framework for running these models, providing a robust environment for developing and training deep learning architectures. However, YOLOv8n utilized the Ultralytics framework, which is specifically optimized for real-time object detection tasks. To further enhance performance, CUDA and cuDNN were utilized for Windows Subsystem for Linux (WSL), enabling accelerated computation on NVIDIA GPUs. This setup is crucial for training deep learning models efficiently, significantly reducing the time required for model training and evaluation.

Pre-processing and Data augmentation

Roboflow was leveraged as a powerful platform for dataset masking and labeling, streamlining the processes of pre-processing and data augmentation. This platform simplifies the preparation of training datasets by providing tools for annotating images, which is critical for supervised learning tasks. Additionally, it facilitates the application of pre-processing and data augmentation techniques, enhancing the dataset's diversity and improving the models' generalization capabilities.

The pre-processing techniques employed for benchmarking include Adaptive Median Filter, Background Masking / Region of Interest, Bilateral Smoothing Filter, Binary Mask, Color Histogram, Contrast Enhancement, Dilation and Erosion, Gaussian Filter, Grayscale Conversion, Hair Removal, Resize, and Sharpening.

While the specific functions of these techniques have been previously discussed in the state-of-the-art section, it is important to highlight their collective role in refining image quality and enhancing model performance. The OpenCV library was utilized to implement these techniques effectively, providing a comprehensive set of tools for image processing. For hair removal, DullRazor was specifically applied to minimize artifacts that could negatively affect segmentation results.

Data augmentation techniques used for benchmarking comprised Brightness, Contrast Enhancement, Crop, Flip, Gaussian Blur, Rotate, Scale, Sharpness Enhancement, Shear, Translate, and Zoom In. These methods were chosen to artificially expand the training dataset, thereby enhancing the model's ability to generalize to unseen data. The libraries employed for these augmentations included OpenCV and PIL (Pillow), with PIL being particularly utilized for Gaussian blur applications. These augmentations play a crucial role in improving the robustness and accuracy of the AI models in segmenting psoriasis lesions.

OpenCV for Pre-processing and Image Processing

OpenCV is employed in the server-side application to implement various pre-processing techniques, data augmentation methods, and image processing tasks. It plays a critical role in preparing the image data for lesion segmentation, enabling the simulation of real-world conditions that improve the robustness of the models. By utilizing OpenCV, the application ensures high-quality image analysis, which is essential for accurate lesion detection and segmentation in medical contexts.

Roboflow for Dataset Management

Roboflow is utilized for effective dataset management throughout the project. It streamlines the processes of masking and labeling images, splitting datasets into training and validation subsets, and applying necessary pre-processing and data augmentation techniques. This platform aids in efficiently organizing and preparing the dataset, thereby facilitating the training of AI models and improving overall workflow efficiency.

TensorFlow Keras for Model Development

TensorFlow Keras serves as the primary framework for configuring and developing the AI models involved in lesion segmentation. By utilizing this library, the application is capable of implementing various deep learning architectures that are critical for accurately identifying lesions in medical images. TensorFlow Keras enables the seamless integration of advanced techniques and customizations, thereby enhancing the model's performance and adaptability.

Scikit-learn for Metric Calculation

Scikit-learn is used to calculate performance metrics such as precision, recall, and F1 score, which are essential for evaluating the effectiveness of the AI models. By leveraging Scikit-learn for these calculations, the project ensures a systematic assessment of model performance, allowing for data-driven decisions in model refinement and selection.

Matplotlib for Image Visualization

Matplotlib is utilized for visualizing, analyzing, reading, and saving images throughout the model development and evaluation process. It enables effective visualization of results, aiding in the interpretation of model outputs and the impact of pre-processing techniques. This capability supports a deeper understanding of model performance and facilitates the communication of findings through clear and informative visual representations, while the functionality for reading and saving images ensures that the workflow remains efficient and organized.

2.1.2 Server-Side Technologies

Python Flask (Web Framework)

Python Flask serves as the foundational web framework for the server-side components of this project. Its lightweight and flexible design facilitates rapid development and deployment, making it suitable for various web applications. Flask's modular architecture allows for easy integration of additional functionalities, which is crucial for maintaining a clean and efficient codebase.

HTTPS for Secure Communication

To ensure secure communication between clients and the server, HTTPS is implemented. This protocol encrypts data exchanged over the network, protecting sensitive information from potential eavesdropping and tampering. The adoption of HTTPS enhances the overall security posture of the application, ensuring that user data remains confidential and secure during transmission.

REST API Routes and Endpoints

The application employs a RESTful API using Flask Restful, facilitating efficient communication between the client and the server. The following routes have been established for handling authentication, lesion data management, and image uploads:

- **/auth/changePassword** - Handles password change requests using the POST method.
- **/auth/login** - Manages user login through the POST method.
- **/auth/loginToken** - Authenticates users with tokens using the HEAD and GET methods.
- **/auth/logout** - Logs users out using the POST method.
- **/auth/refreshToken** - Refreshes authentication tokens with the POST method.
- **/auth/registration** - Registers new users via the POST method.
- **/auth/reset** - Initiates password reset requests using the POST method.
- **/auth/reset/code** - Confirms reset code via the POST method.
- **/auth/reset/password** - Resets the user's password through the POST method.
- **/auth/updateProfile** - Updates user profile data via the POST method.
- **/lesion** - Creates new lesion entries using the POST method.

- **/pictures** - Retrieves body part lesions using the POST method. This is preferred because POST allows for the transfer of larger data payloads, such as images and lesion details, which may not be efficiently handled by GET requests due to size limitations.
- **/picture** - Uploads pictures using the POST method.

SQLAlchemy/Flask SQLAlchemy for Databases

Database interactions are managed using SQLAlchemy, with an emphasis on Flask SQLAlchemy, which simplifies the use of SQLAlchemy within Flask applications. This Object-Relational Mapping (ORM) tool abstracts the complexities of database interactions, enabling seamless data management and manipulation. By utilizing Flask SQLAlchemy, the project can efficiently handle data storage and retrieval for the application.

Table	Column	Type	Description
users	id	INTEGER	Primary Key
	first_name	STRING(15)	User's first name (unique)
	last_name	STRING(15)	User's last name (unique)
	email	STRING(120)	User's email (unique)
	password	STRING(120)	User's hashed password
lesions	id	BIGINTEGER	Primary Key
	user_id	BIGINTEGER	Foreign Key referencing *users(id)*
	body_part_id	INTEGER	Foreign Key referencing *bodyparts(id)*
	pictures_ids	ARRAY(INTEGER)	List of related picture IDs
	created_at	TIMESTAMP	Lesion creation timestamp
	updated_at	TIMESTAMP	Lesion last update timestamp
pictures	id	BIGINTEGER	Primary Key
	local_path	STRING(200)	Local path of the image
	local_contours_path	STRING(200)	Local path of the contour data
	local_pred_contours_path	STRING(200)	Local path of predicted contour data
	path	STRING(200)	Server path of the image
	contours_path	STRING(200)	Server path of the contour data
	pred_contours_path	STRING(200)	Server path of predicted contour data
	created_at	STRING(30)	Image creation timestamp
	size	FLOAT	Size of the image (MB)
	contours_vertices	JSONB	Contour vertices for the lesion
reset_passwords	id	INTEGER	Primary Key
	user_id	BIGINTEGER	Foreign Key referencing *users(id)*
	date	STRING(30)	Date of the password reset request
	code	BIGINTEGER	Generated reset code
	code_match	BOOLEAN	Indicates if the code matches for the request
	old_password	STRING(120)	User's previous password (hashed)
revoked_tokens	id	INTEGER	Primary Key
	jti	STRING(120)	Unique token identifier (JWT ID)

Table 2.1: Server Database Schema

PBKDF2_SHA256 for Password Hash Generation

For password security, the project employs PBKDF2_SHA256, a hashing algorithm designed to securely hash user passwords. While some may debate whether this qualifies as a technology, its implementation is critical for maintaining user security and privacy. This method adds an additional layer of security, protecting user credentials from potential breaches.

Flask JWT and JWT (JSON Web Tokens) for Authentication

For authentication, the project employs Flask JWT (JSON Web Tokens), a method that allows secure transmission of information between parties as a JSON object. This mechanism enables users to authenticate their identity without exposing their credentials with every request, thereby enhancing the security of the application. JWT also facilitates stateless authentication, simplifying the server-side implementation.

Server-Client Architecture for Image and Data Synchronization

The application employs a server-client architecture where the client sends the IDs of the images stored locally, and the server checks for missing images or updated lesion information. If discrepancies are found, the server sends the necessary image bytes and updated data, ensuring consistency across devices.

2.1.3 Client-Side Mobile Application Technologies

Flutter

Flutter is a modern mobile application development framework created by Google, designed for building natively compiled applications for mobile, web, and desktop from a single codebase. Its expressive UI, flexible architecture, and high performance make it a suitable choice for developing responsive mobile applications aimed at tracking psoriasis lesions. The framework leverages Dart, a language optimized for fast execution and a reactive programming model, allowing developers to create visually appealing applications with ease.

Database Management

SQLite (sqflite) The sqflite package provides a way to access SQLite databases in Flutter applications. This technology is utilized for managing and storing critical data related to images and lesions efficiently. The application utilizes a structured database to manage and store critical data related to images and lesions. The database includes two primary tables:

- **Pictures Table:** This table stores information regarding each captured image, including its path, contour vertices, and developmental metrics, as well as the date it was created.
- **Lesions Table:** This table captures essential details about each lesion, including user information, body part affected, lesion size, upload status, and associated image identifiers, ensuring comprehensive tracking of patient conditions.

Table	Column	Type	Description
pictures	id	INTEGER	Primary Key
	path	TEXT	Path to the image
	contoursVertices	TEXT	Contours of the lesion
	developmentPerc	DOUBLE	Development percentage
	developmentPercTotal	DOUBLE	Total development percentage
	createdAt	DATETIME	Creation date
lesions	id	INTEGER	Primary Key
	email	TEXT	User's email address
	bodyPart	TEXT	Affected body part
	size	DOUBLE	Size of the lesion
	canUpload	TEXT	Upload capability
	canUploadDatetime	DATETIME	Upload date and time
	createdAt	DATETIME	Creation date
	updatedAt	DATETIME	Last update date
	picturesIds	TEXT	Associated picture IDs

Table 2.2: Client Database Schema

Network Communication

HTTP The HTTP package provides a simple way to send HTTP requests and receive responses in Dart applications. This library is essential for enabling network communications, allowing the mobile application to interact with remote servers for data retrieval and submission. By utilizing HTTP, the app can seamlessly manage user data and sync lesion information with the backend.

Dio Dio is a powerful HTTP client for Dart that supports Interceptors, FormData, Request cancellation, and File downloading. It enhances the mobile application's capabilities by providing advanced networking features, such as handling complex API requests and responses, thereby improving data management and enhancing user experience.

HTTP Certificate Pinning The http_certificate_pinning package implements SSL certificate pinning to secure communications between the mobile application and the server. By validating server certificates against known trusted certificates, this technology ensures that the application maintains secure and trusted connections, safeguarding user data during transmission.

User Data Management

Shared Preferences The shared_preferences package allows the mobile application to store user-specific data persistently. This technology is crucial for retaining essential information, such as access tokens and user preferences, even when the app is closed. The application stores the following information in shared preferences: accessToken, refreshToken, expiration date of the token, first name, last name, and email. By utilizing shared preferences, the application enhances user experience by minimizing the need for repeated logins and providing quick access to personalized content.

Image Management

Camera The camera package enables users to capture images directly through the mobile device's camera. This functionality is critical for documenting psoriasis lesions in real-time, facilitating a direct link between the patient's condition and their visual progress over time. The integration of camera functionality enhances user engagement and ensures accurate tracking of lesion development.

Photo View The photo_view package allows users to view images with pinch-to-zoom and panning functionalities. This feature enhances the application's image gallery, providing users with an intuitive interface to explore their captured images in detail, which is particularly beneficial for tracking psoriasis lesions.

Permissions Management

Permission Handler The permission_handler package facilitates the management of app permissions on mobile devices. This technology is essential for requesting necessary permissions, such as access to photos, camera, and manageExternalStorage, enabling the application to perform essential functionalities related to image capture and storage.

User Interface

Body Part Selector The body_part_selector package simplifies the selection of body parts for users, enhancing the user experience by providing a visually appealing and user-friendly interface. This functionality aids users in accurately documenting their lesions based on the affected body parts.

2.2 Related works with Psoriasis Detection

In this chapter, several related works are presented, providing an analysis of architectures and methodologies used in problems similar to those addressed in this project. These studies offer valuable insights into the current advancements and limitations in the field, particularly in the detection and classification of dermatological conditions using image processing and machine learning techniques. By examining these works, a clearer understanding of how existing approaches can inform and enhance the development of the proposed solution is gained.

Computer-Aided Diagnosis Psoriasis Lesion Using Skin Color and Texture Features

This study (Jarad and Dawood, 2018) investigates a method for detecting psoriasis through the analysis of skin color and texture. The dataset comprises 200 images, with psoriasis serving as the primary condition for classification. During the pre-processing stage, a color histogram is employed to enhance feature extraction. The classification is executed using an Artificial Neural Network (ANN), which assesses whether the skin exhibits signs of psoriasis or remains healthy.

The study reports exceptional results, achieving 100 % accuracy, 100 % sensitivity, and 100 % specificity. The performance of the system was evaluated using two distinct methodologies: the RGB-Local Binary Pattern method and the RGB Color Co-occurrence Matrix method. Both approaches yielded identical and optimal performance, thereby demonstrating the model's robustness in effectively distinguishing between healthy skin and psoriasis lesions.

Despite these promising outcomes, the study is not without limitations. It lacks semantic segmentation, which is critical for accurate identification and comparison of lesion areas. Furthermore, the system does not incorporate a mechanism for tracking lesion history over time, which restricts its potential for long-term monitoring and patient care. Lastly, the absence of a user-friendly platform or product for daily use diminishes its practicality for patients or healthcare professionals seeking a comprehensive, real-world solution.

A Computer-Aided Diagnosis System for Classifying Prominent Skin Lesions Using Machine Learning

This paper (Hameed et al., 2018) investigates the application of machine learning models for the classification of various skin diseases, including psoriasis. The dataset comprises 1,800 images that represent different skin disease types. In the pre-processing phase, the images are resized to 720x720 pixels, hair is removed, and a Gaussian filter is applied to mitigate noise and enhance pertinent features.

The study utilizes multiple classification methods across 14 experiments to evaluate the performance of different machine learning models. These methods encompass decision trees (Fine, Medium, and Coarse), support vector machines (Linear, Quadratic, Cubic, and Gaussian), K-nearest neighbors (Weighted, Medium, and Cubic), ensemble classifiers (Bagged tree, Boosted, and Subspace Discriminant), and an artificial neural network (ANN). The output for all experiments involves the classification of the type of skin disease based on the input images.

The results indicate a range of performance across the classifiers. Among decision trees, the medium tree achieved the best performance, yielding an accuracy of 83.4 % for three-label classification and 76.2 % for six-label classification. In the context of support vector machines, the quadratic SVM demonstrated superior performance with 93.3 % accuracy for three-label classification and 83 % for six-label classification. The weighted K-nearest neighbor classifier produced accuracy rates of 88.7 % and 76.6 % for three-label and six-label classifications, respectively. Among the ensemble classifiers, the bagged tree attained the highest accuracy, achieving 90.3 % for three-label classification and 80.8 % for six-label classification. The artificial neural network, while effective in three-label classification with an accuracy of 87 %, underperformed in the six-label classification, attaining only 64.4 %.

This system offers several advantages. It facilitates the classification of a wide array of skin diseases, rendering it a versatile tool for dermatological diagnosis across multiple conditions. Moreover, the extensive dataset employed in this study enhances the model's performance, contributing to its relatively high accuracy in classifying skin lesions.

Nevertheless, the study presents several important limitations. It lacks semantic segmentation, which restricts the model's ability to accurately delineate and quantify the area of the lesions. Furthermore, the absence of a mechanism for tracking lesion history is a significant shortcoming, as such tracking is essential for the long-term monitoring of conditions like psoriasis. Lastly, the study does not provide a user-friendly platform or product for everyday use, thereby limiting its applicability in real-world, patient-centered healthcare environments.

A Fuzzy Expert System Design for Diagnosis of Skin Diseases

The paper (Raza et al., 2019a) presents a system that employs fuzzy logic to diagnose a range of skin diseases based on a comprehensive set of symptoms. Specifically, the system considers 15 input symptoms, including erythema, scaling, defined borders, itching, the Koebner phenomenon, papules, family history, age, perivascular lymphocytic infiltrate (PNL infiltrate), exocytosis, acanthosis, parakeratosis, spongiosis, follicular horn plugs, and infiltrate. The system is designed to process a dataset comprising 366 images that represent various skin diseases, including psoriasis.

The classification process is facilitated by a fuzzy logic controller, which analyzes the input symptoms and generates a diagnosis for one of the following skin conditions: psoriasis, seborrheic dermatitis, lichen planus, or pityriasis rosea. By incorporating fuzzy logic, the system effectively addresses the uncertainty and imprecision often associated with medical diagnoses based on symptomatic data.

In terms of results, the fuzzy expert system achieves an accuracy of 90.27 %, demonstrating its effectiveness in diagnosing skin diseases based on the provided symptoms. One notable advantage of this system is its capability to accommodate multiple types of skin diseases, thereby rendering it a versatile tool for dermatological diagnosis across various conditions.

Nevertheless, despite the promising accuracy, several limitations are evident when compared to more advanced approaches. Firstly, the system does not incorporate semantic segmentation, a critical technique for accurately delineating and quantifying lesion areas, which is particularly relevant for diseases such as psoriasis. Additionally, the absence of a mechanism for tracking lesion history over time limits the system's utility in monitoring disease progression. Finally, the lack of a user-friendly product or platform for everyday use restricts the accessibility and practicality of the system for continuous patient monitoring in real-world healthcare settings.

A Novel Approach for Automatic Identification of Psoriasis Affected Skin Area

The study (Raza et al., 2019b) presents a methodology for identifying skin affected by psoriasis through color histogram analysis. The system processes a dataset consisting of 80 images of psoriasis-affected skin and conducts classification based on RGB parameters, expressed in terms of mean and standard deviation. This approach does not utilize artificial intelligence (AI); instead, it relies on the analysis of the color properties of the skin.

The classification process employs RGB color space analysis in conjunction with Doyle's distance, a metric for assessing the severity of psoriasis. The findings indicate that a smaller Doyle's distance correlates with healthier skin, while a larger Doyle's distance is associated with more severe manifestations of psoriasis. This demonstrates the potential of color-based image analysis in differentiating between healthy and affected skin.

Despite the effectiveness of this approach in distinguishing between healthy and unhealthy skin, it presents several limitations when compared to more advanced techniques. Most notably, the system does not employ semantic segmentation, which would enable precise pixel-level classification of lesion areas—an essential component for monitoring the progression of psoriasis. Furthermore, the absence of a mechanism for historical tracking of lesions limits the system's utility in long-term monitoring and patient management. Finally, the lack of a user-friendly product or platform for daily use restricts its applicability in real-world settings, thereby diminishing accessibility for patients and clinicians seeking a practical solution for continuous monitoring.

Advanced Skin Diseases Diagnosis Leveraging Image Processing

This paper (Sinthura et al., 2020) explores an image processing-based approach to diagnose various skin diseases, including psoriasis, using a dataset of 100 images. The pre-processing phase involves resizing the images and applying an adaptive median filter to remove noise and other distortions, improving image quality for analysis. The images are then converted to grayscale, reducing the amount of pixel information, thereby simplifying the computational process.

For segmentation, the method employs edge detection techniques and feature extraction, aimed at identifying disease-relevant characteristics in the images. The system then classifies the images using a support vector machine (SVM) to determine whether the skin is healthy or affected by a skin disease. This approach achieves an accuracy of 89%, showcasing its potential in diagnosing skin diseases through image-based analysis. An advantage of this method is that it supports a wide range of skin disease types, offering versatility in clinical applications.

However, despite its promising results, the study has prominent limitations. The system does not incorporate semantic segmentation, which limits its ability to perform detailed, pixel-level lesion classification—a crucial aspect in providing more precise diagnosis and lesion area measurement. Additionally, it lacks the capacity for historical tracking of lesion progression, a critical feature for monitoring chronic skin conditions such as psoriasis over time. Lastly, the absence of a user-friendly product or platform for daily use restricts its practicality for real-world applications, making it less accessible for continuous monitoring by patients and healthcare providers.

An Automated Dermatological Images Segmentation Based on a New Hybrid Intelligent ACO-GA Algorithm and Diseases Identification Using TSVM Classifier

This study (Ahmed et al., 2019) presents an automated system for dermatological image segmentation and disease classification, employing a hybrid approach that integrates Ant Colony Optimization (ACO) and Genetic Algorithms (GA). The dataset comprises 812 images representing various skin diseases, including psoriasis. To enhance image quality and eliminate noise, the pre-processing stage incorporates several filtering techniques, including median filtering, Gaussian filtering, and contrast enhancement, alongside a region of interest (ROI) methodology to concentrate on pertinent areas of the skin.

Classification is executed using a hybrid Genetic Algorithm-Artificial Neural Network (GA-ANN) approach, a Feed Forward Back Propagation Artificial Neural Network (BP-ANN), and the core innovation—a hybrid ACO-GA algorithm combined with a Twin Support Vector Machine (TSVM). The system's output is the identification of specific skin diseases. While TSVM highlights the lesion pixels, the accuracy of the system is assessed based on disease type classification rather than lesion segmentation.

The system achieves an impressive classification accuracy of 95 %, demonstrating the effectiveness of its hybrid intelligent approach in diagnosing a broad spectrum of dermatological conditions. A significant advantage of this system is its utilization of a relatively large dataset, which enhances model performance and contributes to its high accuracy. Additionally, the system's versatility in managing multiple skin disease types further augments its applicability in clinical settings.

However, despite these strengths, the system has several limitations. The absence of semantic segmentation constrains its ability to conduct precise pixel-level lesion classification, which is essential for accurate measurement of lesion areas. Furthermore, it lacks the capability to track lesion history over time, a critical feature for monitoring chronic skin conditions such as psoriasis. Finally, the lack of a user-friendly platform or product restricts its practicality for continuous patient care, thereby diminishing its accessibility within real-world healthcare environments.

An Automated Detection of Scabies Skin Disease Using Image Processing and CNN

This paper (Halder et al., 2022) presents an automated system for detecting scabies, with a specific focus on dark-colored skin, utilizing a combination of image processing techniques and a convolutional neural network (CNN). The initial dataset consists of 141 images of scabies-infected skin, which is subsequently expanded to 1,817 images through various data augmentation techniques, including cropping, rotation, scaling, translation, and brightening. This augmentation enhances the diversity of the training data, thereby improving the system's robustness.

The segmentation stage employs thresholding, a fundamental image processing technique used to isolate scabies lesions from the surrounding healthy skin. For classification, a CNN is trained to differentiate between healthy skin and scabies-infected skin, achieving an impressive accuracy of 97.25 %.

While the system demonstrates a high degree of accuracy in detecting scabies, its scope remains limited. A significant advantage of this system is its capacity to accommodate multiple disease types; however, its primary focus on scabies detection constrains its broader applicability within dermatology. Moreover, the system does not facilitate psoriasis detection, further limiting its versatility. Additionally, the absence of semantic segmentation precludes pixel-level precision in delineating lesion areas. Furthermore, the system does not track lesion history over time, which is crucial for managing chronic conditions. Finally, the lack of a user-friendly product or platform for daily use restricts its practical application in real-world scenarios, particularly for patients requiring continuous monitoring.

An Intelligent Approach to Segmentation and Classification of Common Skin Diseases in Sri Lanka

This study (Wijesinghe et al., 2019) investigates an intelligent approach for the segmentation and classification of various skin diseases, including psoriasis, employing a comprehensive methodology tailored to the context of Sri Lanka. While the specific number of images utilized in the study is not disclosed, the pre-processing phase includes resizing the images to 500 x 500 pixels, setting non-skin regions to black, and applying Gaussian smoothing to enhance the quality of the input data.

The segmentation process employs a YUV-based color segmentation technique, which effectively isolates skin regions for analysis. Feature extraction is conducted using the gray-level spatial dependence matrix, capturing the spatial relationships of pixel intensities within the images. Subsequently, a genetic algorithm is utilized for feature selection, ensuring that the most relevant features are retained for classification.

For classification, the study implements a Linear Support Vector Classifier (LinearSVC), achieving an overall accuracy of 86.7 %. This level of performance demonstrates the efficacy of the proposed methodology in accurately identifying different types of skin diseases, highlighting the system's versatility across various conditions.

Despite these strengths, the study has prominant limitations. The absence of semantic segmentation restricts the model's ability to perform precise pixel-level lesion classification, which is essential for accurate diagnosis and treatment planning. Additionally, there is no provision for tracking lesion history, a critical factor for the ongoing management of chronic skin conditions such as psoriasis. Furthermore, the lack of a user-friendly product or platform for daily use limits the practical application of the system in real-world healthcare settings, thereby reducing its accessibility for both patients and practitioners.

Analysis and Diagnosis of Erythemato-Squamous Diseases Using CHAID Decision Trees

This study (Elsayad et al., 2018) presents a comprehensive analysis and diagnosis of erythemato-squamous diseases, including psoriasis, employing various classification techniques on a dataset comprising 358 images. The research utilizes the CHAID (Chi-squared Automatic Interaction Detection) decision tree model, along with bagging and boosting ensemble methods, and an artificial neural network (ANN) for the classification of skin disease types.

The results reveal remarkable performance metrics for each classification method. The CHAID decision tree model achieves an impressive F1 score of 95.2 % for psoriasis and lichen planus, 87.0 % for seborrheic dermatitis, 83.3 % for pityriasis rosea, 85.0 % for chronic dermatitis, and 90.9 % for pityriasis rubra pilaris. The bagging ensemble outperforms the CHAID model, with F1 scores of 96.8 % for psoriasis, 93.0 % for seborrheic dermatitis, and perfect scores of 100.0 % for chronic dermatitis and pityriasis rubra pilaris. Boosting yields comparable results, with an F1 score of 95.2 % for psoriasis and 100.0 % for pityriasis rubra pilaris, while the ANN demonstrates strong performance, notably achieving 96.8 % for psoriasis.

A key advantage of this study is its capacity to classify a wide variety of skin disease types, rendering it a valuable tool for dermatological diagnosis across multiple conditions. However, the study does present several limitations. The lack of semantic segmentation restricts the model's ability to accurately delineate and measure lesion areas, which is critical for effective disease management. Furthermore, there is no mechanism for tracking lesion history, which limits the system's utility for monitoring chronic conditions over time. Lastly, the absence of a user-friendly product or platform for daily use constrains the practical application of the findings in real-world healthcare settings, thereby reducing accessibility for both patients and healthcare providers.

Automated Detection of Skin and Nail Disorders Using Convolutional Neural Networks

This study (H et al., 2021) investigates the automated detection of various skin and nail disorders through the application of Convolutional Neural Networks (CNNs). The dataset utilized comprises 2,500 images representing diverse skin diseases, complemented by a second dataset generated through histogram equalization techniques. To enhance the robustness of the model, extensive data augmentation methods are employed, including rotation, flipping, contrast adjustment, brightness adjustment, sharpness adjustment, and Gaussian filtering. This augmentation process increases the total dataset size to 7,500 images for each dataset.

The classification task is performed using the MobileNetV2 architecture, which is well-suited for efficient image classification tasks. The results demonstrate strong performance, with the histogram equalization dataset achieving an accuracy of 92.4 % and the non-histogram equalization dataset attaining an accuracy of 90.1 %.

A significant advantage of this study is its capacity to classify a wide range of skin disease types, coupled with the substantial dataset size, which enhances the model's performance and generalization capabilities. However, the study also presents distinguished limitations. The absence of semantic segmentation restricts the model's ability to provide precise delineation and measurement of lesions, a critical aspect of effective disease management. Additionally, there is no provision for tracking lesion history, which is essential for monitoring chronic conditions over time. Lastly, the lack of a user-friendly product or platform for daily use limits the practical application of the findings in real-world healthcare settings, thereby reducing accessibility for both patients and healthcare providers.

Automatic Scale Severity Assessment Method in Psoriasis Skin Images Using Local Descriptors

This study (George et al., 2020) presents a method for assessing the severity of psoriasis lesions using local descriptors and superpixels. The dataset comprises 676 images specifically depicting psoriasis. Segmentation is performed utilizing superpixels, which simplify the representation of lesions for subsequent feature extraction.

Local feature descriptors are applied to the superpixels, and feature selection is conducted through K-means clustering, resulting in the formation of visual words. Classification is carried out using a multi-class Support Vector Machine (SVM), based on feature vectors derived from these visual words.

Experiment 1 indicated that smaller superpixels enhance the accuracy of scale scoring. Experiment 2 investigated the effects of codebook size and descriptors, revealing that color descriptors outperformed texture-based descriptors. Experiment 3 compared various clustering methods, with K-means demonstrating a slight advantage over Gaussian Mixture Models (GMM). The highest accuracy achieved was 80.81 %, obtained with a superpixel size of $15 \times 15 \times 3$, incorporating both color and texture descriptors, along with a codebook size of 128.

A significant advantage of this method is its utilization of a large dataset, which enhances the robustness and reliability of the results. However, several limitations persist. The absence of semantic segmentation restricts the model's capability to provide precise pixel-level delineation of lesions. Furthermore, the lack of tracking for lesion history hinders its applicability for long-term disease monitoring. Lastly, the study does not present a user-friendly platform or product for daily use, which limits its practicality in real-world healthcare environments.

Categorization of Integumentary System Disorders using Deep Learning

This study (ram asish Madiraju et al., 2022) focuses on the classification of various skin diseases, including psoriasis, using deep learning techniques. The dataset used for the study comprises 1200 images that span multiple types of integumentary disorders. To enhance image quality, the pre-processing step involves increasing contrast, improving the clarity of features within the images.

Data augmentation techniques were extensively applied to increase the diversity of the training data. These augmentations included rotating the images, shifting the width and height, applying shear transformations, magnifying, flipping, and adjusting brightness during every epoch. This variety of augmentations helps simulate different real-world conditions and improve the robustness of the model.

The classifier utilized in this study is a Convolutional Neural Network (CNN), a popular deep learning model known for its effectiveness in image classification tasks. The CNN model was trained to identify and categorize the skin diseases presented in the dataset. Upon evaluation, the model achieved a classification accuracy of 90.28 %.

One of the significant advantages of this approach is the large, diverse dataset used, which contributed to the high accuracy of the model. Additionally, the system was designed to handle multiple skin disease types, making it versatile across different medical conditions.

However, the study also presents several limitations. The model focuses solely on disease categorization and does not perform semantic segmentation, which is necessary for precise localization of the lesions. Additionally, there is no mechanism for tracking the history of lesions over time, which would be crucial for chronic conditions like psoriasis. Finally, the study does not offer a user-friendly platform or product for daily usage by patients, limiting its practical application for regular disease monitoring in real-world settings.

Comparison of Psoriasis Disease Detection and Classification Through Various Image Processing Techniques-A Review

This review paper (Vincent and Jayasingh, 2022) examines the detection and classification of psoriasis through various image processing techniques. The primary objective is to compare findings from existing literature and draw comprehensive conclusions regarding the effectiveness of different methodologies.

The input data comprises images specifically related to psoriasis. During the preprocessing phase, the images are converted to grayscale, thereby simplifying the data and enhancing subsequent analyses. For segmentation, both thresholding and the Otsu method are employed, facilitating the isolation of relevant features within the images.

Feature extraction is conducted using a novel model of Convolutional Neural Networks (CNN), which captures intricate patterns present in the images. The classification process involves the application of both CNN and Support Vector Machine (SVM) techniques, enabling a comparative analysis of their effectiveness in identifying psoriasis.

The results indicate a classification accuracy of 95.3 %, underscoring the potential of these image processing techniques for psoriasis detection. However, this review identifies several limitations inherent in the discussed methodologies. Notably, the absence of semantic segmentation restricts the ability to perform detailed, pixel-level analysis of lesions. Furthermore, there is no mechanism for tracking lesion history, which is essential for monitoring chronic skin conditions over time. Lastly, the lack of a user-friendly product or platform for daily use limits the practical applicability of these methods in real-world healthcare settings.

Contrast Enhancement for Color Dermascopy Images using Equalization based on Luminosity

The primary objective of this study (Sengupta et al., 2020) is to enhance the perceptual quality of color dermoscopy images through a contrast enhancement technique rooted in luminosity equalization. This approach is particularly significant as it serves as an experimental component within the proposed model for skin disease detection..

CURETO: Skin Diseases Detection Using Image Processing And CNN

This research (Karunanayake et al., 2020) introduces CURETO, an automated system for the detection of skin diseases, with a specific emphasis on acne. The dataset utilized comprises 4,000 images that encompass a variety of symptoms associated with acne. During the pre-processing stage, all images are resized to 224x224 pixels to standardize input dimensions for subsequent analyses.

For segmentation and feature extraction, the study employs Squeeze-and-Excitation Networks (Senet) transfer learning, thereby enhancing the model's capability to recognize intricate patterns within the images. The classification process is multifaceted, incorporating four components specifically designed to address various aspects of acne detection and classification:

1. Acne Density and Skin Sensitivity Identification: This component comprises several experiments utilizing diverse architectures, such as ResNeXt101 and ResNeXt50, to ascertain the accuracy of acne density detection.

2. Acne Subtype Identification using Image: This aspect employs SEResNet152 for the classification of different acne subtypes based on visual characteristics.

3. Acne Subtype Classification Model Using Symptoms: Here, classifiers such as Naïve Bayes, Support Vector Machine (SVM), and Convolutional Neural Networks (CNN) are utilized to predict acne subtypes based on symptomatic data.

4. Recommendations Based on Acne Density, Skin Sensitivity, and Acne Subtype: This final component evaluates multiple feature combinations to provide recommendations, achieving an accuracy of 87 % with a content-based algorithm.

The results indicate significant efficacy across various models, with the acne subtype classification model utilizing images achieving an accuracy of 79.55 % with SEResNet152, while the model based on symptoms attained an accuracy of 84 % with CNN.

The acne density model recorded an accuracy of 85.11 % with ResNeXt101, and the skin sensitivity model reached an impressive 95.76 % accuracy with ResNeXt50.

A noteworthy advantage of CURETO is its large dataset, which enhances the robustness of the results, alongside its capability to provide homeopathic remedy recommendations based on identified conditions, acne type, severity, and skin sensitivity.

Nevertheless, the study presents several limitations, including the absence of psoriasis detection, a lack of semantic segmentation for detailed lesion analysis, and the absence of a mechanism for tracking lesion history. Furthermore, the system does not offer a user-friendly platform for daily use, thereby limiting its practical applicability in real-world healthcare settings.

Deep Learning based Multiple Skin Disease Classification in Indian Territory

This study (Gairola et al., 2023) investigates the classification of various skin diseases, including psoriasis, utilizing deep learning techniques within the Indian context. The dataset comprises 302 images with an average resolution of 627x451 pixels, which are partitioned into three distinct datasets for training and evaluation purposes.

The primary outcome of this study is the identification of the skin disease type based on the input images. The results reveal varying performance levels among different models, with DenseNet121 achieving an average F1 score of 64.3 %, while DenseNet201 yields a slightly lower average F1 score of 62.3 %.

One important advantage of this research is its capacity to classify a wide range of skin disease types, rendering it a valuable tool for dermatological assessment. However, the study presents several limitations. The absence of semantic segmentation restricts the model's ability to conduct detailed pixel-level analyses, which are crucial for accurate lesion assessment. Additionally, there is no provision for tracking lesion history, an essential feature for the ongoing management of chronic conditions such as psoriasis. Furthermore, the lack of a user-friendly product or platform for daily use limits the practical applicability of the model in real-world healthcare scenarios.

Deep Learning-Based Dermatological Condition Detection: A Systematic Review With Recent Methods, Datasets, Challenges, and Future Directions

This study (Noronha et al., 2023) introduces DermaDetect, a computer vision model developed to enhance the accuracy of diagnosing various skin conditions, though it does not include psoriasis in its analysis. The dataset utilized comprises 587 images representing a range of skin diseases.

Classification is conducted using a ResNet architecture, which serves as the foundation for the model's diagnostic capabilities. The results from the experiments indicate significant performance variations, with Google AutoML achieving an average precision of 89 % in the first experiment, while a custom ResNet configuration yielded a higher average precision of 92

A salient advantage of this research is its capacity to classify a broad array of skin disease types, supported by a substantial dataset that contributes to improved model performance. However, the study also presents several limitations. Notably, the absence of psoriasis detection restricts its applicability to dermatological conditions involving this common skin disease. Furthermore, the lack of semantic segmentation prevents detailed analysis at the pixel level, which is essential for precise lesion assessment. The system does not offer lesion history tracking, a critical feature for managing chronic skin conditions, nor does it provide a user-friendly product or platform for daily use, thereby limiting its practicality in real-world healthcare settings.

Dermatitis Diagnosis – Modeling and Analysis Using Machine Learning

This study (Rajini et al., 2022) investigates the application of machine learning techniques for the diagnosis of various skin diseases, including psoriasis. Although the specific size of the dataset utilized is not disclosed, the research underscores the importance of effective pre-processing methods, particularly emphasizing noise removal, while the specific techniques employed remain unspecified.

For segmentation, the study employs several feature extraction methods, including intensity mapping, neighborhood operations, the application of distance metrics, and the Gray Level Co-occurrence Matrix (GLCM). These techniques contribute to the identification and characterization of skin lesions.

The classification of skin diseases is conducted using a Support Vector Machine (SVM) model, which achieves an impressive accuracy of 92 %. This result indicates the efficacy of the proposed machine learning framework in accurately identifying diverse skin conditions.

A significant advantage of this approach is its capacity to classify a wide range of skin disease types, thereby enhancing its utility in dermatological diagnostics. However, the study also presents considerable limitations. The absence of semantic segmentation restricts the model's ability to provide detailed pixel-level insights into lesion characteristics. Furthermore, there is no provision for tracking lesion history, which is crucial for the management of chronic conditions such as psoriasis. Lastly, the lack of a user-friendly product or platform for daily use limits the practical applicability of the findings in real-world healthcare settings.

Dermatological Disease Detection Employing Transfer Learning

This study (Mathur and Jain, 2023) investigates the detection of various skin diseases using transfer learning techniques, specifically excluding psoriasis from consideration. The dataset employed comprises 19,000 images, which provides a substantial foundation for model training and evaluation.

Pre-processing steps are undertaken to enhance image quality, incorporating techniques such as grayscale filtering, sharpening, median filtering, bilateral smoothing, and the application of a binary mask. These techniques aim to improve the visibility of relevant features within the images, thereby facilitating more accurate classification.

The classification process involves several prominent architectures, including Inception V3, Inception V2, MobileNet, and Xception. The results indicate varying performance levels across the models, with Inception V3 achieving an F1 score of 90.6 %, Inception V2 scoring 89.6 %, Xception attaining 91 %, and MobileNet exhibiting a significantly lower performance at 61.5 %. These findings suggest that the choice of model architecture plays a critical role in influencing classification accuracy.

A substantial advantage of this approach is its capacity to classify a wide variety of skin diseases, supported by the large dataset, which enhances the robustness of the results. However, the study is constrained by several limitations. The absence of psoriasis detection restricts its applicability to a broader range of dermatological conditions. Additionally, the lack of semantic segmentation inhibits detailed pixel-level analysis of lesions, and there is no mechanism for tracking lesion history. Lastly, the absence of a user-friendly product or platform for daily use diminishes the practical implications of the research in clinical settings.

Differential Diagnosis of Ringworm and Eczema Using Image Processing and Deep Learning

This study (Jardeleza et al., 2023) focuses on the detection of various types of eczema through the application of image processing techniques and machine learning methodologies. The methodology commences with pre-processing steps that involve resizing the images to 200x200 pixels and applying Gaussian blurring to reduce noise and enhance relevant features.

For segmentation, the study employs the YCbCr color model for skin region detection, while eczema regions are identified utilizing the CIELAB color model in conjunction with K-means clustering. This dual approach facilitates more accurate isolation of eczema from the surrounding healthy skin. Feature extraction is conducted using the Gray Level Co-occurrence Matrix (GLCM), which captures spatial relationships between pixel values, thereby aiding in the characterization of the skin conditions.

Classification is performed through a Support Vector Machine (SVM), which outputs classification results indicating either a type of eczema or healthy skin. The results demonstrate an accuracy of 83 %, highlighting the effectiveness of the applied techniques.

The advantages of this approach include the capacity to detect multiple types of eczema, supported by a robust dataset that enhances the reliability of the findings. However, the study is constrained by several limitations. Notably, it does not include psoriasis detection, which restricts its applicability to a narrower range of skin conditions. Additionally, the absence of semantic segmentation inhibits detailed lesion analysis, and there is no mechanism for tracking lesion history over time. Finally, the lack of a user-friendly platform for daily application diminishes its practical utility in real-world dermatological settings.

Differential Diagnosis of Ringworm and Eczema Using Image Processing and Deep Learning

This study (Nimesh and Weerasinghe, 2021) presents a differential diagnostic framework for distinguishing between ringworm and eczema through the application of image processing techniques and deep learning methodologies. The dataset comprises 3,364 images representing various skin types and lighting conditions, thereby ensuring a comprehensive evaluation of the model's performance across diverse scenarios.

To enhance the dataset's robustness, data augmentation techniques such as rescaling, zooming, rotating, flipping, shearing, and brightness adjustment are employed. These techniques effectively artificially expand the dataset and improve the model's generalization capabilities. Pre-processing steps include resizing images to standard dimensions, which prepares the data for input into the classification model.

The classification task is executed using a Convolutional Neural Network (CNN), which provides a diagnosis of the skin disease type. The results yield an F1 score of 82 %, indicating reliable performance in differentiating between the two conditions.

The advantages of this approach are noteworthy, as it accommodates multiple disease types and benefits from a substantial dataset that enhances the accuracy of the findings. Moreover, the model is designed to support various skin types and can process different image formats, with the exception of GIFs.

However, the study also presents several limitations. Notably, it does not include psoriasis detection, which narrows the focus to ringworm and eczema alone. Additionally, the absence of semantic segmentation restricts detailed analysis of skin lesions, and there is no functionality for tracking lesion history over time. Furthermore, the application is supported solely on Android devices, which may limit accessibility for users utilizing other platforms.

Importantly, this study aligns with the proposed model by providing a mobile application, thereby enhancing usability for end users.

Eczema, Hives and Psoriasis Detection with the Application of Local Binary Pattern, Color Histogram, SVM and RGB-HSV Color Space

This study (Cruz et al., 2019) investigates the detection of various skin diseases, including psoriasis, through the application of advanced image processing techniques. The dataset comprises 30 images, which is relatively limited and may potentially impact the robustness of the findings.

The pre-processing steps involve Gaussian blur, Gabor filtering, erosion, and dilation, which are employed to enhance image quality and prepare the data for subsequent analysis. Segmentation is performed using the Local Binary Pattern (LBP) method for texture classification, alongside a color histogram that analyzes pixel distribution across the images. This combined approach facilitates a comprehensive understanding of the texture and color characteristics associated with different skin conditions.

Classification is conducted using a Support Vector Machine (SVM), with the system outputting the identified skin disease type. The results demonstrate a commendable accuracy of 95 %, highlighting the effectiveness of the employed methods in distinguishing between eczema, hives, and psoriasis.

However, the study presents several limitations. The small size of the dataset restricts the generalizability of the results, and no data augmentation techniques are implemented to enhance the dataset's diversity. Additionally, the absence of semantic segmentation limits the ability to conduct detailed analyses at the pixel level, and there is no mechanism for tracking lesion history over time. Furthermore, the research does not provide a user-friendly platform or product for daily use, which may hinder practical applications in real-world settings.

The researchers suggest that improvements in lighting during image capture could further enhance image quality and diagnostic accuracy, thereby underscoring the importance of optimal imaging conditions in dermatological assessments.

Enhanced Deep Learning Approach for Accurate Eczema and Psoriasis Skin Detection

This study (Hammad et al., 2023) focuses on the detection of eczema and psoriasis through an enhanced deep learning approach, utilizing a dataset comprising 3,732 images. The pre-processing steps involved scaling and resizing the images to a uniform dimension of 180x180x3, along with the application of Gaussian blur to improve image quality.

To augment the dataset, various techniques were employed, including rotation, flipping, and cropping, resulting in a total of 6,286 images. This augmentation significantly enhances the diversity of the dataset, which is crucial for training robust models.

The classification experiments were conducted using several architectures: AlexNet, ResNet, VGG-16, and a custom convolutional neural network (CNN). The results of these experiments reveal varied performance levels: AlexNet achieved an accuracy of 60.8 %, ResNet attained 58.7 %, VGG-16 reached 82.2 %, and the custom CNN demonstrated the highest accuracy at 96 %. These findings highlight the potential of deep learning models, particularly custom architectures, in effectively distinguishing between eczema and psoriasis.

Despite the study's strengths, several limitations are noted. The absence of semantic segmentation restricts detailed pixel-level analysis, which is essential for precise lesion assessment. Additionally, there is no mechanism for tracking lesion history over time, a critical aspect of managing chronic skin conditions. Furthermore, the lack of a user-friendly platform or product for daily use limits the practical applicability of the research findings in clinical settings.

Overall, the enhanced deep learning approach presented in this study demonstrates promise for accurately detecting eczema and psoriasis, underscoring the significance of robust dataset augmentation and advanced model architectures in dermatological image analysis.

Conclusion

In this comprehensive review of current literature on psoriasis detection and classification using image processing and machine learning, it becomes evident that while significant progress has been made, there remain critical gaps that our proposed model seeks to address. The analyzed papers demonstrate a variety of methodologies ranging from traditional machine learning classifiers to advanced deep learning architectures. However, they frequently fall short in several key areas.

Most notably, the majority of the studies do not incorporate semantic segmentation, which is essential for accurately delineating lesion boundaries and quantifying lesion area—a core objective of our proposed model. Moreover, many existing systems lack a user-friendly interface or mobile application support, limiting their practical applicability for patients and clinicians alike. Several works have also highlighted the absence of comprehensive datasets that include psoriasis images, thereby hindering the generalization of their models across diverse populations and varying disease presentations.

In contrast, our proposed solution is designed to be a robust, accessible tool for psoriasis detection and monitoring, incorporating semantic segmentation for precise pixel classification, alongside a mobile application for ease of use. Furthermore, we aim to implement historical tracking of lesion characteristics and comparative analysis with prior images, enhancing the model's utility in a real-world clinical setting.

Additionally, our model will explore various pre-processing techniques identified in the literature, including adaptive filters, color space transformations, and segmentation methods. This experimentation will not only refine our approach but also allow us to evaluate the effectiveness of these methods in real-world scenarios, ensuring optimal performance and accuracy in detecting and classifying psoriasis lesions.

The integration of advanced segmentation techniques and a user-oriented design positions our model as a significant advancement in the field of dermatological diagnostics. By bridging the identified gaps, we aim to contribute meaningfully to the ongoing efforts in utilizing artificial intelligence for improved healthcare outcomes.

Looking forward, we plan to expand our research by testing our model across diverse demographics and varying clinical conditions, thereby enhancing its generalizability and effectiveness. Furthermore, we will investigate the potential of integrating additional machine learning techniques and exploring the impact of different input modalities on classification performance.

In summary, our work not only addresses the limitations present in current literature but also paves the way for future advancements in the detection and monitoring of psoriasis, ultimately aiming to improve patient outcomes and enhance clinical practices.

Chapter 3

Psoriasis Support System

3.1 Functional Requisites

The following functional requisites outline the core functionalities that the system must support, with a focus on capturing images, tracking lesion progress, and ensuring seamless interaction between the mobile app and the server:

URF.01	Provide a mobile application that allows users to capture images of psoriasis lesions and monitor their progression.
URF.02	Enable detection of psoriasis lesions through captured images using AI-based methods.
URF.03	Track and display lesion progression over time by analyzing and comparing historical image data.
URF.04	Automatically calculate and present lesion progression percentages based on server-side AI predictions.
URF.05	Synchronize lesion data and images between the mobile application and the server for consistent tracking across devices.
URF.06	Ensure secure user authentication and data access through JWT token mechanisms.
URF.07	Allow users to securely update their profile information and passwords.

3.2 General System Architecture

The architecture of the Psoriasis Support System is structured around a client-server model, emphasizing two principal components: the database and the AI model (see Figure 3.1). The database serves as the central repository for storing user and lesion-related data, while the AI model is crucial for analyzing images and making predictions regarding psoriasis lesions.

Client-Server Interaction:

- **Client Component (Mobile Application):** The mobile application acts as the user interface, allowing users to capture images of their lesions and track their progression over time. The client initializes its local database to manage user data and lesion images effectively. Upon launching the application, users can create an account or log in, initiating a session that enables them to interact with the server and access their lesion data.

- **Server Component:** The server operates as the backbone of the system, responsible for processing requests, analyzing images, and storing data in the database. It utilizes HTTP or HTTPS protocols to ensure secure communication with the client. Upon initialization, the server integrates the AI model and connects to the database, which is pre-populated with essential static data, such as predefined body parts.

Database: The database is a crucial element of the system, storing user profiles, lesion images, and associated metadata. It allows for efficient data retrieval and synchronization between the client and server, ensuring users have up-to-date information about their lesions.

AI Model: The AI model is integral to the system's functionality, performing the essential task of image analysis and lesion classification. When an image is captured, it is sent to the AI model for processing. The model applies various algorithms to detect lesions, providing insights that are stored in the database for future reference. This model not only enhances the accuracy of lesion detection but also contributes to tracking lesion progression over time.

This architecture facilitates a seamless interaction between components, enabling users to efficiently manage their health by capturing, analyzing, and tracking their psoriasis lesions. The client-server model supports a robust framework that allows for continuous improvements, such as updates to the AI model and database management, enhancing the overall performance and user experience of the Psoriasis Support System.

3.3 System Implementation

The project is structured around a client-server model, where the server handles requests, processes images, and stores data, while the mobile client serves as the interface through which users interact with the system. Central to this architecture is the Fully Convolutional Network (FCN) model, which performs image analysis and lesion classification. The FCN model was trained using a dataset of images with dimensions of 480x720, sourced from Dermnet NZ, specifically focused on psoriasis lesions. While the dataset primarily consists of images depicting Caucasian skin tones, efforts were made to ensure the model could generalize as much as possible. Pre-processing was applied using bilateral smoothing to reduce noise while preserving key lesion details. After pre-processing, the images were resized to 480x480 to match the training input requirements. To expand the dataset and improve the model's performance, data augmentation techniques, specifically translation, were used to generate additional training samples, improving the model's generalization.

Once trained, the AI model is integrated into the server. Upon initialization, the server loads the API routes, incorporates the trained AI model, and connects to the database. If the database is being used for the first time, it is populated with static data, such as predefined body parts, which remains unchanged throughout the application's lifecycle.

The server then listens for incoming connections from the client over secure HTTP or HTTPS protocols.

When the application is launched for the first time, the client initializes its local database and sets up the necessary tables for storing user and lesion-related data. The user is prompted to either create a new account or log in. To create an account, the user provides their first name, last name, email address, and password, which are sent to the server. The server securely encrypts the password using the SHA256 algorithm and stores the user's details in the database. Upon subsequent app launches, the client retrieves the JWT (JSON Web Token) stored in shared preferences to attempt automatic login by sending the token to the server. If the token has expired, the server responds with a 401 status, prompting the client to send the refresh token. The server then generates and returns a new access token, which the client updates in its local storage.

Once authenticated, the user can modify their account details, such as first name or last name, and update their password. These updates are sent to the server, which updates the corresponding entries in the database and returns an acknowledgment to the client. The client also saves these updated details in its shared preferences for future use.

The core functionality of the mobile application revolves around tracking psoriasis lesions. Users can capture images of lesions or review the progress of existing lesions by specifying the affected body part. To maintain synchronization with the server, the client first checks its local database for any lesion images associated with the selected body part. It then sends the list of picture IDs stored locally, along with the body part ID, to the server. The server retrieves all images associated with the body part from its database and compares them with the list of IDs provided by the client. If any images are missing on the client-side, the server sends the corresponding image bytes and other relevant metadata, including lesion contours and progress data. The client then stores the received images in its local storage and updates its database with the new information, including file paths, contours, and progress metrics such as lesion size development over time.

Within the body part lesion and picture gallery, users can view all captured images for each lesion. By selecting an image, they can zoom in for a closer examination and see the progress of the lesion, expressed as a percentage, by comparing each image with the previous and the first image captured of that lesion for long-term tracking.

When capturing a new lesion image, users can choose to associate the picture with an existing lesion or create a new one. If linked to an existing lesion, the app retrieves the contour coordinates from the most recent image in the database and overlays them on the camera viewfinder. This assists the user in aligning the camera with the lesion as closely as possible to the previous image, ensuring consistent measurements. If the lesion is new, no contours are displayed since none exist yet. After capturing the image, it is saved locally, and the user is given the option to either confirm or retake the picture. If confirmed, the app sends the body part ID, lesion ID, image path, and image bytes to the server.

On the server, if the lesion is new, a corresponding entry is created in the database. The image is then processed by the AI model in the background. During this process, the image undergoes pre-processing, including bilateral smoothing to reduce noise and resizing to a standard 480x480 resolution to match the training configuration of the model. The AI model then predicts the lesion mask and calculates the size of the lesion in pixels by counting the white pixels in the mask. The masked image is resized back to its original dimensions and saved on the server. Contours are extracted from both the original and the processed images, and the contours are applied to the original image for visual validation and testing purposes. The final lesion size and contour coordinates are updated in the picture's database record, concluding the image classification process.

This implementation ensures a seamless user experience while leveraging the power of AI to track lesion progression over time. The system's design enables efficient synchronization of data between the client and the server, providing users with up-to-date information on their lesions and supporting informed decisions regarding their health management.

3.3.1 Message Sequence Diagrams of Usage Scenarios

The following sequence diagrams illustrate the key interactions between the client, server, and database during various user scenarios. These diagrams help to clarify the message flow and how the system handles requests for each functionality.

Train AI Model

In this use case, the sequence (see Figure 3.2) of operations outlines the systematic process involved in preparing and training a machine learning model for lesion segmentation, with the objective of producing an optimized AI model. This process involves dataset handling, image pre-processing, data augmentation, model training, prediction generation, and performance evaluation. Each of these steps contributes to ensuring the model is both robust and accurate for the target task.

Initially, the benchmark system downloads a dataset containing images and their respective masks from Roboflow. This dataset is delivered as a zipped file, which is extracted and split into three sets: training, validation, and test datasets. These splits allow for an effective distribution of data to train the model, validate its performance during training, and finally test its generalization capability on unseen data.

Subsequently, a data augmentation process is applied to the training images and their corresponding masks using the OpenCV library. This augmentation, which involves transformations such as translation, helps generate additional data from the existing dataset, thereby enhancing the diversity of the training set. Each augmented image is then stored in the server's local storage. This is done iteratively for every image, ensuring that all data is properly augmented and stored for subsequent stages.

Following augmentation, pre-processing techniques are applied to all images, both original and augmented, to enhance image quality and optimize them for model training. These techniques, performed using OpenCV, are designed to remove noise while preserving essential features such as lesion boundaries. The pre-processed images replace the original images in the local storage, ensuring that the system uses enhanced images for training the AI model.

The training phase begins with the initialization of the AI model using TensorFlow Keras. This model is structured based on a Fully Convolutional Network (FCN), which is particularly effective for pixel-wise classification tasks such as lesion segmentation. The benchmark system then trains the model on the pre-processed images, leveraging the training dataset to learn how to accurately detect and classify lesion regions.

Once training is complete, the model is stored in the server's local storage for future use in predictions. This storage mechanism ensures that the trained model is readily accessible for the subsequent stages of evaluation and testing.

Following model training, the system uses the trained model to generate predictions on the test dataset. These predictions result in masked images, which represent the areas of the images classified as lesions by the model. For each predicted image, contours are extracted using OpenCV to delineate the exact boundaries of the lesion. These contours are then overlaid onto the original images for better visual interpretation. Both the masked and contoured images are stored in the server's local storage for further analysis and verification.

To evaluate the trained AI model, the system calculates performance metrics, including precision, recall, and the F1 score, using the scikit-learn (SKLearn) metrics library. These metrics provide a quantitative measure of the model's accuracy and ability to generalize well to unseen data. The calculated metrics are stored in the local storage for reference and further analysis.

Finally, the benchmark system generates a report, which includes an analysis of the model's performance. The system sorts all results by the F1 score, providing a ranked overview of how well the model performed on the test dataset. A CSV file is created containing these results, which is stored in the server's local storage for future reference and comparison with other models or configurations. This comprehensive evaluation enables continuous improvements to the model by providing insight into its strengths and areas for further optimization.

This process of dataset preparation, model training, prediction, and evaluation ensures a thorough and methodical approach to training an AI model for lesion segmentation, ultimately leading to a robust and accurate system capable of assisting in psoriasis lesion detection and tracking.

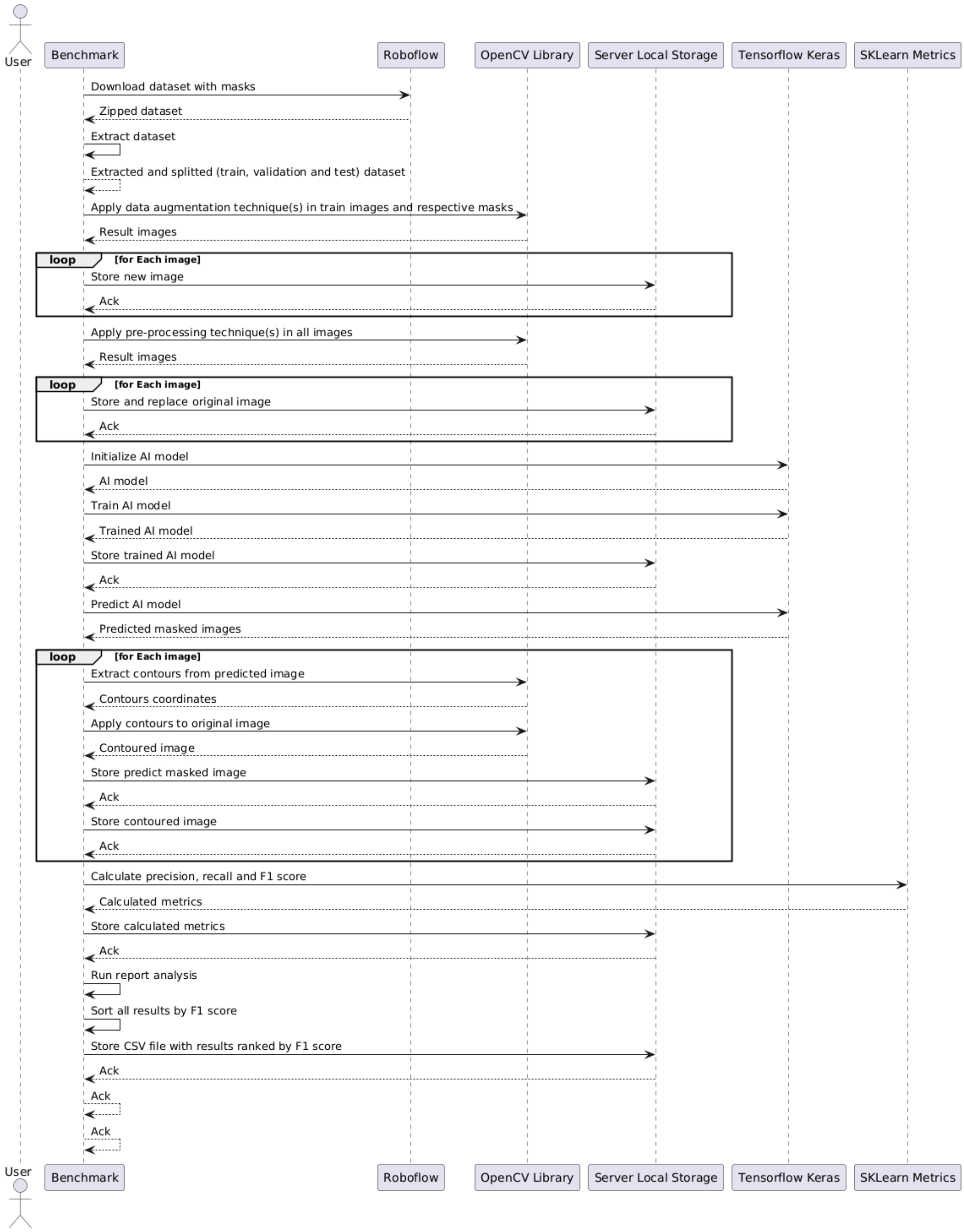

Figure 3.1: Sequence Diagram for Training an AI Model

Registration

In this registration use case, the sequence diagram (see Figure 3.3) illustrates the interaction between the user, the mobile application, the server, and the server's database as part of the user registration process.

The process begins with the User entering their personal data and credentials (such as username, password, email, etc.) into the Mobile App. This action initiates the user registration process, where the application acts as the interface to collect and submit the necessary information for the user to create an account.

Once the user's data is entered, the Mobile App sends a registration request to the Server. This request contains all the inputted personal data and credentials required to register the user.

Upon receiving the registration request, the Server communicates with the Server Database to create a new user model. This model represents the user account within the system, containing all relevant details such as personal information and credentials.

After the user model is successfully created, the Server Database sends an acknowledgment back to the Server, confirming that the user has been registered.

The Server then returns a 200 OK response to the Mobile App, indicating that the registration process was successful. The Mobile App informs the User that the registration is complete by transitioning them to the login screen, where they can proceed to log in with their newly created account.

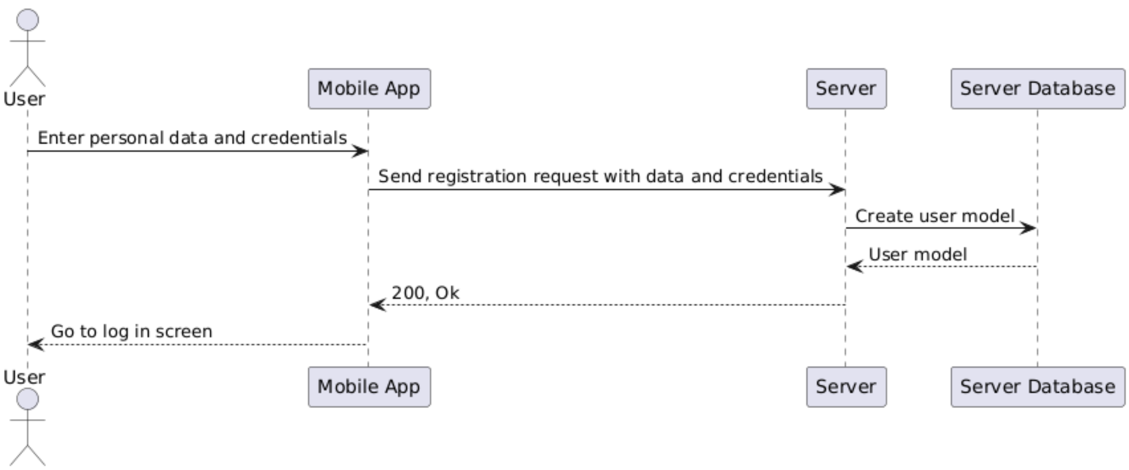

Figure 3.2: Sequence Diagram for Registering an Account

Log In

In this login use case, the sequence diagram (see Figure 3.4) outlines the flow of interactions when a user logs into the system using the mobile application. The process involves the user, the mobile app, the server, the server database, and local shared preferences where authentication tokens are stored.

The login process starts with the User entering their credentials (typically a username or email and a password) into the Mobile App. The App then sends a login request to the Server containing the user's credentials.

Upon receiving the login request, the Server queries the Server Database to retrieve the user model associated with the entered credentials. The Server Database responds with the relevant user model, which includes the user's stored credentials.

Next, the Server validates the password by comparing the provided password with the one stored in the Server Database. If the password is correct, the server proceeds to the next step.

Once the credentials are validated, the Server generates JWT tokens (access and refresh tokens) to manage the user's session securely. These tokens are used for authentication in subsequent requests.

The Server returns a 200 OK response along with the JWT tokens and the user's profile data to the Mobile App.

Upon receiving this information, the Mobile App stores the JWT tokens and the user's profile in the Local Shared Preferences, ensuring that the user remains logged in across sessions. Once stored, an acknowledgment is sent back to the App.

Finally, the Mobile App directs the User to the home screen, completing the login process and granting access to the app's main functionality.

This sequence guarantees that the login process is secure, with JWT tokens used for session management and user data stored locally for seamless access.

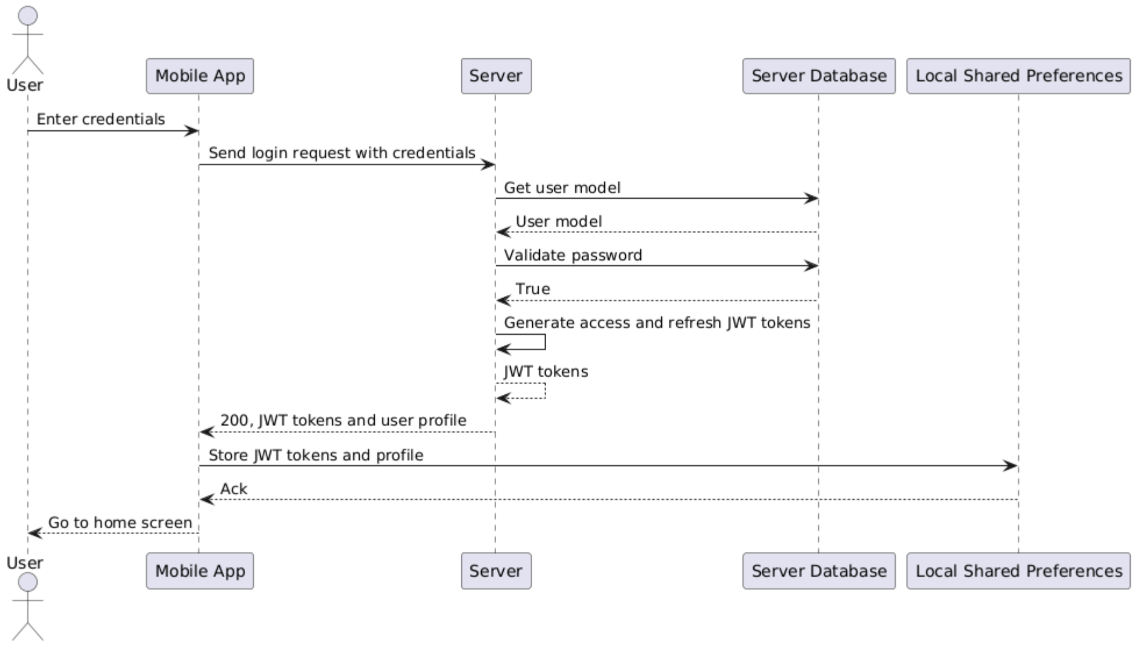

Figure 3.3: Sequence Diagram for Logging In

Log In With JWT Token

In the login with JWT token use case, the sequence diagram (see Figure 3.5) demonstrates how a user who has previously logged in can be authenticated using a stored JWT token, enabling them to bypass re-entering credentials.

When the User opens the Mobile App, the process begins by the App retrieving the previously stored access JWT token from the Local Shared Preferences. This token was saved during the initial login process.

After retrieving the token, the App sends a login request to the Server that contains only the access JWT token.

Upon receiving the request, the Server validates the JWT token to ensure its authenticity and expiration status. If the token is valid, the server extracts the email or user identifier from the token's payload.

The Server then queries the Server Database to fetch the associated user model, which contains the user's profile and other relevant data.

Once the Server retrieves the user model, it sends a response back to the App with the user's profile data, confirming successful authentication.

Finally, the Mobile App directs the User to the home screen, granting them access to the app without requiring a re-login. This process ensures a seamless user experience while maintaining security through token validation.

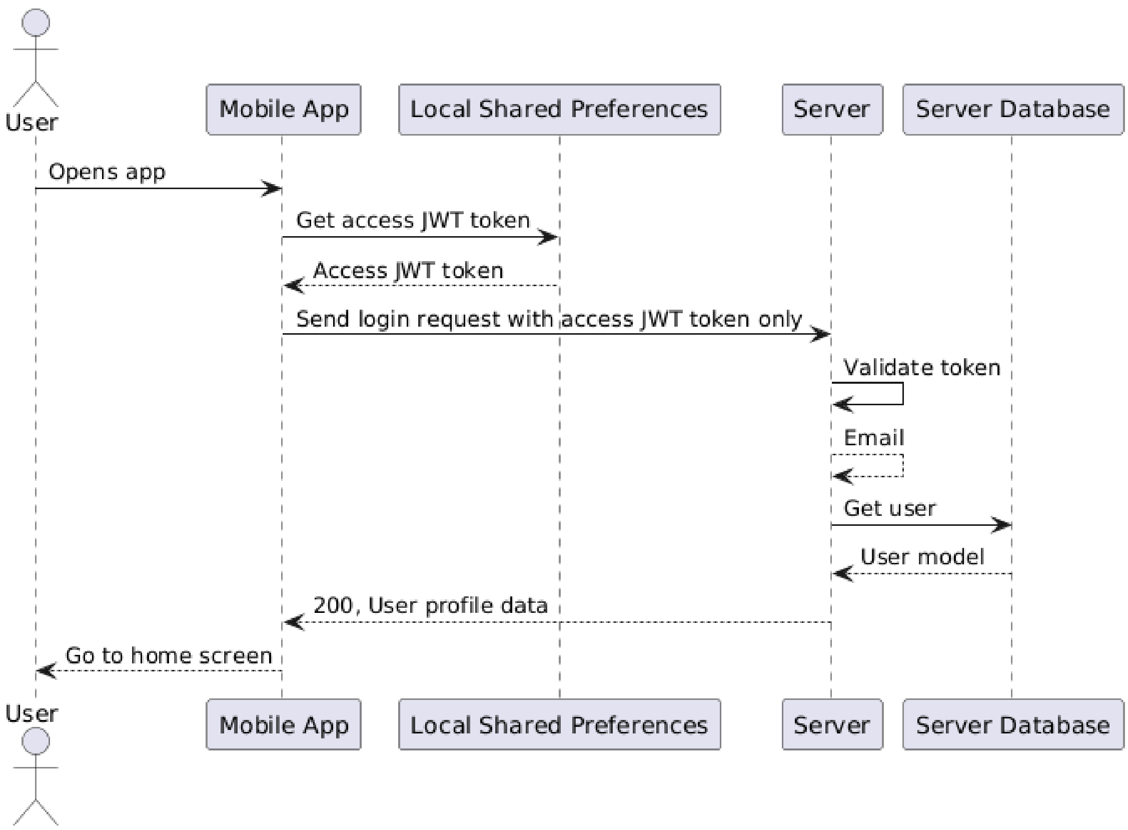

Figure 3.4: Sequence Diagram for Logging In with JWT Token

Refresh Access JWT Token

In the refresh access JWT token sequence diagram (see Figure 3.6), the process shows how the Mobile App handles the case when the access JWT token has expired or is no longer valid, ensuring secure communication with the server without requiring the user to log in again.

When the User initiates an action in the App that requires secure server communication, such as retrieving user data, the App first retrieves the stored access JWT token from the Local Shared Preferences.

The App sends the access JWT token to the Server. The Server attempts to validate the token. If the token is no longer valid (e.g., it has expired), the Server returns a 401 Unauthorized response, indicating that the access JWT token cannot be used.

Upon receiving the 401 Unauthorized response, the App retrieves the refresh JWT token from the Local Shared Preferences. It then sends a refresh token request to the Server, including both the refresh token and the expired access token.

The Server validates the refresh token and generates a new access JWT token. This new token is sent back to the App in the response.

The App stores the newly issued access JWT token in the Local Shared Preferences, ensuring it is used for future requests.

The App now retries its original request to the Server with the new access JWT token, and the User's action proceeds normally.

This process ensures the user remains authenticated without requiring a re-login, maintaining the session securely through the refresh token mechanism.

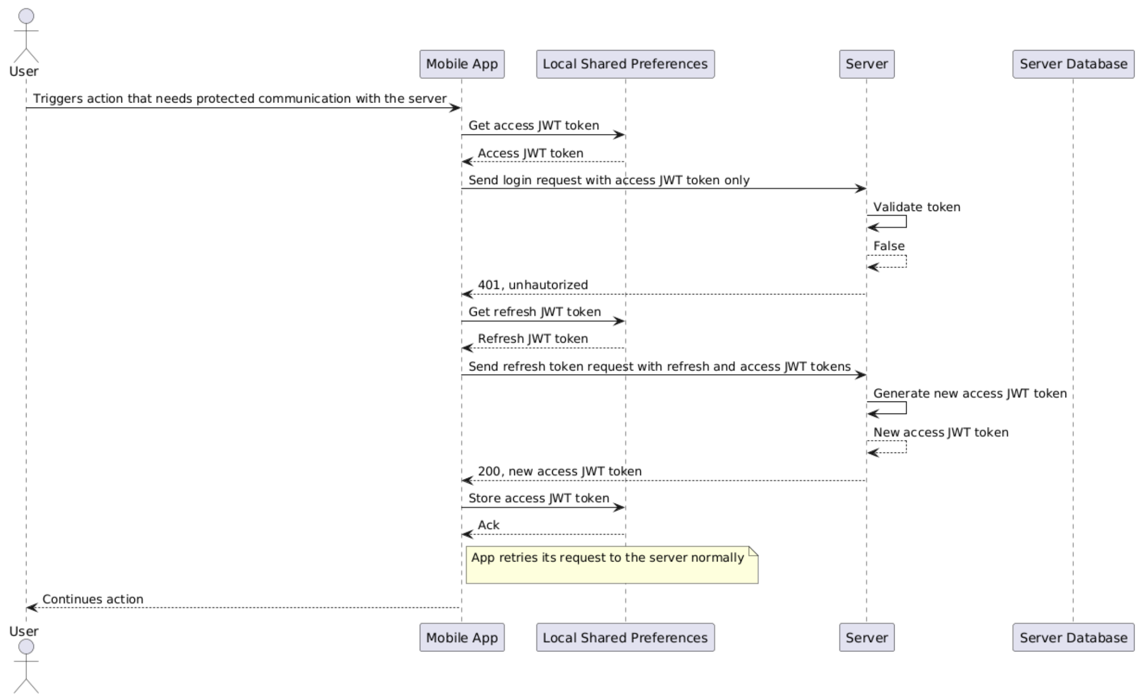

Figure 3.5: Sequence Diagram for Refreshing JWT Token

Log Out

The logout sequence diagram (see Figure 3.7) demonstrates the process through which the user logs out of the mobile app, ensuring that both the access JWT token and the refresh JWT token are invalidated and removed from the system.

The User initiates the logout process by selecting the log-out option in the Mobile App. The App retrieves the current access JWT token from the Local Shared Preferences to begin the log-out request.

The App sends the access JWT token to the Server for validation. However, since the User is logging out, the token may already be invalid (expired, etc.), leading the Server to return a 401 Unauthorized response if the token is invalid.

Upon receiving the 401 Unauthorized response, the App retrieves the refresh JWT token from the Local Shared Preferences. The App then sends a refresh token request to the Server, including both the access and refresh tokens, to continue the logout process.

The Server proceeds by storing both the access token and the refresh token in a blacklist table in the Server Database to ensure they are no longer valid for future requests.

Once the server-side blacklisting is complete, the App clears the Local Shared Preferences, removing all stored tokens and user data.

Finally, the User is redirected to the login screen, completing the logout process.

This approach guarantees that the tokens are invalidated both server-side and client-side, enhancing the security and ensuring the user is fully logged out of the system.

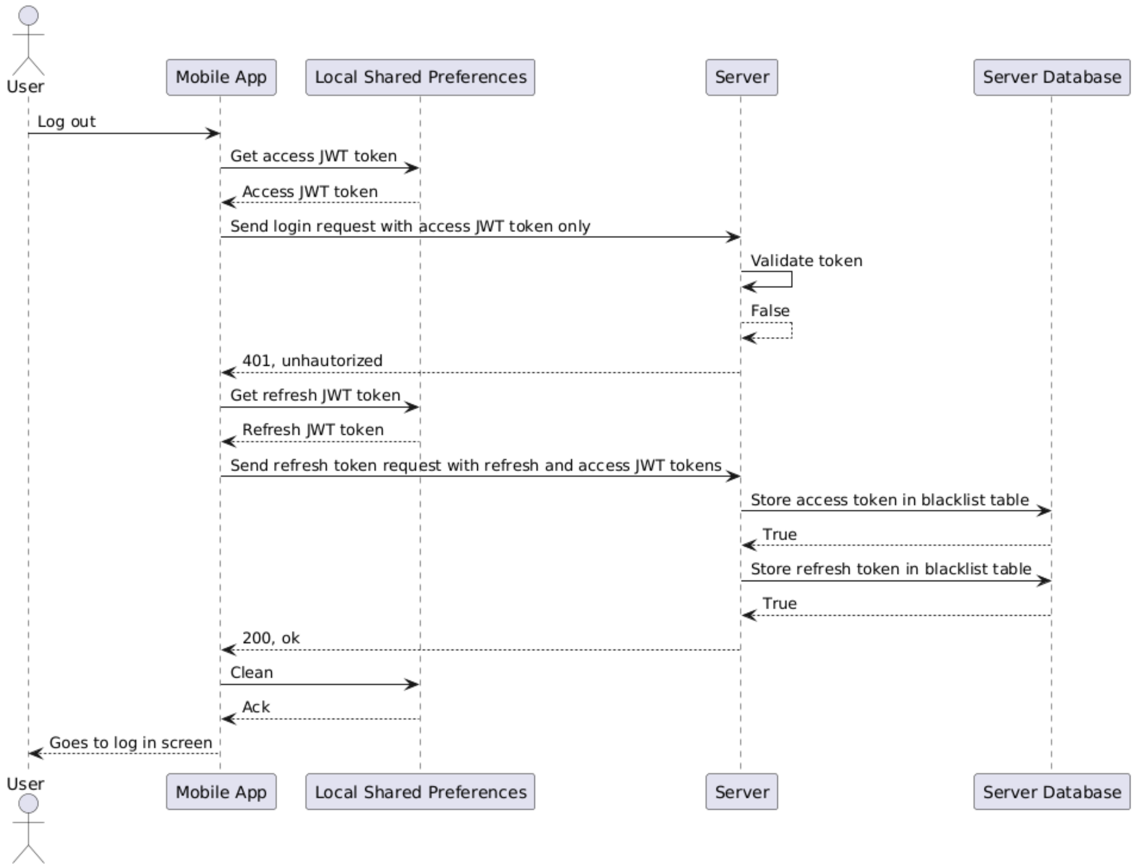

Figure 3.6: Sequence Diagram for Logging Out

Update Profile

The update profile sequence diagram (see Figure 3.8) illustrates the process through which the user can update their account details in the mobile app, ensuring that the changes are validated and reflected in the system.

The User initiates the profile update process by entering the new account details in the Mobile App. The App retrieves the current access JWT token from Local Shared Preferences to authenticate the update request.

The App sends a request to the Server containing the updated account details and the access JWT token.

The Server validates the access JWT token. If the token is valid, the server proceeds to the next step; otherwise, it returns an error response.

Upon successful token validation, the Server retrieves the user profile from the Server Database. The server then updates the user's account details with the new information provided by the User.

Once the user profile is successfully updated, the Server responds with a success message (HTTP 200 OK). The App notifies the User that the update was successful and navigates back to the home screen.

This process ensures that the user's profile is accurately updated while maintaining secure access through token validation.

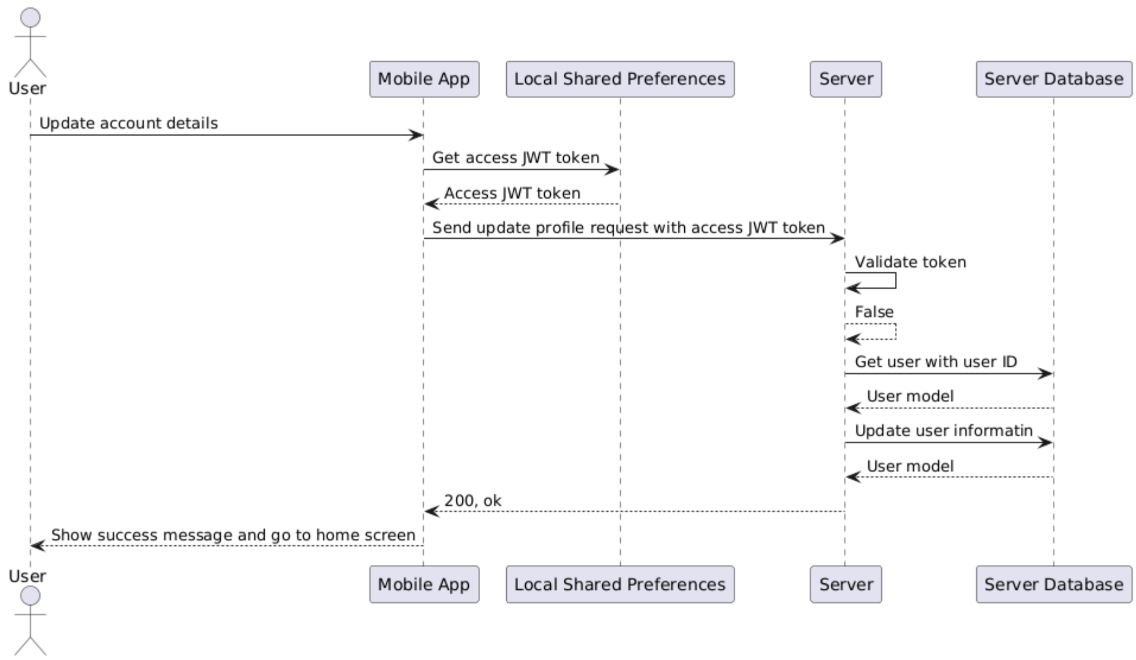

Figure 3.7: Sequence Diagram for Updating Profile

Update Password

The update password sequence diagram (see Figure 3.9) details the process by which the user can change their password within the mobile app, ensuring that the new password is securely updated in the system.

The User initiates the password change process by entering the old and new passwords in the Mobile App. The App retrieves the access JWT token from Local Shared Preferences to validate the request.

The App sends a request to the Server that includes the old password, new password, and the access JWT token.

The Server validates the access JWT token. If the token is valid, the process continues; if not, the server returns an error response.

After successful token validation, the Server retrieves the user profile from the Server

Database. The server generates a new hash for the new password and updates the stored password in the database.

Upon successfully changing the password, the Server sends a success response (HTTP 200 OK) to the App, which informs the User that the password was updated and navigates back to the home screen.

This flow ensures that the user's password is securely changed while maintaining authentication integrity through token validation.

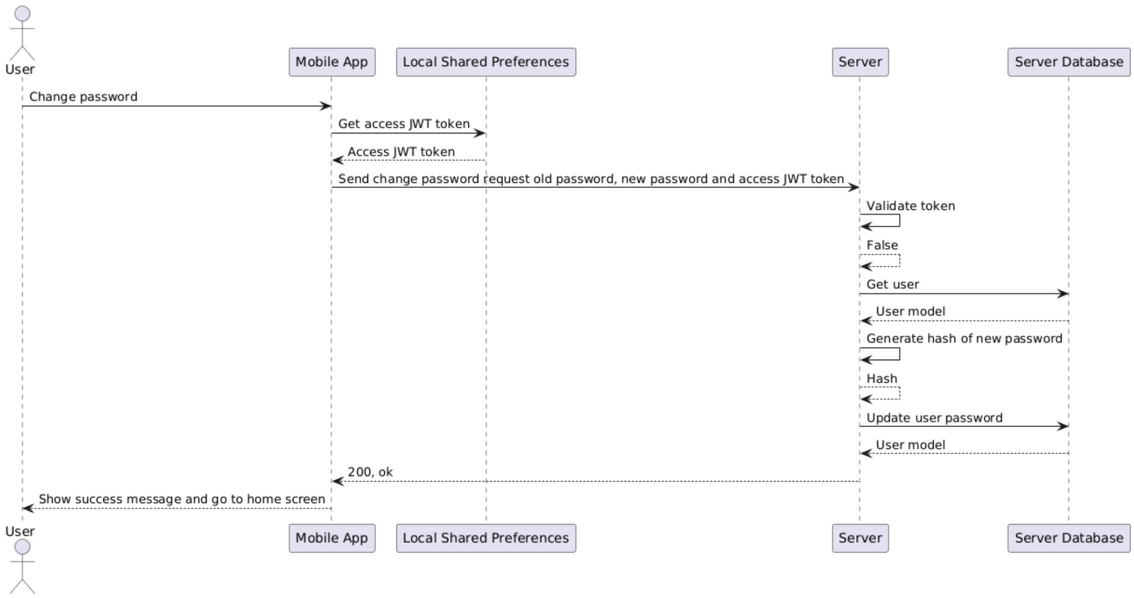

Figure 3.8: Sequence Diagram for Updating Password

Sync Lesions Pictures Client-Server and Show Pictures Gallery

The sync lesions pictures client-server and show pictures gallery sequence diagram (see Figure 3.10) illustrates the process through which the user synchronizes lesion pictures between the mobile app and the server while displaying the relevant images in a gallery format.

The User initiates the synchronization process by navigating to the gallery section within the Mobile App. The App responds by displaying a human body image to the User.

The User selects a specific body part to view associated lesions. The App retrieves lesion models related to the chosen body part from the Client Database.

After obtaining the lesion models, the App requests the corresponding pictures for those lesions from the Client Database. The database returns the relevant picture models.

For each picture, the App checks the Client Local Storage to determine whether the picture already exists. If the picture exists, the App stores its ID in an array for later reference.

The App retrieves the access JWT token from Local Shared Preferences to authenticate the synchronization request.

The App sends a synchronization request to the Server, including the list of existing picture IDs, lesion IDs, and the access JWT token.

Upon receiving the request, the Server validates the access JWT token. If the token is valid, the server proceeds to retrieve the user's lesion picture models from the Server Database.

For each picture model, the Server checks if the client already contains the picture. If the picture ID is not found in the list of existing pictures, the Server retrieves the picture bytes from the Server Local Storage.

Once all missing pictures are gathered, the Server responds to the App with a success message (HTTP 200) and the information of all body part lesion pictures.

The App stores the received lesion information and missing pictures in the Client Database. It then saves the missing pictures in the Client Local Storage.

Finally, the App displays the lesions along with their respective pictures and progress data to the User, completing the synchronization process.

This comprehensive workflow ensures that the user has access to the most up-to-date lesion pictures while maintaining data integrity through proper authentication and synchronization mechanisms.

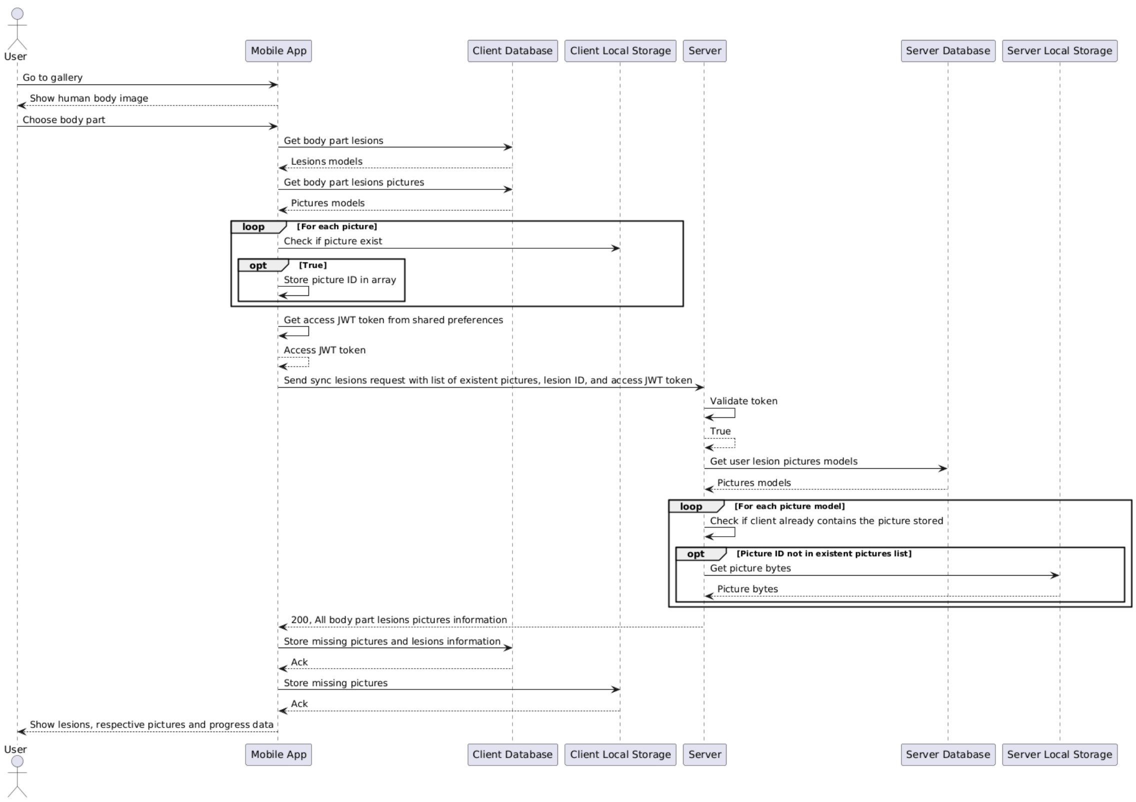

Figure 3.9: Sequence Diagram for Syncing Lesions and Pictures with Server and Show Pictures Gallery

Take Picture

The take picture sequence diagram (see Figure 3.11) illustrates the process through which the user captures an image of a lesion, uploads it to the server, and updates the relevant information in the gallery.

The User initiates the process by navigating to the gallery section within the Mobile App. The App responds by displaying a human body image for the User to select a body part.

After the User selects a specific body part, the App retrieves the lesion models associated with that body part from the Client Database. It then requests the corresponding pictures for those lesions.

The App presents the User with the lesions, their respective pictures, and progress data. This process includes synchronization of lesions between the client and server, which is detailed in another diagram.

The User can then click a button to capture an image for either an existing or a new lesion.

- **For an Existing Lesion:** The App creates a new lesion model in the Client Database and shows the camera preview image to the User.
- **For an Existing Lesion:** The App retrieves the lesion ID and associated picture models. It then gets the contours coordinates of the most recent picture to overlay on the camera preview image. The camera preview is displayed with the contours drawn on the screen.

When the User clicks the button to take a picture, the App instructs the Camera Controller to capture the image. The captured file is stored in the Client Local Storage, and the App displays the taken picture to the User.

- **For an Existing Lesion:** The App draws the contours on the screen after the image is captured.

The User submits the captured picture. The App retrieves the access JWT token from Local Shared Preferences and sends a request to the Server. This request includes the body part name, lesion ID, picture bytes, and the access JWT token.

The Server validates the token. Upon successful validation, it retrieves the body part ID from the Server Database and creates a new picture model.

- **If the Lesion Exists:** The Server retrieves the existing lesion model.
- **If the Lesion Does Not Exist:** A new lesion model is created in the database.

The Server adds the picture ID to the lesion's picture ID list, stores the picture in the Server Local Storage, and acknowledges the operation.

The Server initiates background processing to classify the uploaded picture using an AI model, which is described in another diagram.

Upon successful processing, the Server sends a response back to the App with a status of 200 (OK). The App then displays a success message to the User and navigates back to the home screen.

This workflow ensures a seamless experience for the user in capturing, uploading, and classifying lesion pictures, maintaining data integrity and providing immediate feedback throughout the process.

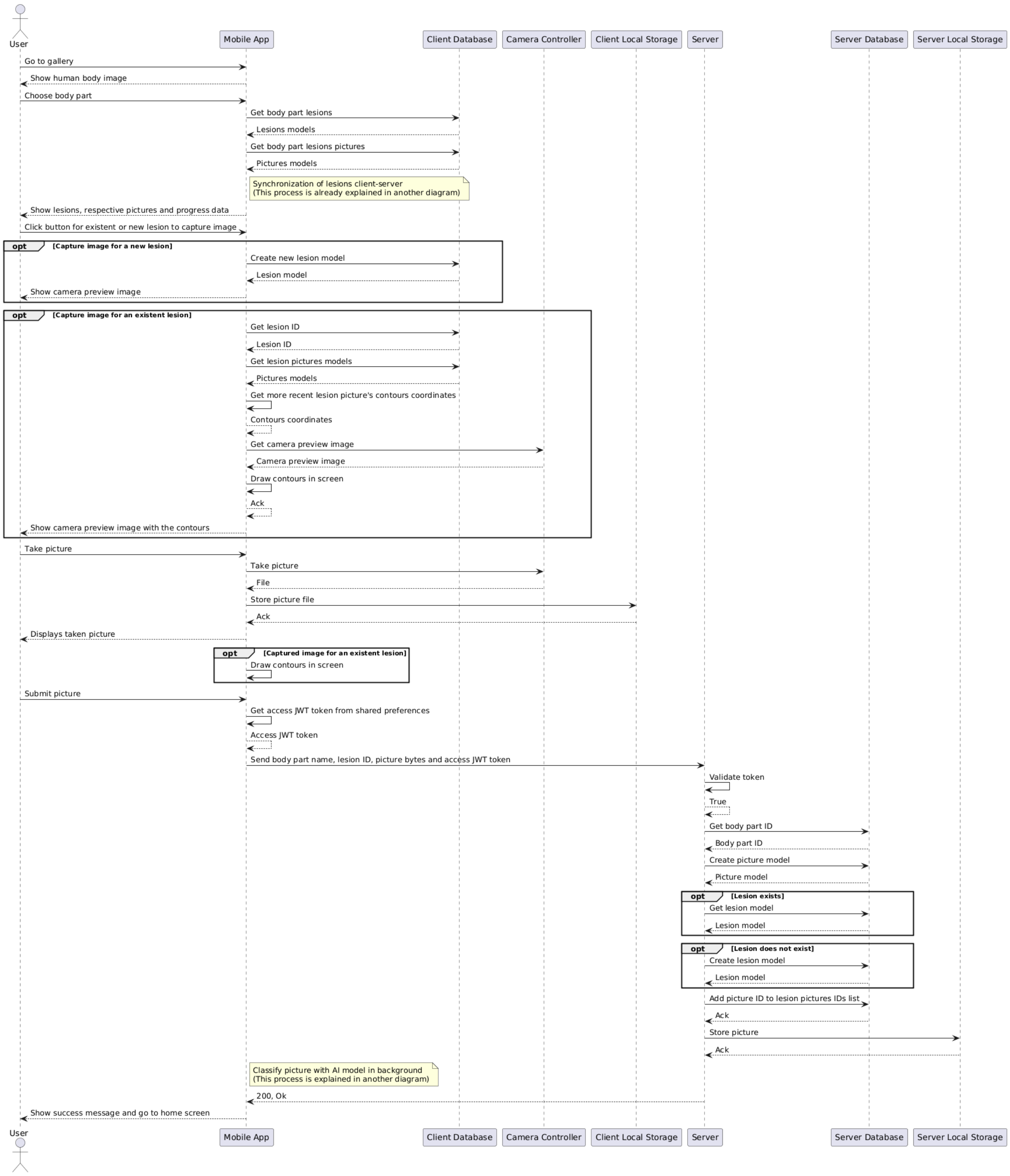

Figure 3.10: Sequence Diagram for Taking a Picture

Classify Image with AI Model

The classify image with AI Model sequence diagram (see Figure 3.12) outlines the process through which the server classifies an uploaded picture using an AI model, applying necessary pre-processing techniques and generating output for lesion detection.

The process begins when the Server receives the picture model ID and picture path from the previous steps, as outlined in another diagram.

The Server initiates image processing by applying a bilateral smoothing technique using the OpenCV Library. This step reduces noise in the image, enhancing the quality for further analysis. Once the pre-processing is complete, the OpenCV Library returns the pre-processed picture to the Server.

The Server then requests the OpenCV Library to resize the pre-processed image to the appropriate dimensions for the AI model. The resized picture is sent back to the Server.

Next, the Server sends the resized image to the AI Model for prediction. The AI Model processes the image and returns a masked predicted picture, indicating the areas identified as lesions.

The Server counts the number of white pixels in the masked predicted picture, which corresponds to the predicted lesions, and stores the result.

The Server requests the OpenCV Library to resize the masked predicted image back to its original dimensions. Once resized, the masked image is stored in the Server Local Storage, with an acknowledgment returned to the Server.

The Server then retrieves the contours of the original image from the OpenCV Library. After obtaining the contours, the Server applies these contours to the original image, creating a contoured image, which is also stored in the Server Local Storage.

Similarly, the Server obtains the contours of the masked predicted image. After acquiring these contours, it applies them to the original image, generating another contoured image for analysis.

Finally, the Server retrieves the corresponding picture model from the Server Database. It updates the model by storing the lesion size and contours coordinates associated with the detected lesions. An acknowledgment from the Server Database confirms that the updates were successfully made.

This process ensures that the uploaded images are thoroughly analyzed for lesion detection, providing valuable information regarding lesion size and contour details for further evaluation and tracking.

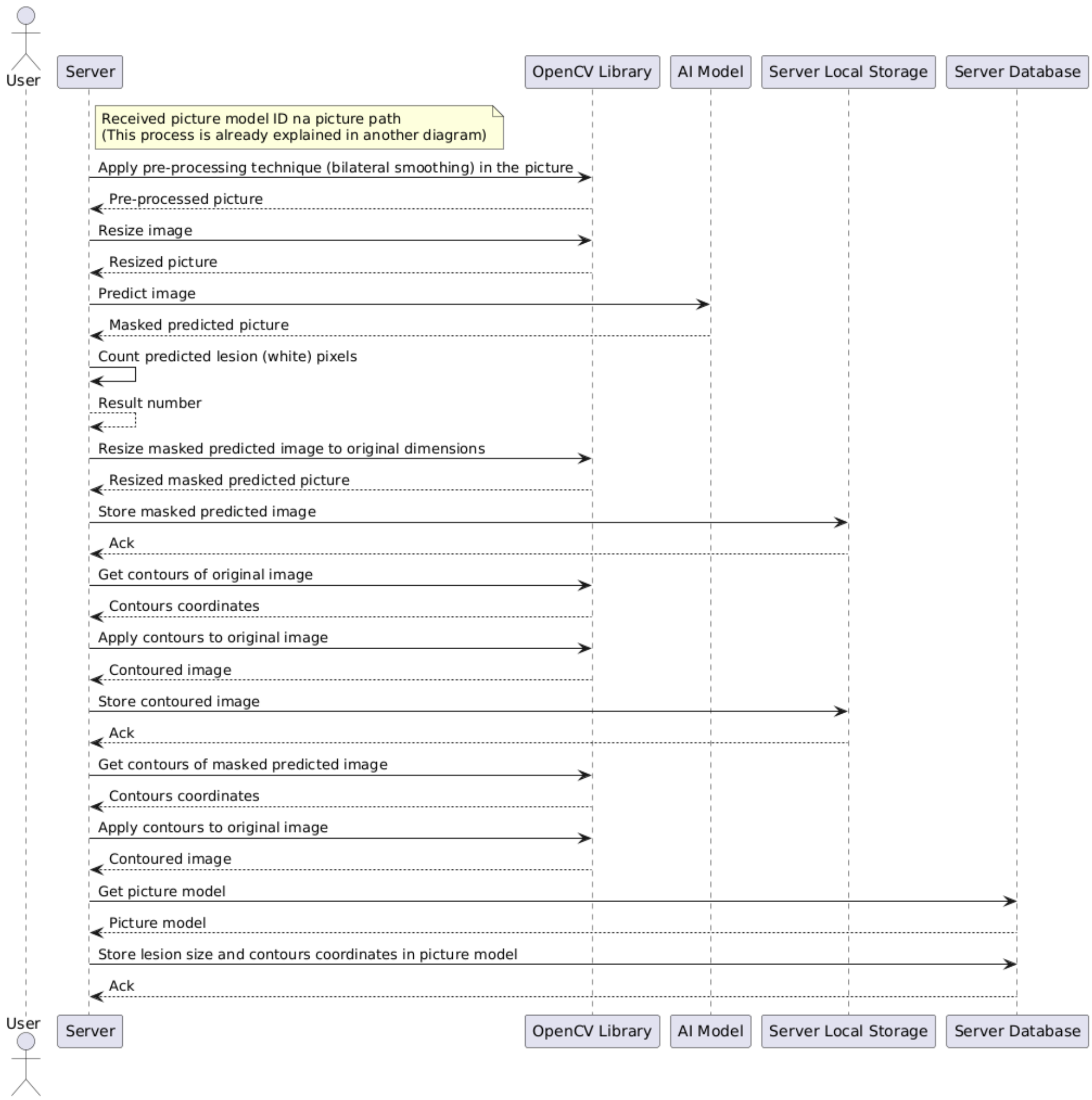

Figure 3.11: Sequence Diagram for Classifying Image with AI Model

Chapter 4

Psoriasis Support System Evaluation

4.1 Dataset

The dataset employed in this project consists of 45 high-resolution images of chronic plaque psoriasis, all sourced from Dermnet NZ, a reputable and extensively used dermatological resource. Importantly, the dataset includes only images with a Caucasian skin tone, a factor that is essential to consider in evaluating the model's generalizability to diverse populations. Limiting the focus to a single type of psoriasis, namely chronic plaque psoriasis, allows for consistency in lesion appearance, texture, and morphology, which is essential for the robust training and evaluation of segmentation models. This specific type of psoriasis was chosen because of its well-documented clinical presentation and prevalence, supporting the model's ability to handle characteristic lesion features and complexities effectively.

Given the relatively small dataset size, meticulous attention was paid to the selection of images, prioritizing those with high resolution, minimal noise, and well-defined lesion borders. These criteria were essential to optimize model learning and generalization under ideal conditions. Specifically, the selected images featured optimal lighting and clarity to eliminate ambiguity during segmentation, as well as high contrast between lesion and non-lesion areas to enhance segmentation accuracy.

Additionally, the dataset required manual modification to remove watermarks present in the images.

Minimal artifacts and noise were maintained to reduce the likelihood of false positives. The dataset was further processed and augmented using Roboflow, a comprehensive platform for managing and preparing image datasets for machine learning tasks. Roboflow facilitated tasks such as image annotation, pre-processing, and augmentation, while also supporting automatic dataset splitting into the following sets: training set (69 %, 31 images), validation set (20 %, 9 images), and test set (11 %, 5 images).

Roboflow provided various pre-processing and augmentation techniques aimed at en-

Figure 4.1: Examples of Chronic Plaque Psoriasis Dataset Images

hancing dataset quality. Pre-processing operations included Auto-Orient, Isolate Objects, Static Crop, Resize, Grayscale, Auto-Adjust Contrast, and Tile. These steps ensured uniformity in image sizes and enhanced contrast between lesion and non-lesion regions. Furthermore, a Gaussian filter was applied manually to smooth images and reduce noise.

Given the small dataset size, data augmentation was essential to improve model generalization. Roboflow supported augmentations such as Flip, 90° Rotate, Crop, Rotation, Shear, Grayscale, Hue, Saturation, Brightness, Exposure, Blur, Noise, Cutout, and Mosaic. These transformations introduced variability in the training images while preserving key features necessary for accurate lesion detection. Additionally, horizontal and vertical flipping with a 2x multiplier was employed to increase the number of samples and introduce positional variation, helping to mitigate the risk of overfitting.

Roboflow also played a crucial role in mask annotation, ensuring accurate labeling with binary masks, where lesion areas were marked as class 0. Although manual annotation was initially considered, Roboflow's Smart Polygon automation tool, which uses machine learning to suggest object shapes, significantly streamlined the annotation process. However, all automatic annotations were carefully reviewed and manually refined to ensure high-quality ground truth masks.

By employing these pre-processing and augmentation techniques, the models were better equipped to generalize from the limited dataset, ultimately leading to more consistent and reliable segmentation performance.

4.2 Semantic Segmentation AI Models

This section presents a comparative evaluation of various AI models designed for semantic segmentation of psoriasis lesions. The performance of these models is evaluated using key metrics, including precision, recall, F1 score, and prediction time per image, with the goal of identifying the most effective model for accurately detecting and classifying lesion pixels.

The semantic segmentation models tested include Mask R-CNN, U-Net, YOLOv8n, FCN (Fully Convolutional Networks), DeepLabV3+, BiSeNet (Bilateral Segmentation Network), HRNet (High-Resolution Network), PSPNet (Pyramid Scene Parsing Network), and SegNet (Segmentation Network). While all models are intended for pixel-wise segmentation, they differ significantly in their underlying architectures and optimization strategies. For instance, Mask R-CNN performs instance segmentation by addressing both object detection and segmentation, whereas U-Net and SegNet are designed for pixel-level binary segmentation, making them particularly suitable for the task of distinguishing lesion and non-lesion regions.

Given that the task at hand is binary classification—where each pixel is classified as either part of a lesion (class 1) or not (class 0)—semantic segmentation is the most appropriate approach. Unlike instance segmentation, which separates and labels individual objects, semantic segmentation treats all lesions as a single class, focusing on differentiating lesion pixels from background. Consequently, the output layer of the models is configured for binary classification with a sigmoid activation function.

To ensure a fair and equitable comparison among the models, each was configured to utilize convolutional filters with identical dimensions of 64, 128, and 256 filters at various stages. This consistency in filter configuration facilitates a robust assessment of the performance of each model.

The comparative analysis aims to identify the model that best balances computational efficiency with segmentation accuracy, particularly considering the medical imperative to accurately delineate lesion boundaries while minimizing both false positives and false negatives. This evaluation will provide insights into which model performs optimally for the specific task of psoriasis lesion segmentation, ensuring robust and reliable clinical outcomes.

Benchmark

To facilitate the evaluation of various AI models, a comprehensive benchmark was developed and made publicly available in a GitHub repository at https://github.com/tmsl9/thesis-ss-ai-models.git. The benchmark was designed to automate the experimentation process for different AI models, providing configurable settings to streamline testing. The repository includes the following key components:

- **data_augmentations/:** Contains implementations of data augmentation techniques not supported by the Roboflow platform.
- **pre_processing/:** Includes implementations of various pre-processing techniques, alongside a configurable script for applying and experimenting with these techniques.

- **models/:** Houses the implementations of the semantic segmentation AI models.
- **results_analysis/:** Contains automated tools for extracting and processing experimental results into a CSV file (full_report.csv), sorted by the F1 score metric. It also processes the CSV report, identifies the top 5 use-cases per metric, saves the analysis in a JSON file (report_analysis.json), and generates a secondary CSV file (report_analysis.csv) with results ranked by the number of top appearances across different metrics.
- **config.py:** A configurable script that allows for setting up experimental parameters, such as the number of epochs, AI model selection, pre-processing techniques, and data augmentation methods.
- **run_models.py:** A script responsible for training and evaluating one or more AI models. During each run, the results—including predicted images and metrics—are saved to a designated results folder.

This benchmark provides a robust framework for systematically comparing different AI models, ensuring reproducibility and facilitating efficient experimentation in the domain of semantic segmentation.

Initial Experiments with Semantic Segmentation Models

The initial phase of experimentation focused on evaluating three semantic segmentation models: Mask R-CNN, U-Net, and YOLOv8n. The primary objective was to assess each model's performance under various training conditions, in alignment with the project's goal of accurately detecting and segmenting psoriasis lesions. A range of key parameters were varied to examine the models' robustness, adaptability, and overall segmentation performance.

Experiment Parameters

- **Epochs:** The models were trained for both 50 and 75 epochs to observe how their performance evolved with moderate and extended training durations.
- **Learning Rate:** A fixed learning rate of 0.0001 was applied across all experiments, chosen for its effectiveness in previous studies to ensure stable convergence and mitigate overfitting risks.
- **Training Image Sizes:** Three training image sizes — 480x480, 640x640, and 720x720 pixels — were tested to investigate the models' adaptability to different spatial resolutions, assessing the impact of image resolution on segmentation accuracy. Resizing was applied to each image using bilinear interpolation to ensure consistency across the tested resolutions.

  - **45 Images:** A smaller, nearly uniform dataset with images of similar dimensions.
  - **63 Images:** An expanded dataset comprising the original 45 images and 18 additional images of varying resolutions, to test the models' ability to generalize across a broader range of conditions.

- **Pre-processing Techniques:** Various pre-processing methods were applied to assess their impact on model performance:
  - **None:** Raw, unaltered images.
  - **Gaussian Filter:** Applied to reduce noise and smooth the images, potentially enhancing lesion detection.
  - **Grayscale:** Conversion to grayscale to determine whether removing color information improves segmentation.
  - **Combination of Gaussian Filter and Grayscale:** This combination was explored to see if it further improved performance.

- **Data Augmentation:** Data augmentation techniques were employed to improve model generalization:
  - **No Augmentation:** Baseline performance without transformations.
  - **Horizontal and Vertical Flips:** Each image was flipped twice, horizontally and vertically, effectively increasing training data and introducing variability to reduce overfitting.

To run the experiments, masks were generated based on the labels provided in the dataset. YOLOv8n utilizes these labels for training and evaluation, while Mask R-CNN and U-Net require image masks. The dataset labels, provided by Roboflow, consist of a set of points representing lesions in the format: " <class_name> <point1_x*image_-width> <point1_y*image_height> ... . ". These points were calculated based on image dimensions, and the corresponding masks were generated with lesion pixels in white and background pixels in black.

Figure 4.2: Contoured and Masked Dataset Image

The goal of these experiments was to comprehensively assess the behavior of each model across a range of training conditions. By systematically varying parameters such as the number of epochs, training image sizes, and pre-processing techniques, the experiments aimed to uncover valuable insights into how each model responds to changes in training configurations, specifically in the context of psoriasis lesion segmentation.

This systematic approach provided a detailed analysis of the models' strengths and weaknesses. For example, varying epoch counts enabled an evaluation of when each model reached optimal performance, while guarding against overfitting. Understanding the relationship between training duration and model accuracy is vital for guiding future training strategies and efficiently allocating computational resources.

The experiments also explored how training image size affected the models' ability to detect and segment lesions. Given that clinical images often vary significantly in resolution, assessing how well the models handle different sizes is crucial for ensuring their robustness in real-world applications. These insights are critical for developing a model that can perform effectively across diverse imaging conditions.

The inclusion of varied pre-processing techniques was instrumental in understanding the models' adaptability. Comparing raw images with those enhanced via Gaussian filtering or grayscale conversion provided key information on how these modifications impacted segmentation outcomes. This analysis aids in refining pre-processing workflows, ensuring that the models are trained on image formats that best support accurate lesion detection.

Data augmentation played an important role in expanding the training dataset, especially given its small size. Applying horizontal and vertical flips introduced variability that helped the models generalize better to unseen data. Evaluating the effectiveness of these augmentations offered insights into how well each model adapted to variations in input data, which is crucial for developing a reliable system suitable for real-world applications.

Best results: The highest performance was achieved by the Mask R-CNN model, which attained an accuracy of 0.852, a recall of 0.854, and an F1 score of 0.853. This result was obtained under the following configuration:

- **Model:** Mask R-CNN;
- **Epochs:** 75;
- **Learning Rate:** 0.0001;
- **Training Image Size:** 480x480;
- **Dataset:** 45 images;
- **Pre-processing:** None;
- **Data Augmentations:** Horizontal and vertical flips (2x).

Interestingly, the second-best result was nearly identical to the best result, using only 50 epochs. Consequently, the 50-epoch configuration was adopted for future experiments to expedite the testing process, given the large number of use cases to explore.

These findings underscore the effectiveness of the Mask R-CNN model for psoriasis lesion segmentation, as it consistently outperformed other models. The use of a uniform dataset of 45 images with consistent resolution contributed to this strong performance, highlighting the importance of high-quality, uniform training data.

Pre-processing techniques, such as Gaussian filtering, had minimal impact on performance, indicating that models like Mask R-CNN and U-Net are capable of effectively learning from raw data. This suggests that pre-processing requirements may be simplified in future workflows.

Data augmentation proved beneficial, improving the robustness of the models and enhancing performance metrics. This was especially important given the limited dataset size, as augmentations enabled the models to generalize better to new, unseen data.

Among the models tested, Mask R-CNN and U-Net emerged as the top performers, each achieving accuracy above 85 %, recall above 81 %, and F1 scores exceeding 84 %. Conversely, YOLOv8n struggled to reach even 80 % accuracy, with its best metrics around 76 % for precision, recall, and F1 score. This discrepancy highlights the distinct capabilities of each model for the specific task of psoriasis lesion segmentation.

Conclusion: In conclusion, this analysis demonstrates that Mask R-CNN and U-Net are well-suited for the task of psoriasis segmentation, with Mask R-CNN delivering the best overall results. These findings emphasize the importance of careful parameter selection and data augmentation for improving model performance. As future experiments are conducted, these insights will inform further refinements and testing, with the ultimate goal of developing an optimal solution for psoriasis detection and segmentation.

Testing Each Pre-Processing Technique from Related Works for Psoriasis Lesion Segmentation

Following the promising results obtained with the Mask R-CNN model, the subsequent phase of the research focused on evaluating various pre-processing techniques derived from related literature. The goal was to assess the impact of these techniques on segmentation performance and to explore potential improvements in the model's efficacy for detecting psoriasis lesions.

The pre-processing techniques evaluated included adaptive median filter (with smoothing levels of 3, 5, and 7), background masking/region of interest, bilateral smooth filter (with parameters for moderate smoothing: diameter of 9, sigma color of 75, and sigma space of 75; stronger smoothing: diameter of 15, sigma color of 100, and sigma space of 100; and minimal smoothing: diameter of 5, sigma color of 50, and sigma space of 50), binary mask, color histogram analysis, contrast enhancement, dilation and erosion, Gaussian filter, grayscale conversion, hair removal (utilizing the DullRazor algorithm), Gabor filter (with parameters: ksize of 15, sigma of 2.0, theta of 0, lambd of 8.0, gamma of 0.5, and psi of 0), image resizing using bilinear interpolation (to dimensions of 480x480, 640x640, and 720x720), and sharpening.

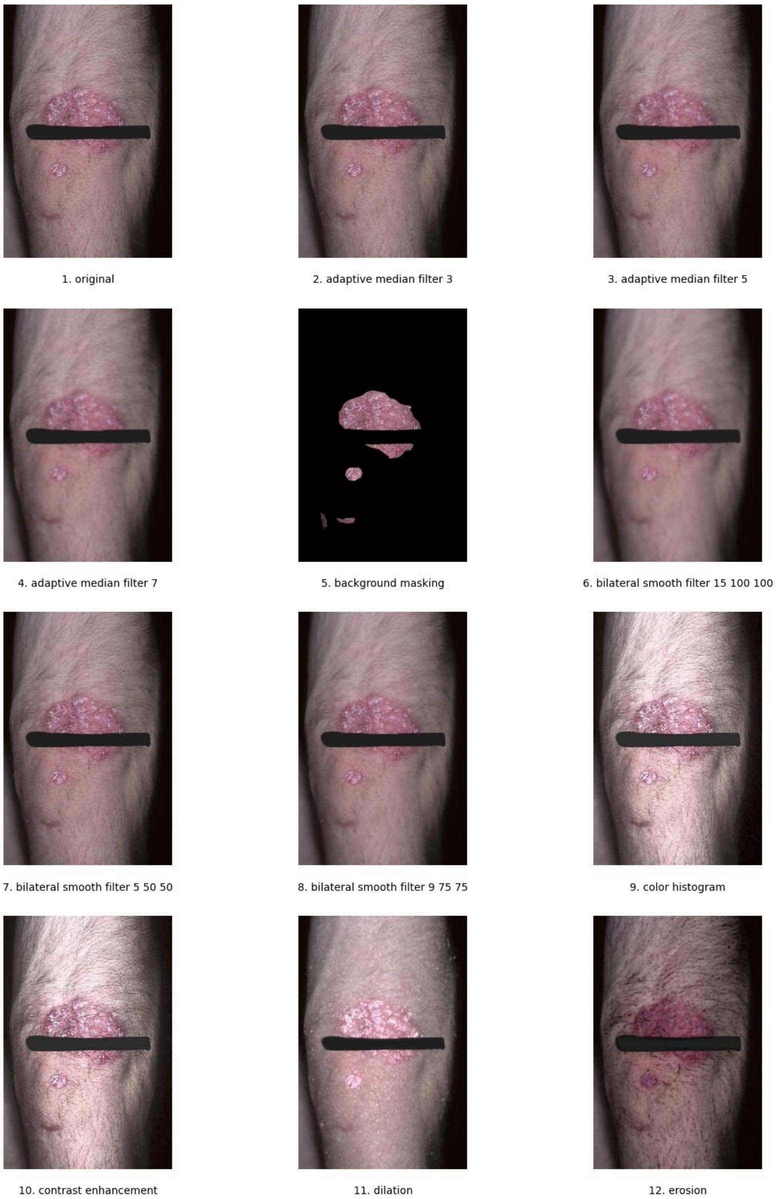

Figure 4.3: 1/2 Pre-processing Techniques Applied to a Dataset Image

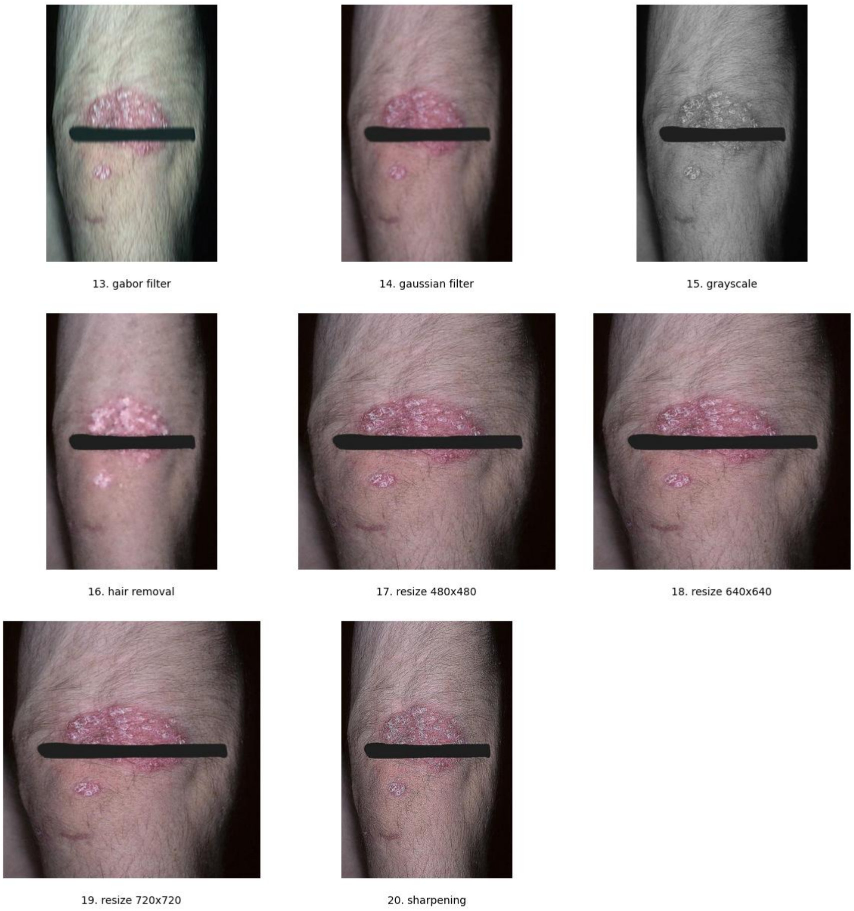

Figure 4.4: 2/2 Pre-processing Techniques Applied to a Dataset Image

These experiments aimed to identify which pre-processing techniques could yield the most significant improvements in segmentation when applied to the best-performing model configuration. The ultimate objective was to refine the pre-processing pipeline to optimize the overall effectiveness of psoriasis lesion detection and segmentation. By rigorously evaluating each technique, the optimal strategies that complement the strengths of the Mask R-CNN model can be determined.

The primary evaluation metric for these experiments was the F1 score, with the results tables sorted accordingly.

To date, 91 runs have been conducted using the previously established optimal configuration (model, epochs, learning rate, and training image size). Table 4.1 summarizes the best results achieved in this phase.

The results presented in Table 1 provide insights into the efficacy of various pre-processing techniques applied in psoriasis lesion segmentation. The bilateral smooth filter emerged as the most effective approach, achieving the highest F1 score of 0.871. This demonstrates the filter's ability to smooth images while preserving essential edge details, critical for identifying lesion boundaries. The moderate level of smoothing (15) allows the model to maintain sufficient detail, optimizing its ability to detect lesions accurately.

The adaptive median filter at a level of 3 ranked second with an F1 score of 0.864. The high recall (0.900) signifies this filter's effectiveness in capturing lesions with minimal noise while maintaining good precision. This suggests that adaptive median filtering successfully reduces unwanted artifacts without losing crucial structural details, aiding in lesion localization.

Resizing images to 720x720 ranked third, achieving an F1 score of 0.862. This approach balances image resolution with computational efficiency, enabling the model to process consistent dimensions with high precision (0.845) and recall (0.880), supporting effective segmentation without distorting lesion features.

Dilation ranked fourth, with an F1 score of 0.860. By expanding detected regions, dilation effectively enhances lesion boundaries. However, its lower recall (0.829) suggests a potential over-segmentation issue, useful for lesions with clear boundaries but less suitable for smaller or faint lesions.

Resizing to 480x480 ranked sixth, showing an F1 score of 0.856. This technique produced excellent recall (0.918), indicating an ability to detect lesions extensively, although precision was slightly lower. The method's strength lies in capturing more true positives, albeit with a minor increase in false positives.

The Gabor filter, with an F1 score of 0.846, ranked ninth. Its emphasis on texture and orientation highlights lesion patterns well, evidenced by a high recall (0.891). However, a precision of 0.805 suggests challenges in differentiating between lesion and non-lesion areas, leading to more false positives.

The hair removal technique, typically achieved through traditional methods like DullRazor, ranked fifteenth with an F1 score of 0.822. Although effective in removing hair artifacts, its lower recall (0.737) indicates it may miss some lesion areas, possibly due to excessive removal of essential image elements.

Sharpening ranked twenty-fifth with an F1 score of 0.784, benefiting precision (0.851) by enhancing edges but negatively impacting recall (0.727). This indicates a trade-off between clearer boundaries and potential noise amplification, impacting segmentation accuracy.

Grayscale conversion ranked thirtieth, with an F1 score of 0.760. While simplifying the image, it sacrifices color information crucial for accurate lesion differentiation. Despite a high recall (0.894), the lower precision (0.661) reflects increased false positives.

The color histogram technique ranked thirty-first, with an F1 score of 0.753. This approach's reliance on intensity distribution limits its capture of structural details, resulting in moderate precision (0.825) but lower recall (0.693), impacting its effectiveness.

Contrast enhancement, ranking forty-third with an F1 score of 0.723, improves feature visibility at the cost of image distortion, lowering recall (0.643). Although it demonstrates a good precision score (0.826), its reduced ability to detect subtle lesions limits overall effectiveness.

The Gaussian filter ranked sixty-sixth with an F1 score of 0.534, showing the drawbacks of excessive smoothing. While reducing noise, it removes essential details, resulting in low precision (0.512) and recall (0.557), thus detracting from segmentation performance.

Lastly, background masking ranked seventy-first with an F1 score of 0.296, reflecting the poorest performance. Although it achieves perfect precision (1.0), its extremely low recall (0.174) suggests it severely restricts lesion detection. This emphasizes the risk of overly aggressive masking approaches that exclude valuable lesion areas.

In summary, these findings underscore the importance of pre-processing methods that preserve or enhance lesion-relevant details, such as bilateral smoothing and adaptive median filtering, for improved segmentation. In contrast, techniques that overly simplify or blur the image, like the Gaussian filter and background masking, degrade performance significantly by eliminating critical information necessary for accurate lesion identification.

Position	Model	Total Images	Training Image Size	Pre-processing	Augmentations	Precision	Recall	F1 Score
1	Mask R-CNN	45	480x480	bilateral smooth filter 15 100 100	flip horizontal vertical & 2x	0.89	0.852	0.871
2	Mask R-CNN	45	480x480	adaptive median filter 3	no	0.831	0.9	0.864
3	Mask R-CNN	45	480x480	dilation	no	0.893	0.829	0.86
5	Mask R-CNN	45	480x480	resize 480x480	flip horizontal vertical & 2x	0.801	0.918	0.856
6	Mask R-CNN	45	480x480	gabor filter	flip horizontal vertical & 2x	0.805	0.891	0.846
8	Mask R-CNN	45	480x480	erosion	no	0.899	0.793	0.843
9	Mask R-CNN	45	640x640	resize 640x640	flip horizontal vertical & 2x	0.87	0.811	0.839
10	Mask R-CNN	45	480x480	adaptive median filter 7	flip horizontal vertical & 2x	0.922	0.764	0.836
11	Mask R-CNN	45	720x720	resize 720x720	no	0.846	0.826	0.836
12	Mask R-CNN	45	480x480	bilateral smooth filter 5 50 50	flip horizontal vertical & 2x	0.903	0.776	0.835
15	Mask R-CNN	45	480x480	hair removal	flip horizontal vertical & 2x	0.93	0.737	0.822
19	Mask R-CNN	63	480x480	bilateral smooth filter 9 75 75	flip horizontal vertical & 2x	0.754	0.834	0.792
26	Mask R-CNN	45	480x480	sharpening	no	0.851	0.727	0.784
35	Mask R-CNN	45	480x480	grayscale	flip horizontal vertical & 2x	0.661	0.894	0.76
85	Mask R-CNN	45	480x480	background masking	no	0.999	0.19	0.319

Table 4.1: Best Results of Each Related Works Pre-processing Technique Used

Testing the Pre-Processing Techniques from Related Works for Psoriasis Lesion Segmentation

The pre-processing techniques utilized in the related articles were replicated, with some studies applying multiple techniques simultaneously. The following pre-processing methods were employed across the articles:

- **Article 1:** employed a color histogram technique.
- **Article 2:** applied resizing to 720x720, hair removal, and a Gaussian filter.

- **Article 5:** used resizing (though the dimensions were unspecified), an adaptive median filter (with no filter level specified), and grayscale conversion.
- **Article 6:** implemented an adaptive median filter (with no filter level specified), Gaussian filtering, and contrast enhancement.
- **Article 8:** adopted resizing to 500x500, a Gaussian filter with a 7x7 mask, and background masking.
- **Article 12:** relied on contrast enhancement.
- **Article 13:** used grayscale conversion.
- **Article 15:** applied resizing to 224x224.
- **Article 18:** addressed noise removal, although the specific technique for noise removal was not specified.
- **Article 19:** utilized grayscale conversion, sharpening, an adaptive median filter (without a specified filter level), and a bilateral smooth filter.
- **Article 20:** applied resizing to 200x200 and a Gaussian filter.
- **Article 21:** included resizing, though the specific dimensions were not mentioned.
- **Article 22:** employed erosion, dilation, a Gabor filter, and a Gaussian filter.
- **Article 23:** applied resizing to 180x180 and a Gaussian filter.

These diverse techniques reflect the variety of approaches taken in the literature to enhance segmentation performance in psoriasis lesion detection tasks.

Technique	Article [1]	Article [2]	Article [5]	Article [6]	Article [8]	Article [12]	Article [13]	Article [15]	Article [18]	Article [19]	Article [20]	Article [21]
Color histogram	+	-	-	-	-	-	-	-	-	-	-	-
Resize	-	+	+	-	+	-	-	+	-	-	+	+
Hair removal	-	+	-	-	-	-	-	-	-	-	-	-
Gaussian filter	-	+	-	+	+	-	-	-	-	-	+	-
Adaptive median filter	-	-	+	-	-	-	-	-	-	-	-	-
Grayscale	-	-	+	-	-	-	+	-	-	+	-	-
Median filter	-	-	-	+	-	-	-	-	-	+	-	-
Contrast enhancement	-	-	-	+	-	+	-	-	-	-	-	-
Region of interest	-	-	-	+	-	-	-	-	-	-	-	-
Background masking	-	-	-	-	+	-	-	-	-	-	-	-
Noise removal*	-	-	-	-	-	-	-	-	+	-	-	-
Sharpening	-	-	-	-	-	-	-	-	-	+	-	-
Bilateral smooth filter	-	-	-	-	-	-	-	-	-	+	-	-
Binary mask	-	-	-	-	-	-	-	-	-	+	-	-
Erosion	-	-	-	-	-	-	-	-	-	-	-	-
Dilation	-	-	-	-	-	-	-	-	-	-	-	-
Gabor filter	-	-	-	-	-	-	-	-	-	-	-	-

Figure 4.5: Pre-processing Techniques Used in the Related Works

The following images illustrate the outcomes of applying the respective pre-processing techniques described in each article, as depicted in Figures 4.6, 4.7, 4.8, and 4.9. Each set of images corresponds to the pre-processing pipeline utilized by the articles, showcasing how these techniques influence the visual characteristics of the input data before segmentation. By examining these results, it is possible to observe the effects of various techniques, such as resizing, noise reduction, contrast enhancement, and filtering, on the image quality and lesion visibility. The visual outcomes serve as a basis for understanding how these methods impact the model's ability to identify and segment psoriasis lesions. Moreover, these images provide insight into the practical application of pre-processing strategies across different studies, allowing for a comparative analysis of their effectiveness in enhancing segmentation performance.

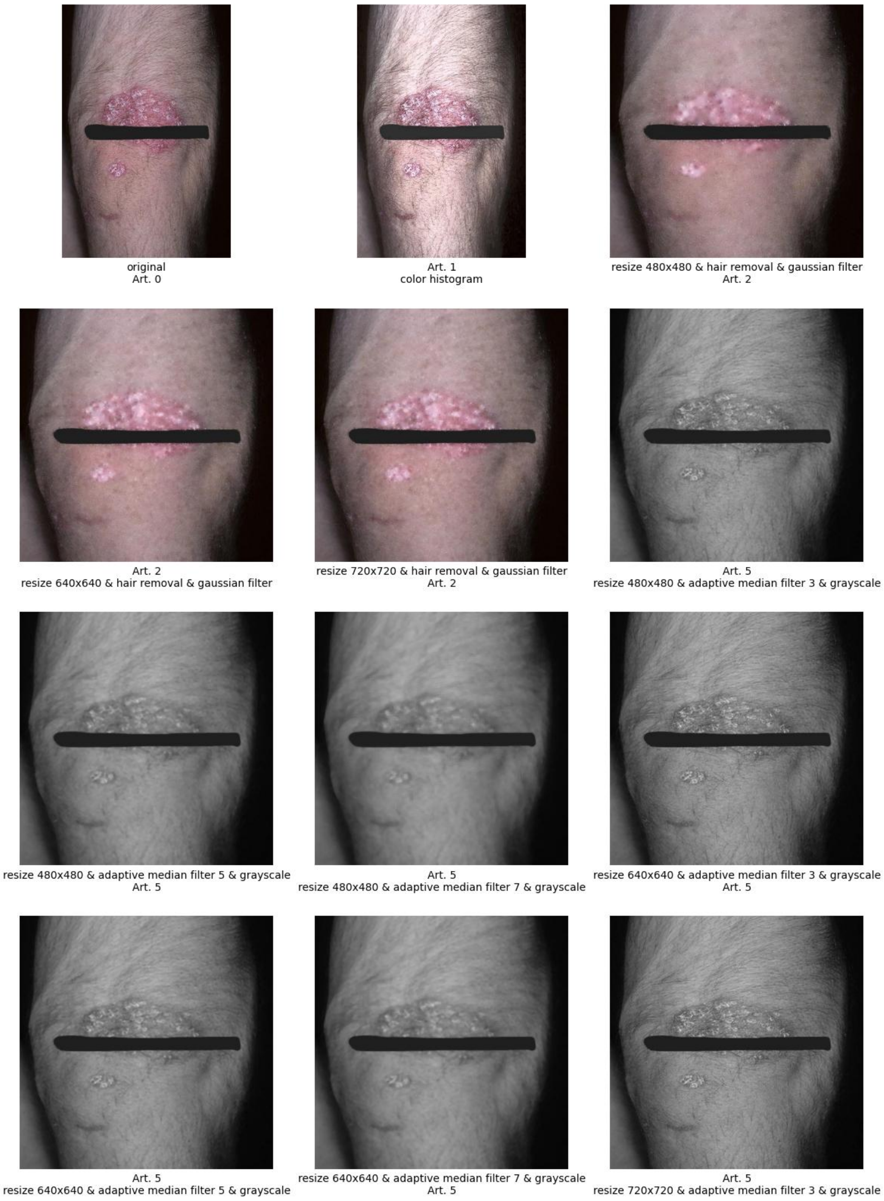

Figure 4.6: 1/4 Articles Used Pre-processing Techniques applied to a Dataset Image

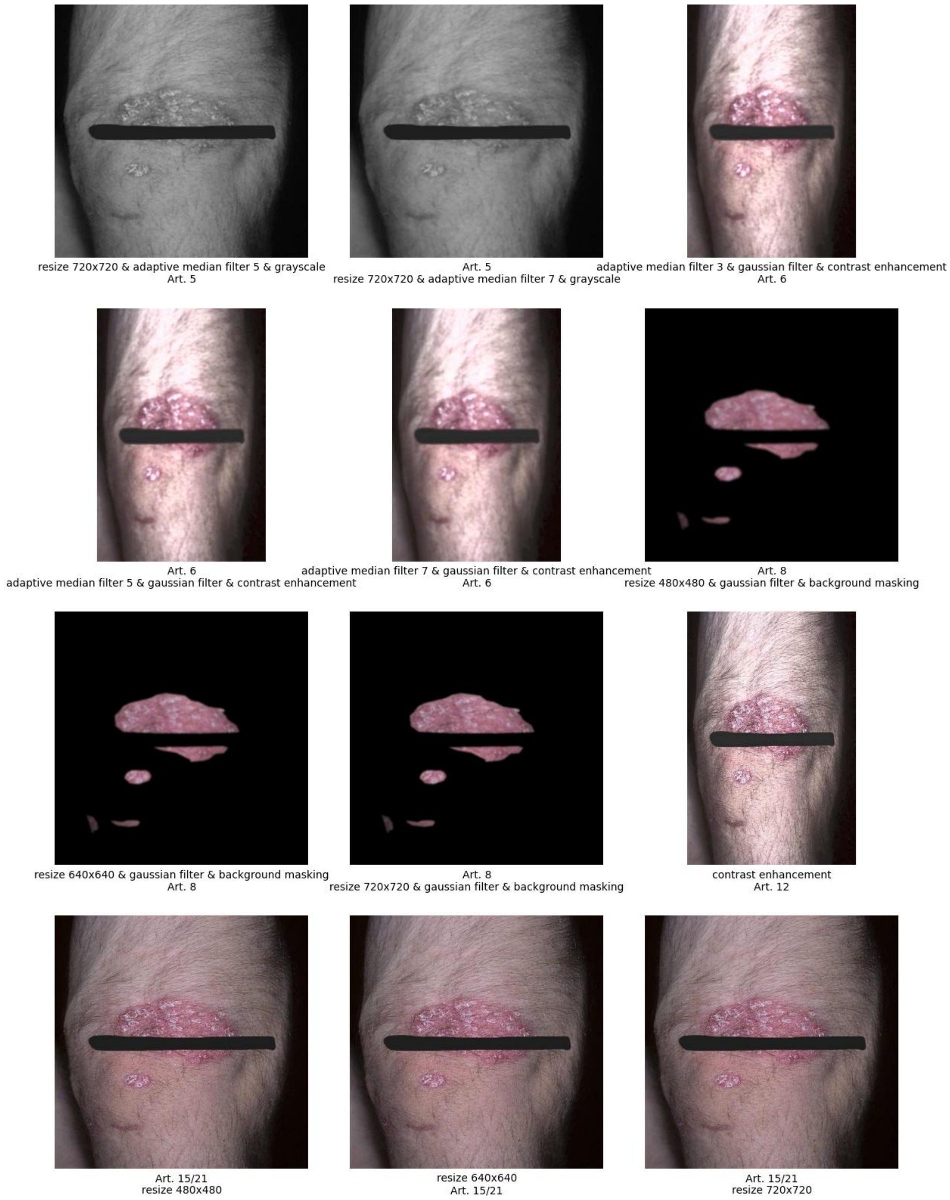

Figure 4.7: 2/4 Articles Used Pre-processing Techniques applied to a Dataset Image

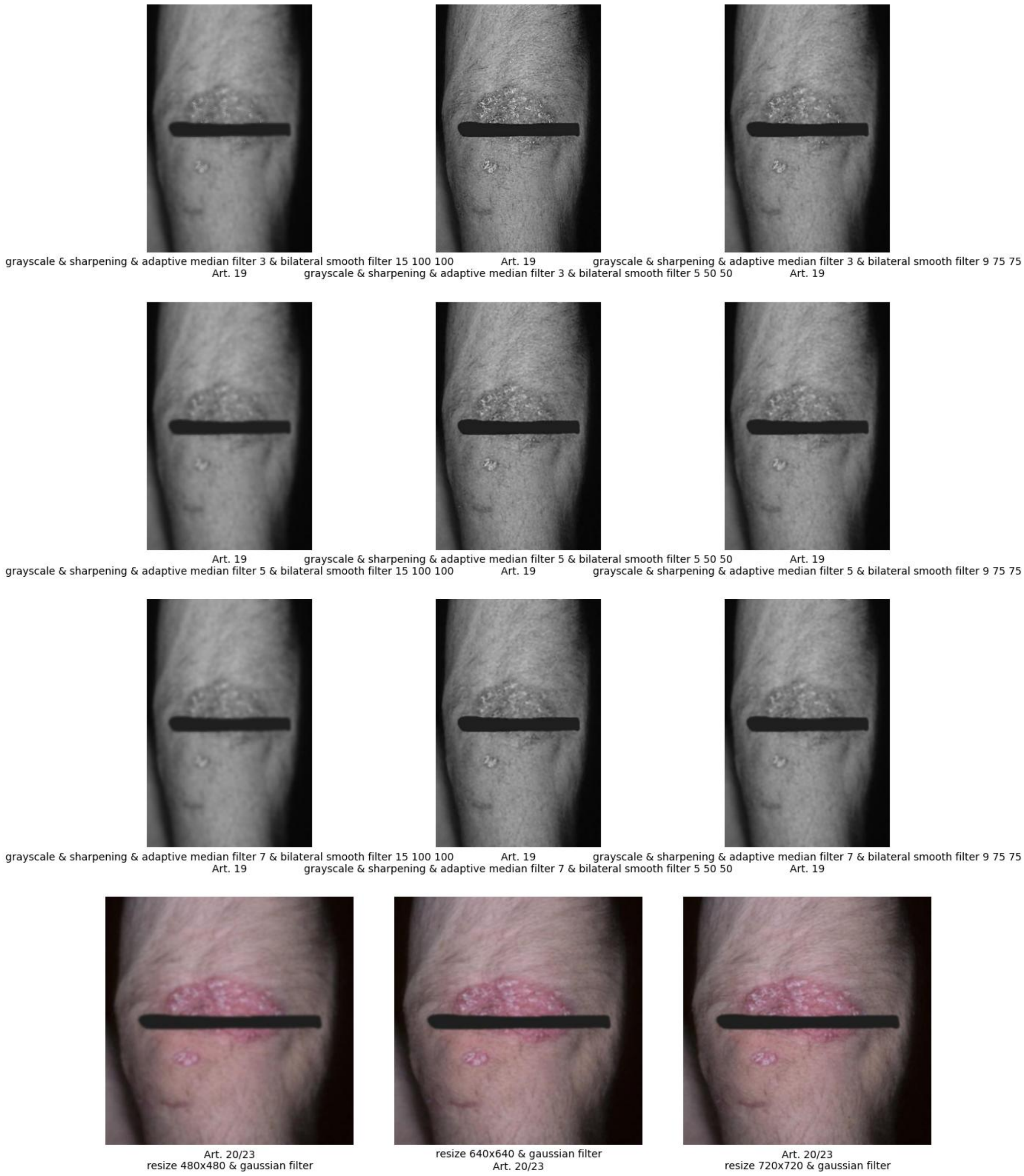

Figure 4.8: 3/4 Articles Used Pre-processing Techniques applied to a Dataset Image

erosion & dilation & gabor filter & gaussian filter
Art. 22

Figure 4.9: 4/4 Articles Used Pre-processing Techniques applied to a Dataset Image

The subsequent phase involved replicating the pre-processing techniques cited in the reviewed literature. The results from 212 experimental runs are encapsulated in Table 4.6, which presents the optimal outcomes for each use case. The analysis indicates considerable variability in the efficacy of these techniques, with certain approaches exhibiting a pronounced effect on model precision, recall, and F1 scores, while others produced less favorable results.

In the examined articles that reference resizing techniques, the image dimensions are notably low, reflecting a focus on binary classification (lesion versus healthy skin). However, for pixel-level classification, such reduced resolutions are suboptimal, as they limit the model's ability to accurately detect detailed lesion boundaries and achieve precise segmentation. Consequently, image dimensions of 480x480, 640x640, and 720x720 were employed to enhance the model's performance in lesion segmentation.

For example, the implementation of background masking, as illustrated in Article 8, resulted in highly inaccurate predictions. The model erroneously classified nearly the entire image as a lesion, achieving a near-perfect precision of 99.6 % but with only 19.7 % recall. This discrepancy can be attributed to the masking technique, which rendered the background black, thereby misleading the model into over-segmenting the lesion area. Such suboptimal results underscore the limitations of this technique in accurately distinguishing lesion boundaries when the background is manipulated in this manner.

Conversely, the combination of resizing, hair removal, and Gaussian filtering, as applied in Article 2, yielded comparatively strong performance, with an F1 score of 0.859. This finding suggests that pre-processing steps aimed at enhancing lesion visibility while preserving the overall structure of the image are advantageous for segmentation tasks. Similarly, the combination of resizing with Gaussian filtering and augmentations, as demonstrated in Article 20/23, also performed well, achieving a high precision of 0.922 and an F1 score of 0.842.

The results further indicate that more sophisticated pre-processing techniques, such as adaptive median filtering combined with contrast enhancement (Article 6), led to moderate improvements in performance, yielding an F1 score of 0.818. In contrast, grayscale conversion (Article 13) resulted in poor performance, with an F1 score of 0.643, likely due to the loss of crucial color information that is vital for accurately identifying lesions.

In conclusion, techniques that preserve the integrity of the lesion's visual characteristics—such as resizing and filtering—tend to demonstrate superior performance. Conversely, approaches that excessively manipulate the background or diminish critical color information, such as background masking or grayscale conversion, are associated with significant reductions in performance. These results emphasize the necessity for careful selection and rigorous testing of pre-processing methods to effectively improve lesion segmentation outcomes.

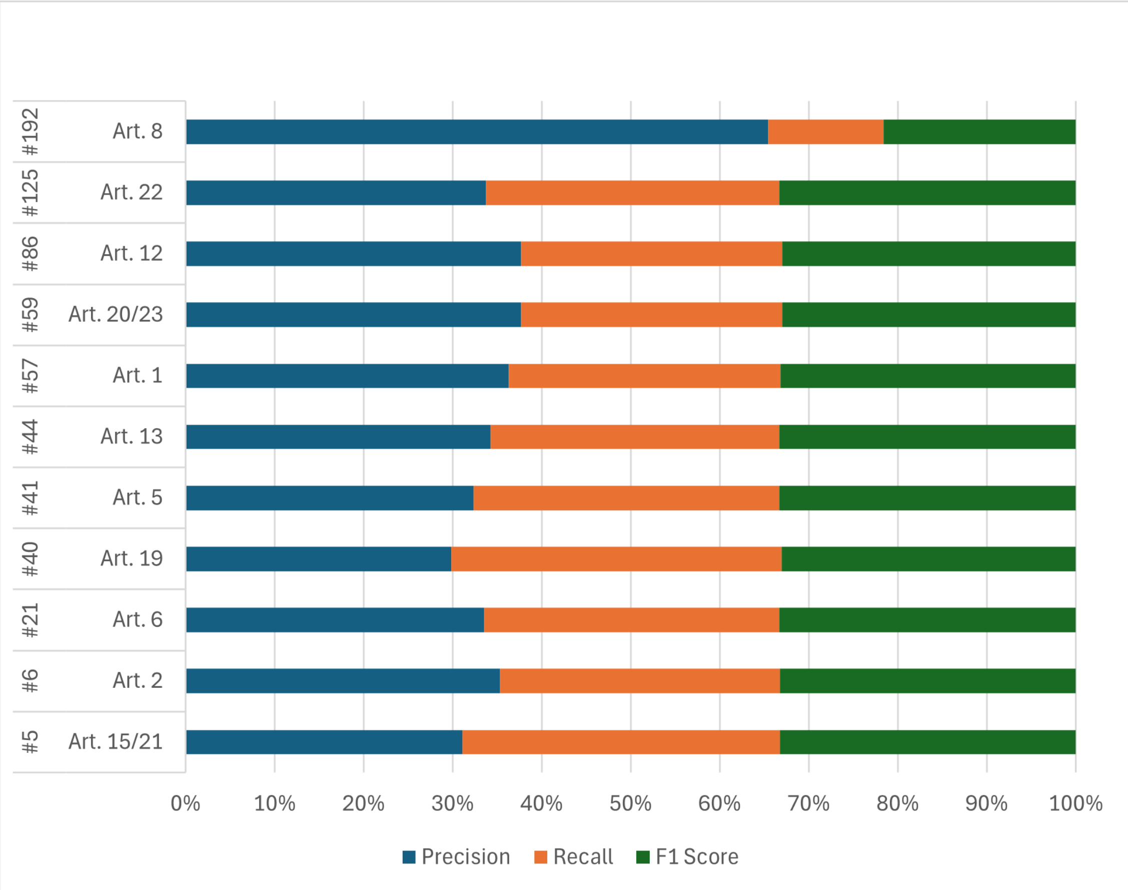

Figure 4.10: Best Results of Related Works Pre-processing Techniques

Testing Each Data Augmentation Technique from Related Works for Psoriasis Lesion Segmentation

The articles employed the following pre-processing techniques, with the specific parameter values being defined by the author, as the original articles did not provide these details:

- **Brightness: Random Adjustment Between -15 % and 15 %** This technique adjusts the brightness of the image randomly within a range of -15 % to +15 %, simulating varying lighting conditions. It enhances the model's robustness by making it more adaptable to different brightness levels that may occur in real-world scenarios.

- **Contrast Enhancement** Contrast enhancement increases the distinction between light and dark areas in the image, thereby making critical features more pronounced. This process improves the model's capacity to differentiate between different structures or objects, particularly in images with inherently low contrast.

- **Crop: Random Between 10 % and 20 %** Cropping randomly removes 10 % to 20 % of the original image. By exposing the model to varying image portions, this technique encourages it to focus on multiple areas, reducing the reliance on specific features located in fixed positions.

- **Flip: Horizontal or Vertical** Flipping the image horizontally or vertically introduces variability in orientation, enabling the model to recognize objects from different perspectives. This augmentation aids in ensuring that the model does not rely on the object's original orientation for identification.

- **Gaussian Blur: Random Between 0 and 1.7 Pixels** Gaussian blur is applied with a random standard deviation between 0 and 1.7 pixels, introducing a softening effect. This technique simulates slight defocus or noise, enhancing the model's ability to handle imperfect or blurry images.

- **Rotate: Random Between -45° and +45°** The images are rotated by a random angle within the range of -45° to +45°. This augmentation helps the model become less sensitive to the orientation of objects, allowing it to effectively learn from objects at various angles.

- **Scale: 3x** Scaling enlarges the image by a factor of 3, allowing the model to learn from images of different resolutions and sizes. This augmentation enables the model to detect features at various scales, enhancing its capacity for recognizing objects at multiple levels of detail.

- **Sharpness Enhancement: Kernel with Central Value 5 and -1 as Neighbours** This sharpening technique applies a filter kernel that accentuates edges and fine details in the image. By enhancing sharpness, this method improves the model's ability to detect small, intricate features, which are often crucial in detailed tasks such as segmentation.

- **Shear: Random Between -10° and +10° Horizontally or Vertically** Shearing introduces a slant in the image, applying a random angular distortion between -10°

and +10°, either horizontally or vertically. This augmentation enhances the model's ability to generalize across slight perspective changes or distortions in the objects.

- **Translate: 20 % Width and 20 % Height** Translation shifts the image by 20 % of its width and height in a random direction. This ensures that the model does not become overly reliant on objects being located in fixed regions of the image, thereby improving its generalization capabilities.

- **Zoom In: 2.0x** Zooming in by a factor of 2 focuses on the central region of the image. This technique helps the model capture finer details within the image, thereby improving its ability to learn from both overall patterns and minute features of the objects.

Figure 4.11: Data Augmentations Techniques applied to a Dataset Image

Several data augmentation techniques referenced in the related works are not natively supported by the Roboflow platform. These include contrast enhancement, scaling, sharpening, translation, and zooming in. Consequently, it was necessary to implement these techniques manually and apply them to the training images and their corresponding masks. Additionally, the labels were adjusted accordingly to ensure consistency across the datasets.

Currently, the two highest-performing results, correspond to both the 45-image dataset, which predominantly consists of images with similar dimensions, and the 63-image dataset, which contains images of varying dimensions. Notably, results employ bilateral smoothing and adaptive median filtering as pre-processing techniques, indicating that these methods significantly enhance overall model performance. This observation underscores the effectiveness of these configurations for lesion segmentation tasks and supports their potential application in future experiments.

To comprehensively evaluate the data augmentation techniques used in the related works, the following pre-processing techniques were employed. These configurations were established to benchmark the performance of the augmentation methods across varying datasets.

The specific configuration for benchmarking the data augmentation techniques is as follows:

- **Model:** Mask R-CNN;
- **Epochs:** 50;
- **Learning Rate:** 0.0001;
- **Training Image Size:** 480x480;
- **Dataset:**
  - 45 images (composed primarily of images with similar dimensions);
  - 63 images (comprising images with varying dimensions).
- **Pre-processing:**
  - Bilateral smooth filter: diameter=15, sigma_color=100, sigma_space=100;
  - Adaptive median filter: kernel_size=3;
  - Bilateral smooth filter: diameter=9, sigma_color=75, sigma_space=75.

This configuration was not only designed to evaluate the effectiveness of each data augmentation technique within a standardized experimental framework but was also selected based on the superior performance observed with this setup.

These techniques were prioritized due to their strong performance, as shown by the results from testing datasets with 45 images (largely uniform dimensions) and 63 images (varying dimensions).

The inclusion of data augmentation techniques in the current top 10 results reinforces the notion that augmentations are beneficial in improving model performance, particularly when dealing with limited datasets. The variety of augmentations applied—such as adjustments in brightness and image rotation—simulate real-world conditions, where images may be captured under varying lighting and angles, thus enhancing the model's ability to generalize across diverse scenarios.

Moreover, increasing the quantity of training input images has yielded significant improvements, particularly for datasets containing images of varying dimensions. This suggests that larger and more diverse training datasets enhance the model's capability to accurately classify lesions, even when presented with images of different sizes. This observation is particularly relevant in real-world applications, where image dimensions can vary significantly due to the use of different mobile devices for capturing photographs. Cell phone cameras, in particular, generate images of differing sizes, making this an important factor for model robustness in practical, real-life deployment

Position	Model	Total Images	Training Image Size	Pre-processing	Augmentations	Precision	Recall	F1 Score
1	Mask R-CNN	45	480x480	bilateral smooth filter 15 100 100	scale & 2x	0.864	0.894	0.879
2	Mask R-CNN	45	480x480	bilateral smooth filter 9 75 75	crop & 2x	0.878	0.87	0.874
3	Mask R-CNN	63	480x480	bilateral smooth filter 9 75 75	scale & 2x	0.867	0.877	0.872
4	Mask R-CNN	45	480x480	bilateral smooth filter 15 100 100	flip horizontal vertical & 2x	0.89	0.852	0.871
5	Mask R-CNN	45	480x480	bilateral smooth filter 9 75 75	translate & 2x	0.827	0.917	0.87

Table 4.2: Best Results of the Model with Data Augmentations

Testing the Data Augmentation Techniques from Related Works for Psoriasis Lesion Segmentation

The data augmentation techniques employed in each article were replicated to closely mimic their configurations, resulting in an additional 30 runs utilizing the previously specified model and pre-processing techniques. This approach ensures that the evaluation aligns with the methodologies described in the literature, thereby facilitating a more comprehensive assessment of the impact of data augmentations on model performance.

As illustrated in Table 4.9, Articles 23, 12, and 7 achieved positions among the top 25 results overall, each obtaining an F1 score exceeding 85 %. Article 10 recorded an F1 score of 83 %, while Article 21 was ranked 93rd out of 586 runs. The relatively lower performance of Article 21 can be attributed to the combination of scaling and zooming in on the same image, which generated images with excessive magnification, as depicted in Figure 4.1. This excessive zoom likely had a detrimental impact on model training, ultimately hindering its performance.

Technique	Article [7]	Article [10]	Article [12]	Article [21]	Article [23]
Cropping	+	-	-	-	+
Rotating	+	+	+	+	+
Scaling	+	-	-	+	-
Translation	+	-	+	-	-
Brightening	+	+	+	+	-
Flipping	-	+	+	+	+
Contrast adjustment	-	+	-	-	-
Sharpness adjustment	-	+	-	-	-
Gaussian filtering	-	+	-	-	-
Shear transforming	-	-	+	+	-
Zooming	-	-	+	+	-

Figure 4.12: Data Augmentations Techniques used in the Related Works

The results suggest that certain data augmentation techniques, when implemented judiciously, can significantly enhance model performance in the context of limited datasets. The presence of high F1 scores among the top-performing articles indicates that these models are effectively balancing precision and recall, thus demonstrating their capability to accurately classify lesions. However, the observed decline in performance for Article 21 underscores the importance of careful consideration when applying multiple augmentations, particularly those that alter the scale and focus of the images.

In conclusion, while data augmentation is a valuable strategy for improving model robustness, it is crucial to optimize the selection and configuration of these techniques to avoid introducing noise or artifacts that could impair learning. Future research should focus on refining augmentation strategies, particularly in scenarios involving images captured at varying resolutions and perspectives, to enhance model generalization across diverse real-world conditions.

Position	Article	Model	Total Images	Training Image Size	Pre-processing	Precision	Recall	F1 Score
10	23	Mask R-CNN	45	480x480	adaptive median filter 3	0.875	0.845	0.86
13	12	Mask R-CNN	45	480x480	bilateral smooth filter 15 100 100	0.816	0.904	0.858
20	7	Mask R-CNN	45	480x480	bilateral smooth filter 9 75 75	0.778	0.943	0.853
40	10	Mask R-CNN	45	480x480	bilateral smooth filter 9 75 75	0.907	0.779	0.838
79	21	Mask R-CNN	45	480x480	adaptive median filter 3	0.933	0.689	0.793

Table 4.3: Best Results of Related Works Data Augmentations Techniques

Testing the AI semantic models

The best 5 results wit >= 87 % F1 score until now use the following configuration:

- **Model:** Mask R-CNN;
- **Epochs:** 50;

- **Learning Rate:** 0.0001;
- **Training Image Size:** 480x480;
- **Dataset:**
  - 45 images (images mostly with same dimensions);
  - 63 images (different images dimensions).
- **Pre-processing:**
  - Bilateral smooth filter diameter=9 sigma_color=75 sigma_space=75;
  - Bilateral smooth filter diameter=15 sigma_color=100 sigma_space=100.
- **Data augmentations:**
  - crop;
  - flip;
  - scale;
  - translate.

The subsequent artificial intelligence models will be evaluated based on the specified configurations: U-Net, YOLOv8n, Fully Convolutional Networks (FCN), DeepLabV3+, BiSeNet (Bilateral Segmentation Network), HRNet (High-Resolution Network), PSPNet (Pyramid Scene Parsing Network), and SegNet (Segmentation Network). It is noteworthy that the Mask R-CNN model has been previously examined, providing a benchmark for comparison against the results obtained from the other models.

The analysis of the semantic segmentation models highlights their performance based on the F1 score, a critical metric that balances precision and recall. In total, 705 experiments were executed across different configurations and pre-processing techniques, focusing specifically on the evaluated models. This comprehensive assessment underscores the robustness of the findings and the reliability of the performance metrics obtained.

- **FCN (Position 1)** achieved the highest F1 score of 0.889, demonstrating superior accuracy in identifying true positives while minimizing false positives. Its precision (0.851) and recall (0.93) indicate a well-balanced model, aided by effective pre-processing and the "translate" augmentation.
- **UNet (Position 2)** followed closely with an F1 score of 0.885. This model exhibited a precision of 0.881 and recall of 0.889, suggesting reliable performance in segmentation tasks, bolstered by the "crop" augmentation technique.

- **SegNet (Position 4)** recorded an F1 score of 0.880, characterized by a precision of 0.882 and recall of 0.878. This indicates its robust capability in accurately segmenting images, supported by similar pre-processing methods.
- **BiseNet (Position 6)** achieved an F1 score of 0.879, with a precision of 0.825 and recall of 0.940. This suggests a tendency toward identifying true positives effectively, though it may experience a higher rate of false positives, facilitated by the "translate" augmentation.
- **Mask R-CNN (Position 7)** also obtained an F1 score of 0.879. It demonstrated a precision of 0.864 and recall of 0.894, reflecting a strong performance in segmentation tasks. The model's effectiveness was enhanced through specific pre-processing techniques and augmentations.
- **HRNet (Position 11)** scored 0.873, showing consistent performance with precision at 0.880 and recall at 0.866. Its results demonstrate a solid balance in segmenting lesions accurately.
- **DeepLabv3+ (Position 110)** exhibited an F1 score of 0.806, with a notable precision of 0.686 and recall of 0.976. While its recall indicates a strong ability to identify true positives, the lower precision suggests challenges in minimizing false positives.
- **YOLOv8n (Position 324)** achieved an F1 score of 0.711, with precision at 0.727 and recall at 0.695. This model's performance indicates room for improvement in both identifying true positives and reducing false positives.
- **PSPNet (Position 229)** recorded the lowest F1 score of 0.756. Its precision (0.618) and recall (0.973) reflect a significant disparity, highlighting issues with false positives despite a strong identification of true positives.

In conclusion, while FCN leads in performance metrics, other models like UNet and SegNet also exhibit strong capabilities. Models such as DeepLabv3+ and YOLOv8n indicate potential for enhancement in precision and overall segmentation accuracy.

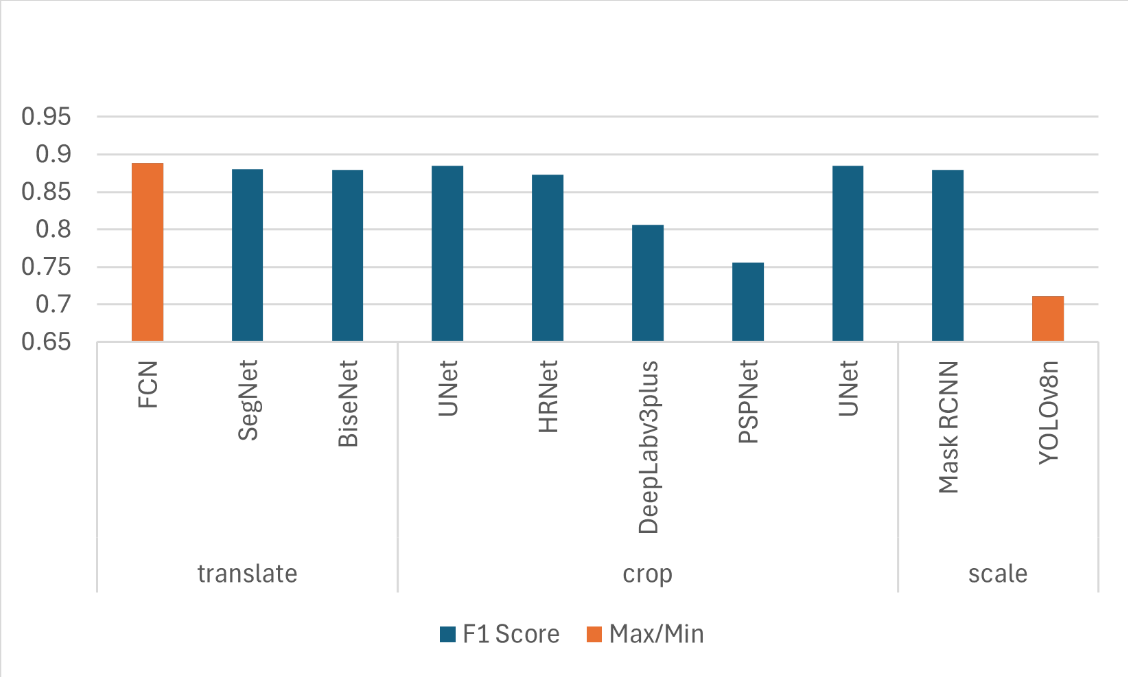

Figure 4.13: Top Performance Results for Each AI Model

Conclusion Overall, these findings underscore the critical importance of utilizing preprocessing techniques that preserve image integrity while simultaneously optimizing performance for psoriasis lesion segmentation. The integration of data augmentation methods, such as rotation, flipping, cropping, and zooming, significantly enhances the robustness of the models by simulating diverse real-world scenarios. Moreover, the employment of filters like bilateral smooth and adaptive median filters effectively reduces noise and enhances the clarity of the lesions, contributing to improved segmentation accuracy. Collectively, these strategies not only enhance model performance but also ensure that the segmentation outcomes are reliable and applicable in clinical settings.

4.3 Influence of External and Dermatological Conditions on AI Models Performance

In the realm of image/instance/semantic segmentation, particularly concerning various skin conditions such as chronic plaque psoriasis, atopic dermatitis, and even healthy skin, several factors can significantly impede model performance. Understanding these obstacles is crucial for improving predictive accuracy across different skin types and conditions.

One major obstacle is lighting conditions. Overexposed images pose a notable challenge, as excessive brightness often results in a washout effect, obscuring critical details necessary for accurate segmentation. In such cases, the model struggles to detect clear lesion boundaries or skin textures, complicating its ability to distinguish lesions from surrounding tissue. Conversely, underexposed images, where low light and shadows prevail, obscure essential features and details. Shadows, in particular, may conceal lesions or create artificial boundaries, leading to false negatives or misclassifications in the segmentation process.

Hair obstruction presents another persistent challenge. Hair often covers the area of interest, further complicating the model's ability to segment skin accurately. The varying shades and textures of hair can blend with both skin and lesion tones, which makes distinguishing healthy skin from lesions more difficult. This issue becomes even more prominent when dealing with skin conditions that present similar colors or textures to the hair, such as hyperpigmentation or darker lesion patches.

The influence of reddish backgrounds also complicates the segmentation process, especially when lesions themselves exhibit red tones, as in chronic plaque psoriasis or inflamed areas of atopic dermatitis. When the background shares a similar reddish hue, the model struggles to differentiate between healthy skin and affected regions, increasing the likelihood of false negatives or missed detections.

Reddish skin tones further exacerbate this challenge. For conditions like eczema or dermatitis, where redness is a symptom, distinguishing between inflamed areas and healthy but reddish skin becomes particularly difficult. The subtle contrasts often confuse the model, affecting its ability to achieve accurate predictions, especially in terms of recall rates for lesions that blend into surrounding skin.

Moreover, variations in the angles at which images are captured create additional complexity. Photos taken from oblique perspectives may distort the appearance of lesions or skin folds, further impeding the model's ability to maintain accuracy. Angle variability introduces distortions that make it difficult for the model to consistently identify and segment lesions, regardless of the skin condition.

Finally, healthy skin, though seemingly simpler, introduces its own set of challenges. In some cases, areas like lashes, nose holes, and ears, which are naturally darker or shadowed, confuse the model, leading to false positives. These healthy areas are sometimes misclassified as lesions due to the shadow effects or darker pigmentation, which closely resemble the characteristics of lesion boundaries in skin conditions.

In this study, a comparative analysis was conducted using images of both healthy skin and various skin conditions. For the Caucasian skin sample with chronic plaque psoriasis, images were manipulated to replicate different conditions, including variations in lighting, the presence of hair, and changes in color attributes. Furthermore, distinct datasets for darker skin tones (Liu et al., 2023) were employed to demonstrate the model's adaptability and performance across diverse skin tones and disease presentations. Images representing other skin conditions were sourced from a Kaggle dataset, providing a comprehensive basis for evaluating model generalization across different types of dermatological presentations.

Baseline Condition: Original Psoriasis Case

Caucasian Skin Color with Chronic Plaque Psoriasis Results: Precision = 0.819, Recall = 0.949, F1 Score = 0.879 (Figure 4.14). This condition exemplifies the optimal scenario for model performance, evidenced by both high precision and recall values. The model operates effectively on standard Caucasian skin under ideal lighting conditions, affirming its capabilities. Given that the original image yields strong predictive values, any subsequent low performance in manipulated use cases cannot be attributed to the quality of the original image, but rather to the additional challenges presented in those specific conditions.

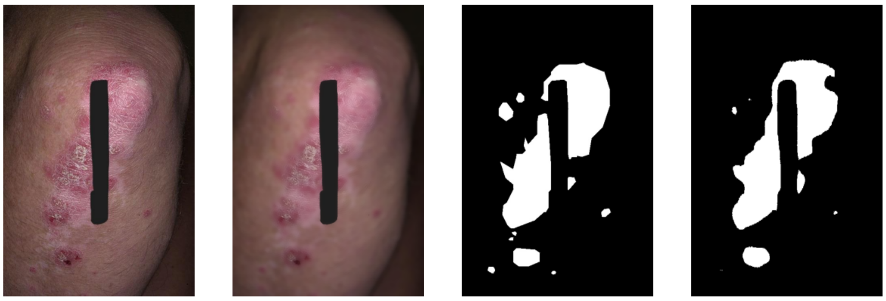

Figure 4.14: Prediction of Caucasian Skin with Chronic Plaque Psoriasis

Lighting and Image Acquisition Conditions

Caucasian Skin Color with Chronic Plaque Psoriasis: Overexposed Results: Precision = 0.522, Recall = 0.843, F1 Score = 0.645 (Figure 4.15). The model experiences a significant decline in precision when analyzing overexposed images, where excessive brightness leads to the loss of critical details, resulting in a higher frequency of false positives. The high recall value indicates that a considerable number of lesions are still identified; however, this further emphasizes the model's difficulties in accurately distinguishing true positive cases within washed-out regions. The diminished precision can be attributed to the high brightness, which generates lighter-toned areas in the lesions that are often misidentified as positive predictions. This situation illustrates the challenges posed by photographic conditions, such as the use of camera flash, which can create white spots and ultimately impair the model's ability to achieve accurate segmentation.

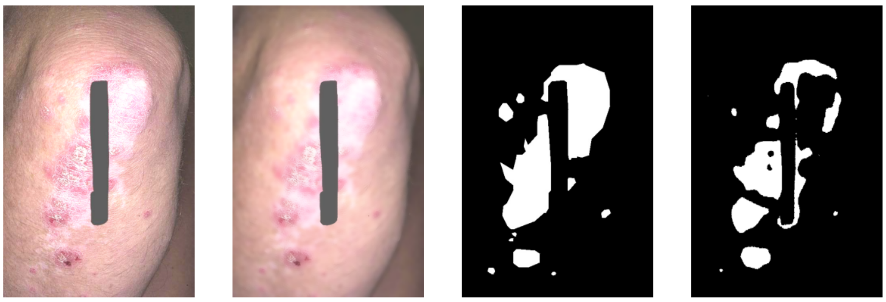

Figure 4.15: Prediction of Caucasian Skin with Chronic Plaque Psoriasis With High Brightness

Caucasian Skin Color with Chronic Plaque Psoriasis: Underexposed Results: Precision = 0.970, Recall = 0.305, F1 Score = 0.464 (Figure 4.16). The model demonstrates a high precision rate, indicating that when it makes a prediction, it is typically accurate. However, the notably low recall reveals a significant shortcoming in the model's ability to detect lesions, which can be attributed to the effects of underexposure. The decreased brightness results in darker skin tones that obscure the distinction between healthy skin and lesions, leading the model to erroneously classify most skin regions as non healthy tissue. This tendency to predict skin with dark tones further underscores the model's struggle in suboptimal lighting conditions. These findings underscore the critical importance of adequate lighting for effective segmentation and accurate lesion detection.

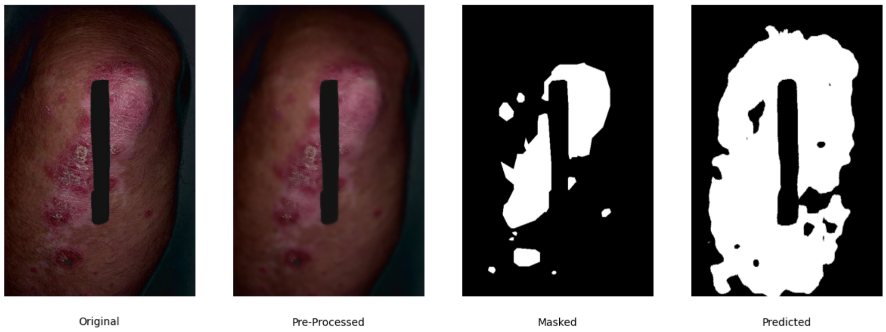

Figure 4.16: Prediction of Caucasian Skin with Chronic Plaque Psoriasis With Low Brightness

Caucasian Skin Color with Chronic Plaque Psoriasis: Low Resolution Results: Precision = 0.767, Recall = 0.963, F1 Score = 0.854 (Figure 4.17). In low-resolution images, the model demonstrates high recall, indicating its ability to identify a substantial number of lesions. Nevertheless, the decrease in precision suggests that the model may misclassify non-lesion areas as psoriasis due to blurriness. This reflects the model's tendency to prioritize sensitivity, ensuring that most potential lesions are detected, albeit at the cost of increased false positives.

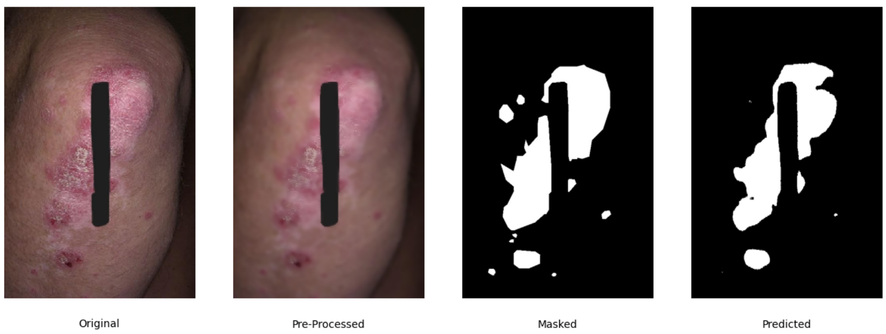

Figure 4.17: Prediction of Caucasian Skin with Chronic Plaque Psoriasis with Low Resolution

Caucasian Skin Color with Chronic Plaque Psoriasis: Warped Results: Precision = 0.543, Recall = 0.821, F1 Score = 0.654 (Figure 4.18). In the case of warped images, the model achieves high recall but suffers from decreased precision. The distortion caused by the angle of capture leads to missed details, which impacts the model's ability to accurately differentiate between healthy skin and lesions. When images are taken from oblique angles, the three-dimensional structure of the skin can result in foreshortening or obscuring of lesions, making them less detectable. This underscores the necessity of capturing images as parallel to the skin surface as possible to enhance segmentation accuracy and ensure that the model can effectively identify and classify lesions.

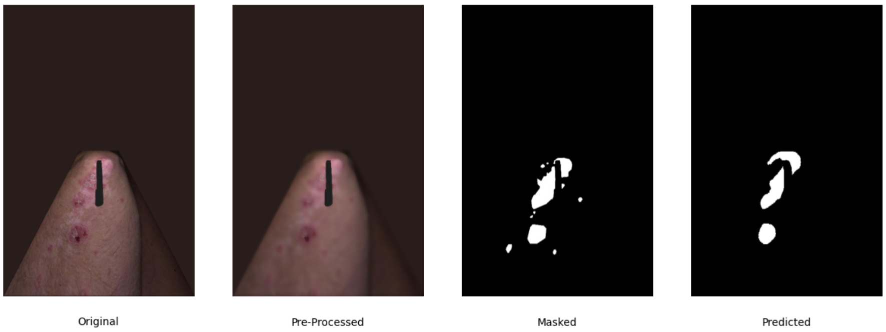

Figure 4.18: Prediction of Warped Caucasian Skin with Chronic Plaque Psoriasis

Caucasian Skin Color with Chronic Plaque Psoriasis: Red Color Background Results: Precision = 0.824, Recall = 0.277, F1 Score = 0.414 (Figure 4.19). The model's performance in this scenario reveals high precision; however, the low recall signifies substantial difficulties in identifying lesions due to the confusion caused by the reddish background. The overlap in color palettes complicates the model's ability to accurately detect lesions, resulting in missed predictions.

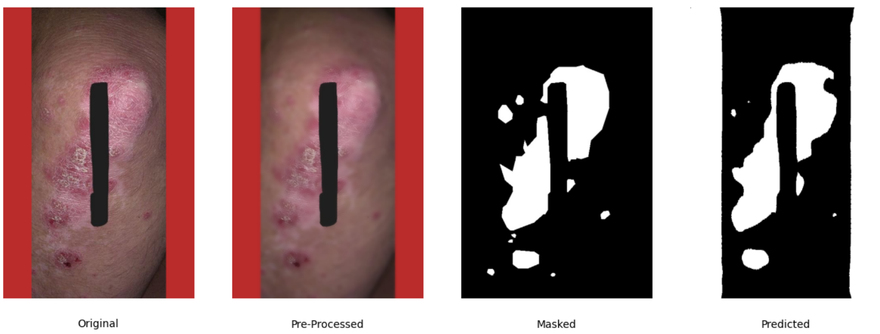

Figure 4.19: Prediction of Caucasian Skin with Chronic Plaque Psoriasis with Reddish Background

Skin Tone and Pigmentation Variations

Psoriasis on Black Skin Results: Precision = 0.823, Recall = 0.064, F1 Score = 0.118 (Figure 4.20). In the context of dark skin, the model exhibits high precision; however, the extremely low recall indicates that numerous true positives are overlooked. The model appears to misclassify much of the skin as lesions, likely due to the brownish tone resembling the crust color of lesions. Moreover, given that the lesions on dark skin are typically white, it appears that the model struggles to accurately predict scabbed lesions, as evidenced by the absence of predictions for these lesions in the output.

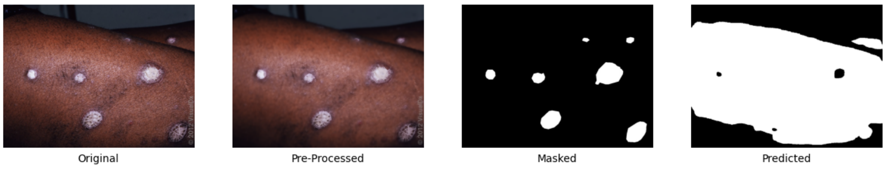

Figure 4.20: Prediction of Dark Skin with Chronic Plaque Psoriasis

Caucasian Skin Color with Chronic Plaque Psoriasis: Reddish Results: Precision = 0.981, Recall = 0.164, F1 Score = 0.281 (Figure 4.21). Although the model demonstrates exceptional precision when it does predict a lesion, the alarmingly low recall reflects a significant challenge. The reddish tones of the skin can effectively camouflage many lesions, leading to missed detections. This finding underscores the necessity for improved training data that encompasses a broader spectrum of skin tones, particularly those with reddish hues, to enhance detection capability.

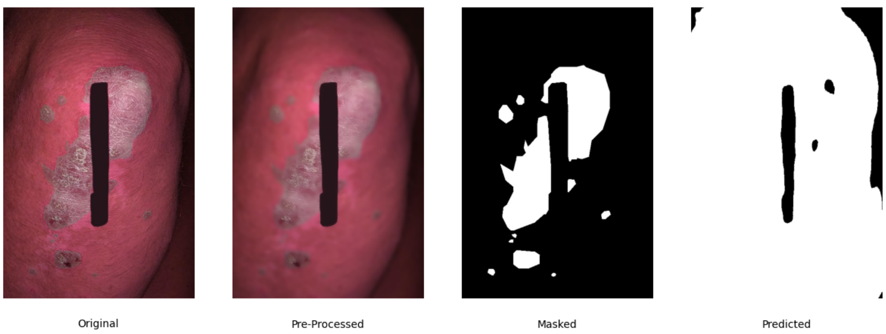

Figure 4.21: Prediction of Reddened Caucasian Skin with Chronic Plaque Psoriasis

Impact of Other Dermatological Conditions

Caucasian Skin Color with Actinic Keratosis Results: Precision = 0.642, Recall = 0.651, F1 Score = 0.646 (Figure 4.22). Actinic keratosis (Wang et al., 2020) (Cockerell, 2003) lesions are characterized by reddish patches with a rough texture, but they lack the crusting typical of plaque psoriasis. The model performs well in this scenario due to its ability to detect reddish areas, which explains the balanced precision and recall scores. However, the differences in lesion texture and the absence of plaque-like scaling likely caused some difficulties in exact lesion differentiation.

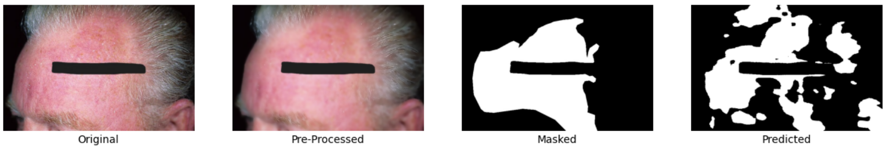

Figure 4.22: Prediction of Caucasian Skin with Actinic Keratosis

Caucasian Skin Color with Allergic Contact Dermatitis Results: Precision = 0.372, Recall = 0.832, F1 Score = 0.514 (Figure 4.23). The lesions in allergic contact dermatitis (Tramontana et al., 2023) (Streit, 2001) often appear reddish and inflamed, which aligns with the model's strengths in identifying red-toned regions. This explains the high recall score, but the differing texture, characterized by peeling skin and an absence of the thick plaques seen in psoriasis, reduces precision. The model struggles to precisely distinguish between these two conditions due to the subtle variations in surface texture.

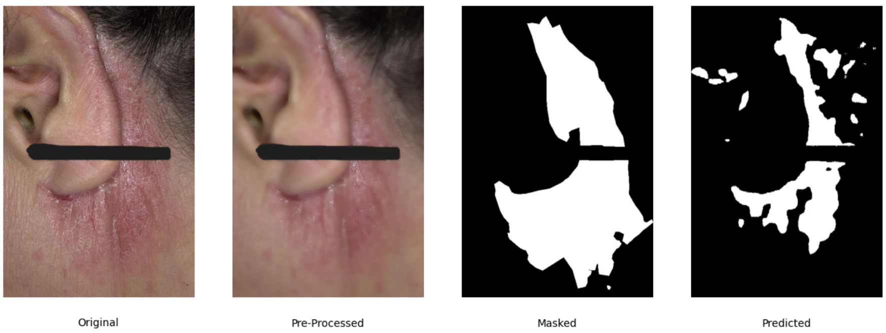

Figure 4.23: Prediction of Caucasian Skin with Allergic Contact Dermatitis

Caucasian Skin Color with Atopic Dermatitis Results: Precision = 0.002, Recall =

1.0, F1 Score = 0.003 (Figure 4.24). Atopic dermatitis (Weidinger et al., 2018) often lacks strong color contrast with healthy skin, appearing more like a healing scar. The absence of significant redness or crusting, key indicators for psoriasis detection, explains the low precision score. Despite achieving full recall, the model misinterprets healthy areas as lesions, indicating that it struggles to distinguish this subtle skin condition from more visible psoriasis lesions.

Figure 4.24: Prediction of Caucasian Skin with Atopic Dermatitis

Caucasian Skin Color with Basal Cell Carcinoma Results: Precision = 0.452, Recall = 0.991, F1 Score = 0.621 (Figure 4.25). Basal cell carcinoma (Ramachandran et al., 2000) lesions are reddish but have a smoother texture compared to psoriasis plaques. The red hue enables the model to predict some of the lesion as psoriasis, reflected in the high recall score. However, the differing texture and lack of scaling in carcinoma reduce precision, as the model incorrectly includes some healthy areas.

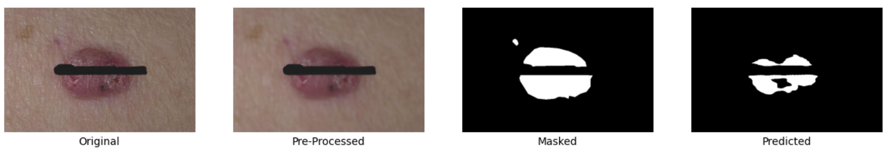

Figure 4.25: Prediction of Caucasian Skin with Basal Cell Carcinoma

Caucasian Skin Color with Crest Syndrome Results: Precision = 0.388, Recall = 0.518, F1 Score = 0.44 (Figure 4.26). Crest syndrome (Arana-Guajardo and Villarreal-Alarcón, 2017) presents with fine vascular patterns on the skin, creating localized reddish tones. The model's performance is moderate, as it can detect the red areas but struggles with the more complex vascular lesions, which differ from the broader, crusted plaques seen in psoriasis. The precision and recall values reflect this complexity, as the model is not fully adapted to such subtle variations.

Figure 4.26: Prediction of Caucasian Skin with Crest Syndrome

Caucasian Skin Color with Hemangioma Results: Precision = 0.959, Recall = 0.609, F1 Score = 0.74 (Figure 4.27). Hemangiomas (Tuder et al., 1987) are bright red lesions with sharp contrast against healthy skin, which explains the high precision. The model effectively distinguishes the lesion but also misclassifies some darker areas of healthy skin, such as shadows near the nose and ears, as psoriasis due to lighting effects. This leads to a moderate recall score, as these darker regions confuse the model.

Figure 4.27: Prediction of Caucasian Skin with Hemangioma

Caucasian Skin Color with Herpes Cutaneous Results: Precision = 0.843, Recall = 0.694, F1 Score = 0.761 (Figure 4.28). Herpes cutaneous (Leinweber et al., 2006) lesions have a distinctive appearance, often red and inflamed with small vesicles. The model performs well due to the reddish tone of the lesions, which it detects similarly to psoriasis plaques. However, the differing texture causes the model to miss some areas, resulting in a slightly lower recall.

Figure 4.28: Prediction of Caucasian Skin with Herpes Cutaneous

Caucasian Skin Color with Lupus Chronic Cutaneous Results: Precision = 0.266, Recall = 0.579, F1 Score = 0.364 (Figure 4.29). Lupus lesions (Pramatarov, 2004) often exhibit a mix of red tones and crusting, which the model can partially detect. However, its inability to fully distinguish between the specific crust patterns of lupus and those of psoriasis affects its performance. Additionally, the model incorrectly classifies features like lashes, nasal openings, and other reddish areas of healthy skin as lesions, leading to reduced precision.

Figure 4.29: Prediction of Caucasian Skin with Lupus Chronic Cutaneous

Caucasian Skin Color with Malignant Melanoma Results: Precision = 0.0, Recall = 0.0, F1 Score = 0.0 (Figure 4.30). Malignant melanoma (Clark et al., 1969), characterized by dark, homogeneous patches, presents no red tones or scaling, which are key to the model's psoriasis detection capabilities. The model fails to detect the lesion entirely, as the absence of typical psoriasis features (redness and crust) leads to zero predictions. The model does misclassify dark areas, such as the mouth, as lesions due to the similar darker pigmentation.

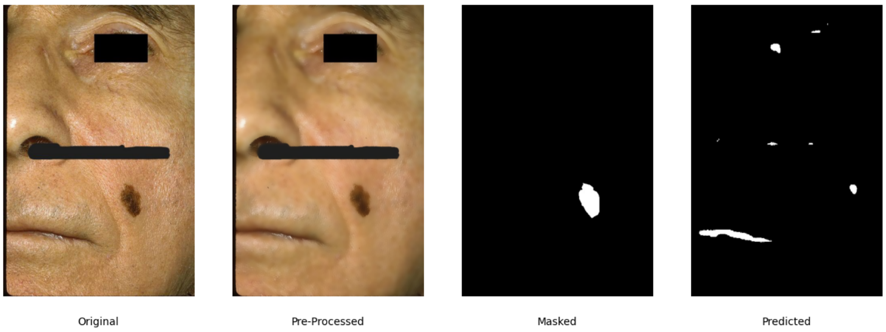

Figure 4.30: Prediction of Caucasian Skin with Malignant Melanoma

Caucasian Skin Color with Perioral Dermatitis Results: Precision = 0.056, Recall = 1.0, F1 Score = 0.106 (Figure 4.31). Perioral dermatitis (Lipozenčić and Hadžavdić, 2014) presents with slight redness but lacks the distinct crusting of psoriasis. The model detects only the reddest parts of the lesion, resulting in a low precision score. Most of the lesion remains undetected due to its subtle appearance, which reduces the model's ability to distinguish it from psoriasis lesions accurately.

Figure 4.31: Prediction of Caucasian Skin with Perioral Dermatitis

Caucasian Skin Color with Rosacea Results: Precision = 0.092, Recall = 0.755, F1 Score = 0.164 (Figure 4.32). Rosacea (Sharma et al., 2022) has some redness but minimal crusting, and the overexposure in the image likely contributed to the model's high recall but low precision. The bright light obscured the red areas, causing the model to fail in detecting the lesion accurately. The impact of lighting confirms earlier findings that overexposure degrades model performance, particularly when the red tones are diminished.

Figure 4.32: Prediction of Caucasian Skin with Rosacea

Caucasian Skin Color with Venous Lake Results: Precision = 0.007, Recall = 0.359, F1 Score = 0.014 (Figure 4.33). Venous lakes (Moore, 2020) appear as dark, homogeneous lesions, lacking any redness or crusting. The model, trained to detect red psoriasis lesions, fails to detect the venous lake accurately, resulting in near-zero precision. Only a small part of the lesion is detected, likely due to confusion with redder areas near the mouth.

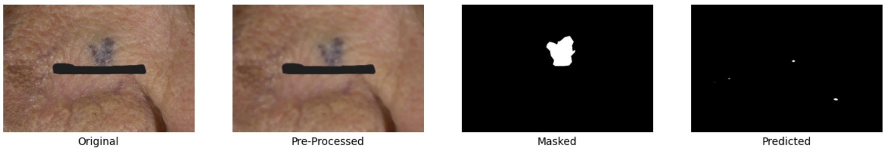

Figure 4.33: Prediction of Caucasian Skin with Venous Lake

Caucasian Skin Color with Xanthomas Results: Precision = 0.305, Recall = 0.146, F1 Score = 0.198 (Figure 4.34). Xanthomas (Vinay et al., 2017), characterized by yellowish deposits on the skin, contrast with psoriasis lesions in both color and texture. The model struggles to detect the xanthomas correctly, predicting some areas near the eye, likely due to wrinkles in older skin, as psoriasis. The precision score reflects this misclassification, as the model identifies non-lesion areas incorrectly.

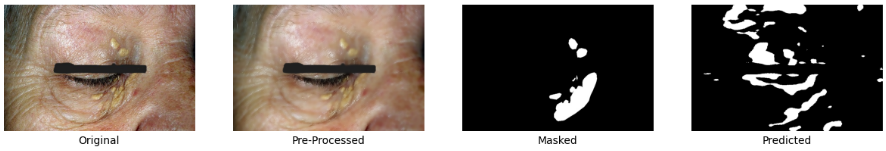

Figure 4.34: Prediction of Caucasian Skin with Xanthomas

Challenges in Predicting Hair and Skin Interactions

Caucasian Skin Color with Chronic Plaque Psoriasis: Haired Results: Precision = 0.947, Recall = 0.603, F1 Score = 0.737 (Figure 4.35). The model demonstrates a high precision rate, signifying that when lesions are predicted, the predictions are predominantly accurate. However, the presence of hair results in a reduced recall value. This reduction can be attributed to the fact that hairs, when interspersed with the lesion, create a darker tonal quality that the model erroneously identifies as a component of the lesion. Such findings underscore the inherent challenges associated with accurately segmenting lesions in scenarios where hair obfuscates or alters the visual characteristics of the lesion.

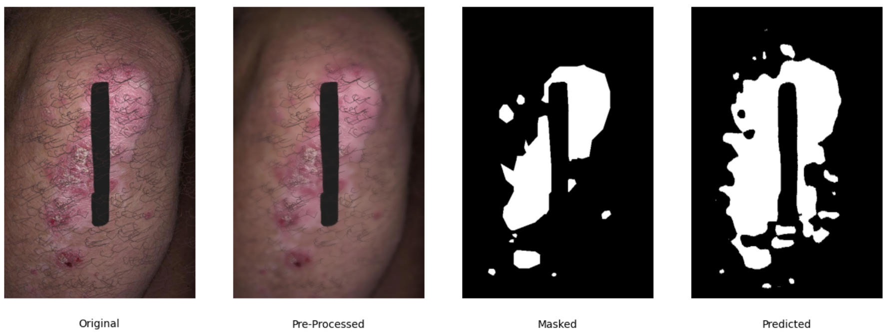

Figure 4.35: Prediction of Caucasian Skin with Hairs with Chronic Plaque Psoriasis

Healthy Skin Comparisons

Caucasian Healthy Skin Color Results: Precision = 0.0, Recall = 0.0, F1 Score = 0.0 (Figure 4.36). In the case of healthy skin, the model demonstrates no predictive capability, as expected, since there are no lesions to segment. The results indicate that the model is highly specialized for detecting pathological skin conditions, such as chronic plaque psoriasis, and does not produce false positives when presented with images of healthy skin. This underscores the model's robustness in distinguishing between diseased and non-diseased skin, effectively ignoring areas that do not exhibit psoriasis characteristics. These findings highlight the model's capacity to prevent unnecessary or incorrect classifications in cases where no skin abnormalities are present.

Figure 4.36: Prediction of Caucasian Healthy Skin

Conclusion

In conclusion, the findings indicate that while the model demonstrates strong performance under ideal conditions, it faces significant challenges when applied to manipulated scenarios and diverse skin conditions. For example, overexposed images, particularly those with white spots or excessive brightness, drastically reduce lesion detection accuracy, especially when using flash. These results underscore the importance of maintaining optimal lighting during image capture to avoid misclassification and performance degradation.

Moreover, the model's performance on healthy skin reveals an additional layer of complexity. Darker areas, such as eyelashes, nose holes, and ears, were frequently misclassified as lesions due to their resemblance to typical lesion characteristics, highlighting the need for further refinement in detecting healthy regions accurately.

Images taken at oblique angles also led to distorted segmentations, emphasizing the importance of capturing images that are as parallel to the skin surface as possible. Proper alignment helps to minimize distortions and avoid foreshortening, which can obscure lesions or produce incorrect segmentations.

The model's limitations in handling reddish backgrounds and skin tones, particularly in cases of inflamed skin, highlight the necessity for continued model training and refinement. Improving the model's ability to distinguish between inflamed skin and lesion-affected areas across various conditions, such as atopic dermatitis, chronic plaque psoriasis, and eczema, is essential for broader clinical applicability.

By addressing these challenges—enhancing performance across different lighting conditions, skin tones, and diseases—future models can become more reliable tools for both clinical practice and patient self-monitoring. Additionally, further training with larger datasets that encompass a broader range of skin diseases and healthy skin variations is necessary to improve overall prediction accuracy and reduce false positives or negatives.

Chapter 5

Conclusion

In summary, this project presents a comprehensive solution for the semantic segmentation of psoriasis lesions through a mobile application designed to enhance patient engagement and improve tracking of lesion progression. The developed application facilitates user authentication, image capture, and secure data synchronization with a robust server, addressing the pressing need for accessible dermatological care. Unlike existing solutions, which typically achieve over 90 % accuracy in binary image segmentation (determining the presence or absence of psoriasis), our approach emphasizes pixel-level classification, resulting in superior accuracy specifically for chronic psoriasis lesions in white skin tones under average lighting conditions. This complexity underscores the project's innovative nature, as it leverages a carefully curated dataset to optimize model performance.

The architecture of our system, built on state-of-the-art technologies such as TensorFlow and Flask, demonstrates the potential for scalable and efficient deployment in clinical practice. Through rigorous evaluation, the project highlights the critical role of image quality and the influence of real-world conditions on model accuracy. The introduction of advanced pre-processing techniques and data augmentation has further enhanced the robustness of our AI models, providing meaningful insights into the progression of psoriasis lesions.

A significant contribution of this project is the creation of a public benchmark, available to all researchers and practitioners who wish to reuse it. All results from our evaluations are documented in the accompanying repository, ensuring transparency and accessibility for future research. Additionally, our best-trained model is also made available in the repository, facilitating further exploration and application of our findings.

Looking ahead, several avenues for future work are identified. Expanding the dataset to include a wider variety of psoriasis types, skin tones, and image dimensions will bolster the model's predictive accuracy and scalability. Additionally, benchmarking heavier configurations, such as increased epochs and learning rates, may yield further improvements in performance. Application enhancements, including the development of an offline mode and the exploration of augmented reality features for real-world area calculations of lesions, are also proposed to augment the user experience. Finally, ongoing server improvements will ensure the continued security and efficiency of data management.

Collaborations with public or private clinical settings could support initial lesion size estimation, allowing doctors to provide a reliable baseline measurement that the system can use as a reference for future analysis. Additionally, such partnerships would enable the detection of significant changes in lesion size over time, prompting alerts for healthcare providers when notable developments are observed. This dual approach not only establishes an accurate initial size but also facilitates proactive patient care by enabling clinics to contact patients for timely consultations. Remote medical appointments would also be possible, where physicians could review lesion histories during discussions, enhancing continuity of care through streamlined communication and timely intervention.

In conclusion, this project represents a significant advancement in the field of dermatological applications, combining sophisticated AI techniques with practical mobile solutions to empower patients and facilitate improved clinical outcomes.

References

Ahmed, M. H., Ema, R. R. and Islam, T. (2019). An automated dermatological images segmentation based on a new hybrid intelligent aco-ga algorithm and diseases identification using tsvm classifier, *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)* pp. 1–6.
URL: *https://api.semanticscholar.org/CorpusID:209460206* 31

Allen, G. and Owens, M. (2006). The definitive guide to sqlite. 6

Arana-Guajardo, A. and Villarreal-Alarcón, M. (2017). Crest syndrome: Clinical expression of the disease., *Journal of clinical rheumatology : practical reports on rheumatic & musculoskeletal diseases* **23 5**: 285. 100

Badrinarayanan, V., Kendall, A. and Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**: 2481–2495. 13

Bandi, A., Adapa, P. V. S. R. and Kuchi, Y. E. V. P. K. (2023). The power of generative ai: A review of requirements, models, input–output formats, evaluation metrics, and challenges, *Future Internet* . 14, 15

Bhagat, S. (2022). Review on mobile application development based on flutter platform, *International Journal for Research in Applied Science and Engineering Technology* .
URL: *https://api.semanticscholar.org/CorpusID:251901205* 5

Boehncke, W.-H. (2018). Systemic inflammation and cardiovascular comorbidity in psoriasis patients: Causes and consequences, *Frontiers in Immunology* **9**.
URL: *https://api.semanticscholar.org/CorpusID:4591630* 1

Clark, W., From, L., Bernardino, E. A. and Mihm, M. (1969). The histogenesis and biologic behavior of primary human malignant melanomas of the skin., *Cancer research* **29 3**: 705–27. 102

Cockerell, C. (2003). Pathology and pathobiology of the actinic (solar) keratosis, *British Journal of Dermatology* **149**. 99

Cruz, J. C. D., Garcia, R. G., Dimaunahan, E. D., Labaclado, J. J., Reyes, G. B., Riomero, H. M. C. P. and Salamatin, P. M. (2019). Eczema, hives and psoriasis detection with the application of local binary pattern, color histogram, svm and rgb-hsv color space, *Proceedings of the 2019 9th International Conference on Biomedical Engineering and Technology* . 42

Elsayad, A. M., Dhaifallah, M. A. and Nassef, A. (2018). Analysis and diagnosis of erythemato-squamous diseases using chaid decision trees, *2018 15th International Multi-Conference on Systems, Signals & Devices (SSD)* pp. 252–262.
URL: *https://api.semanticscholar.org/CorpusID:54461322* 33

Gairola, A. K., Kumar, V. and Sahoo, A. (2023). Deep learning based multiple skin disease classification in indian territory, *2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT)* pp. 559–564. 38

George, Y., Aldeen, M. and Garnavi, R. (2020). Automatic scale severity assessment method in psoriasis skin images using local descriptors, *IEEE Journal of Biomedical and Health Informatics* **24**: 577–585. 35

H, M., A, D., Krishnan, A. and D, K. S. (2021). Automated detection of skin and nail disorders using convolutional neural networks, *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)* pp. 1309–1316. 34

Halder, M., Kabir, H. M. E., Rity, A. M. and Vowmik, A. (2022). An automated detection of scabies skin disease using image processing and cnn, *2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI)* pp. 1–6.
URL: *https://api.semanticscholar.org/CorpusID:258322275* 32

Hameed, N., Shabut, A. M. and Hossain, M. A. (2018). A computer-aided diagnosis system for classifying prominent skin lesions using machine learning, *2018 10th Computer Science and Electronic Engineering (CEEC)* pp. 186–191.
URL: *https://api.semanticscholar.org/CorpusID:90263399* 28

Hammad, M., Pławiak, P., Elaffendi, M., El-latif, A. and Latif, A. A. A. (2023). Enhanced deep learning approach for accurate eczema and psoriasis skin detection, *Sensors (Basel, Switzerland)* **23**. 42

Ivan, G. I., Elizarov, A. I. and Artyom, S. V. (2023). Image processing methods to increase the distinguishability of objects, **12780**: 127800E – 127800E–4. 16

Jarad, T. S. and Dawood, A. J. (2018). Computer-aided diagnosis psoriasis lesion using skin color and texture features, *International Journal of Computing and Digital Systems* **07**: 145–154.
URL: *https://api.semanticscholar.org/CorpusID:54716917* 28

Jardeleza, S. G. S., Jose, J. C., Villaverde, J. and Latina, M. A. (2023). 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)* pp. 231–236. 40

Karunanayake, R. K. M. S. K., Dananjaya, W., Peiris, M. S. H., Gunatileka, B. R. I. S., Lokuliyana, S. and Kuruppu, A. (2020). Cureto: Skin diseases detection using image processing and cnn, *2020 14th International Conference on Innovations in Information Technology (IIT)* pp. 1–6. 37

Lateef, F. and Ruichek, Y. (2019). Survey on semantic segmentation using deep learning techniques, *Neurocomputing* **338**: 321–348. 14

Leinweber, B., Kerl, H. and Cerroni, L. (2006). Histopathologic features of cutaneous herpes virus infections (herpes simplex, herpes varicella/zoster): A broad spectrum of presentations with common pseudolymphomatous aspects, *The American Journal of Surgical Pathology* **30**: 50–58. 101

Lipozenčić, J. and Hadžavdić, S. L. (2014). Perioral dermatitis., *Clinics in dermatology* **32 1**: 125–30. 103

Liu, Y., Primiero, C., Kulkarni, V., Soyer, H. and Betz-Stablein, B. (2023). Artificial intelligence for the classification of pigmented skin lesions in populations with skin of color: A systematic review, *Dermatology (Basel, Switzerland)* **239**: 499 – 513. 93

Mathur, S. and Jain, T. (2023). Dermatological disease detection employing transfer learning, *2023 11th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON)* pp. 1–6. 40

Miller, D. and Brown, E. (2017). Artificial intelligence in medical practice: The question to the answer?, *The American journal of medicine* **131 2**: 129–133. 1

Moore, R. (2020). Venous lake, *Definitions* . 104

Nimesh, V. and Weerasinghe, R. (2021). Differential diagnosis of ringworm and eczema using image processing and deep learning, *2021 21st International Conference on Advances in ICT for Emerging Regions (ICter)* pp. 147–152. 41

Noronha, S. S., Mehta, M., Garg, D., Kotecha, K. and Abraham, A. (2023). Deep learning-based dermatological condition detection: A systematic review with recent methods, datasets, challenges, and future directions, *IEEE Access* **11**: 140348–140381. 38

Pramatarov, K. (2004). Chronic cutaneous lupus erythematosus–clinical spectrum., *Clinics in dermatology* **22 2**: 113–20. 102

Rajini, S., Nesamani, S. L., Abirami, P., Anuradha, K., Soundharyadevi, R. and Marappan, R. (2022). Dermatitis diagnosis – modeling and analysis using machine learning, *2022 2nd International Conference on Innovative Sustainable Computational Technologies (CISCT)* pp. 1–5. 39

ram asish Madiraju, S., Jakka, M. S., Aishwarya, R. and R, Y. (2022). Categorization of integumentary system disorders using deep learning, *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)* pp. 1226–1231. 35

Ramachandran, S., Fryer, A., Smith, A. G., Lear, J., Bowers, B., Griffiths, C., Jones, P. W. and Strange, R. C. (2000). Basal cell carcinoma, *Cancer* **89**. 100

Rasi, D., Antobennet, M., Renjith, P., Arun, M. and Vanathi, D. (2023). Yolo based deep learning model for segmenting the color images, *International Journal of Electrical and Electronics Research* . 13

Raza, M. A. A., Liaqat, M. S. and Shoaib, M. (2019a). A fuzzy expert system design for diagnosis of skin diseases. 29

Raza, M. A. A., Liaqat, M. S. and Shoaib, M. (2019b). A fuzzy expert system design for diagnosis of skin diseases, *2019 2nd International Conference on Advancements in Computational Sciences (ICACS)* pp. 1–7.
URL: *https://api.semanticscholar.org/CorpusID:116863310* 30

Sengupta, S., Mittal, N. and Modi, M. (2020). Contrast enhancement for color dermascopy images using equalization based on luminosity, *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)* pp. 845–848. 37

Sg, M., Ak, T., St, A., Sv, S. and Mj, O. (2017). Role of artificial intelligence in health care, *BioChemistry: An Indian Journal* **11**.
URL: *https://api.semanticscholar.org/CorpusID:56519291* 1

Sharma, A., Kroumpouzos, G., Kassir, M., Galadari, H., Goren, A., Grabbe, S. and Goldust, M. (2022). Rosacea management: A comprehensive review, *Journal of Cosmetic Dermatology* **21**: 1895 – 1904. 103

Shelhamer, E., Long, J. and Darrell, T. (2017). Fully convolutional networks for semantic segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**: 640–651. 13

Shorten, C. and Khoshgoftaar, T. (2019). A survey on image data augmentation for deep learning, *Journal of Big Data* **6**: 1–48. 17

Sinthura, S. S., Sharon, K. R., Bhavani, G., Mounika, L. and Joshika, B. (2020). Advanced skin diseases diagnosis leveraging image processing, *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)* pp. 440–444.
URL: *https://api.semanticscholar.org/CorpusID:221086885* 31

Streit, M. (2001). Contact dermatitis: clinics and pathology, *Acta Odontologica Scandinavica* **59**: 309 – 314. 99

Su, Z., Li, W., Ma, Z. and Gao, R. (2021). An improved u-net method for the semantic segmentation of remote sensing images, *Applied Intelligence* **52**: 3276–3288. 13

Tramontana, M., Hansel, K., Bianchi, L., Sensini, C., Malatesta, N. and Stingeni, L. (2023). Advancing the understanding of allergic contact dermatitis: from pathophysiology to novel therapeutic approaches, *Frontiers in Medicine* **10**. 99

Tuder, R., Young, R., Karasek, M. and Bensch, K. (1987). Adult cutaneous hemangiomas are composed of nonreplicating endothelial cells., *The Journal of investigative dermatology* **89 6**: 594–7. 101

Vinay, K., Sawatkar, G., Saikia, U. and Dogra, S. (2017). Dermatoscopic features of xanthoma disseminatum, *International Journal of Dermatology* **56**. 104

Vincent, L. and Jayasingh, J. R. (2022). Comparison of psoriasis disease detection and classification through various image processing techniques-a review, *2022 6th International Conference on Devices, Circuits and Systems (ICDCS)* pp. 122–124. 36

Wang, Y., Huang, G., Song, S., Pan, X., Xia, Y. and Wu, C. (2020). Regularizing deep networks with semantic data augmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**: 3733–3748. 99

Weidinger, S., Beck, L., Bieber, T., Kabashima, K. and Irvine, A. (2018). Atopic dermatitis, *Nature Reviews Disease Primers* **4**: 1–20. 100

Wijesinghe, L., Kulasekera, D. and Ilmini, W. (2019). An intelligent approach to segmentation and classification of common skin diseases in sri lanka, *2019 National Information Technology Conference (NITC)* pp. 47–52.
URL: *https://api.semanticscholar.org/CorpusID:219690397* 33

Xu, Z., Zhang, W., Zhang, T. and Li, J. (2020). Hrcnet: High-resolution context extraction network for semantic segmentation of remote sensing images, *Remote. Sens.* **13**: 71. 13

Yang, C. and Guo, H. (2022). A method of image semantic segmentation based on pspnet, *Mathematical Problems in Engineering* . 13

Yu, C., Gao, C., Wang, J., Yu, G., Shen, C. and Sang, N. (2020). Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation, *International Journal of Computer Vision* **129**: 3051 – 3068. 13

Yu, L., Zeng, Z., Liu, A., Xie, X., Wang, H., Xu, F. and Hong, W. (2022). A lightweight complex-valued deeplabv3+ for semantic segmentation of polsar image, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **15**: 930–943. 13

Zhang, Y., Chu, J., Leng, L. and Miao, J. (2020). Mask-refined r-cnn: A network for refining object details in instance segmentation, *Sensors (Basel, Switzerland)* **20**. 13

Zheng, D., He, X. and Jing, J. (2023). Overview of artificial intelligence in breast cancer medical imaging, *Journal of Clinical Medicine* **12**. 1