SUPERVISOR'S RECOMMENDATION

I hereby recommend that the report prepared under my supervision by Amisha Basnet (TU Exam Roll No. 28903/078), Saisa Koirala (TU Exam Roll No. 28932/078), Sandesh Khatiwada (TU Exam Roll No. 28936/078**)** entitled **"AI FOR SKIN DISEASE DETECTION"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for evaluation.

…………………………….

Supervisor

Pawan Niroula

Lecturer

Orchid International College

Bijayachowk, Gaushala

CERTIFICATE OF APPROVAL

This is to certify that this project prepared by Amisha Basnet, Saisa Koirala, and Sandesh Khatiwada entitled **"AI FOR SKIN DISEASE DETECTION"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in scope and quality as a project for the required degree.

Signature of Supervisor …………………………… Mr. Pawan Niroula Lecturer Supervisor Orchid International College Bijayachowk, Gaushala	**Signature of HOD Coordinator** …………………………… Er. Dhiraj Kumar Jha Head of Department, Department of IT, Orchid International College Bijayachowk, Gaushala
Signature of Internal Examiner …………………………… Hemant Mehta Fulltime faculty Orchid International College Bijayachowk, Gaushala	**Signature of External Examiner** ……………………………

ACKNOWLEDGEMENT

It is truly a privilege to express our profound gratitude to all those who supported us. We want to express our appreciation to **Er. Dhiraj Kumar Jha** and **Pawan Niroula** for his invaluable assistance in completing this project. His expert guidance and willingness to dedicate time from his busy schedule for project reviews have been instrumental. The continuous inspiration we received from him played a pivotal role in achieving our goals.

Additionally, we would like to extend our sincere thanks to the faculty of Orchid International College, Department of Computer Science and Information Technology, for providing us with this opportunity. Our gratitude also goes to our friends and colleagues for their selfless efforts in aiding us to successfully complete the project.

Amisha Basnet (28903/078)
Saisa Koirala (28932/078)
Sandesh Khatiwada (28936/078)

ABSTRACT

Skin diseases affect millions worldwide, and timely diagnosis is critical for effective treatment. Traditional diagnostic methods rely on manual visual examination by dermatologists, which can be time-consuming and prone to errors. With the increasing availability of digital skin images, automated detection using artificial intelligence offers a faster and more reliable alternative. In this project, "AI for Skin Disease Detection," a system was developed using Convolutional Neural Networks (CNN), MobileNet, and DenseNet121 to classify images into nine common skin disease categories, including Actinic Keratosis, Melanoma, and Atopic Dermatitis. Among the models, DenseNet121 achieved the highest accuracy and was selected as the primary deployed model. The system preprocesses images by resizing and normalizing them, then performs feature extraction and classification using the trained deep learning models. The dataset was sourced from Kaggle and augmented to improve model generalization. DenseNet121 achieved a final test accuracy of 82%, outperforming CNN and MobileNet models. Confusion matrices and accuracy plots were used to evaluate model performance, confirming reliable classification across all disease categories. This AI-based approach provides a practical and efficient solution for automated skin disease detection, supporting dermatologists in diagnosis and improving patient care.

Keywords: Skin Disease, Deep Learning, CNN, MobileNet, DenseNet121, Image Classification, DermNet

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 - INTRODUCTION

1.1. Introduction

Skin diseases are not unusual health troubles affecting people of every age, starting from mild conditions like pimples and eczema to excessive, life-threatening ailments along with melanoma. Accurate and early prognosis of these conditions is crucial, however traditional techniques depend heavily on dermatologists and are liable to human blunders and delays, mainly in regions with confined healthcare get admission to. Skin disorder detection the use of Artificial Intelligence (AI) leverages deep getting to know and image classification techniques to analyze pores and skin lesion snap shots and mechanically identify sickness kinds. This approach minimizes human bias, quickens prognosis, and can help dermatologists in making knowledgeable decisions.

The AI for Skin Disease Detection venture implements this idea via growing a deep studying machine educated on 900 photos throughout nine skin disorder classes. The system makes use of a custom CNN, MobileNetV1, and DenseNet121, with DenseNet121 more suitable and best-tuned for the dataset. The mission covers all preprocessing, function extraction, and model education steps, which includes convolution, activation, pooling, absolutely linked layers, and softmax output, providing a sensible, code-pushed solution for reliable and efficient skin disease classification.

1.2. Problem Statement

Although pores and skin diseases are commonplace, early and correct diagnosis stays a good-sized difficulty, specifically in regions wherein experts lack suitable get entry to. Delayed or wrong analysis may additionally lead to critical headaches. Traditional visible diagnosis methods are subjective and unreliable. The venture addresses these issues by means of proposing a deep mastering-based tool to diagnose skin illnesses from dermoscopic pictures for rapid, correct, and low-cost diagnosis.

1.3. Objectives

- To build a deep learning-based classification model for skin diseases using image data.
- To build and train deep learning models (CNN, MobileNet, DenseNet121) for accurate skin disease classification.
- To compare the performance of these models using evaluation metrics such as accuracy, loss, and confusion matrix.
- To deploy the model through a simple web interface for user interaction.

1.4. Scope and Limitations

This venture focuses completely on classifying skin sicknesses using static medical photo facts thru deep learning fashions. It goals to offer early detection and cognizance but excludes real-time diagnostics, external tool integration, and live affected person monitoring. The machine functions as a standalone assistive tool and isn't always a licensed medical diagnostic product. Its effectiveness relies upon on clean photograph input and the exceptional and diversity of the dataset. It is trained in small datasets which may not accurately predict real word cases perfectly along with the limitation of classification of only 9 diseases.

1.5. Development Methodology

The mission follows an incremental delivery method, dividing improvement into a series of small, attainable iterations. Each generation builds upon the preceding one by using including new functions, permitting non-stop development and versatility. This method suits projects with evolving requirements or high complexity by way of allowing normal remarks and adjustments at some stage in the process. For this task, the preliminary iteration set up core capabilities which includes internet scraping for information collection and fundamental facts processing. Later iterations more advantageous category accuracy and summarization techniques. Each degree become thoroughly examined to make sure reliability before transferring forward. Overall, this technique enabled a bendy, adaptive improvement procedure that efficaciously added a functional and robust system.

The record is dependent into six chapters:

1. Introduction - protecting undertaking evaluate, hassle assertion, targets,
2. scope, constraints, and development technique.
3. Background Study - reviewing relevant theories, ideas, and comparable works through others.
4. Analysis - including device necessities and feasibility assessment.
5. System Design - detailing the overall layout and shape.
6. Implementation and Testing - describing system used and trying out strategies.
7. Conclusion and Future Recommendations - summarizing findings and suggesting upgrades.

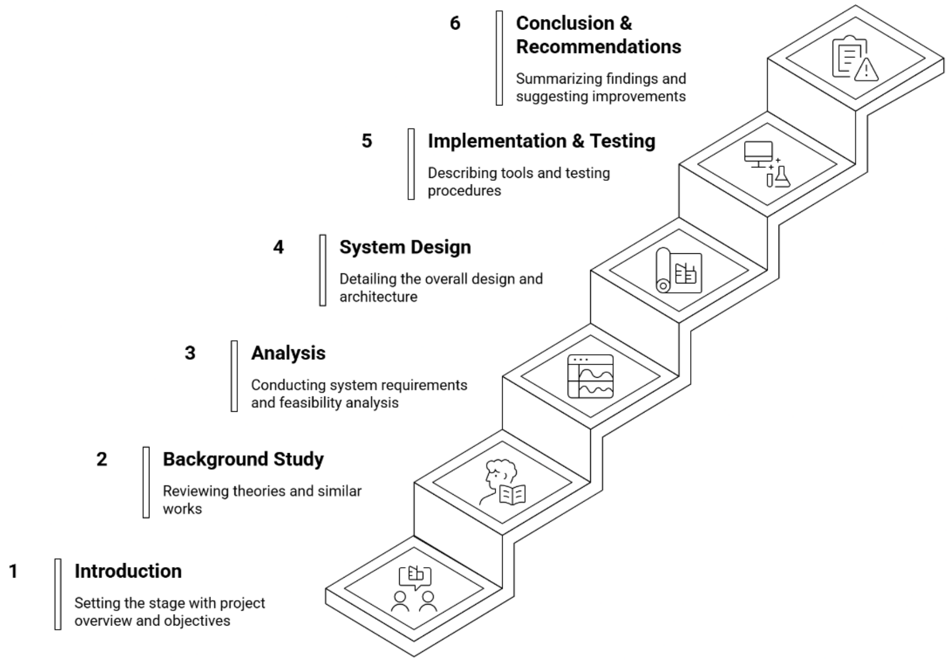


Figure 1.1: Report Organization

CHAPTER 2 - BACKGROUND AND LITERATURE REVIEW

2.1. Background Study

Skin diseases, in particular cancer, retain to pose serious threats to public fitness, specifically in international locations which includes New Zealand, Australia, and the USA, where prevalence rates have substantially risen in latest decades. Melanoma, which arises due to the uncontrolled boom of melanocytes, is one of the deadliest varieties of pores and skin cancer. Early detection of cancer extensively increases survival fees, but accurate prognosis stays tough due to visual similarities among benign and malignant lesions and overlapping features with regular pores and skin.

Traditionally, dermatologists rely on dermoscopic pics and manual evaluation to diagnose skin lesions. However, this procedure needs significant clinical knowledge and is time-eating. In response, computer-aided prognosis systems powered with the aid of Artificial Intelligence (AI) and Machine Learning (ML) have emerged as a promising opportunity to help in early and correct detection.

The advancement of Deep Learning (DL), in particular Convolutional Neural Networks (CNNs), has revolutionized image classification responsibilities. CNNs have shown exceptional fulfillment in recognizing complicated styles in scientific imaging, together with pores and pores and skin lesion category. This method leverages hierarchical characteristic extraction, mimicking the human visual cortex, to successfully look at and distinguish among diverse lesion sorts.

In the research performed through Viswanatha Reddy Allugunti, a deep analyzing-primarily based totally CNN version is developed to categorise varieties of cancer - which consist of lesion maligna, superficial spreading, and nodular cancer. The model achieves excessive classification accuracy (82 %) and outperforms traditional ML algorithms which include Decision Trees, Random Forests, and Gradient Boosted Trees. The studies underscores the potential of CNNs to feature a powerful diagnostic help device, helping clinicians in early cancer detection and treatment making plans.

2.2. Literature Review

Numerous researchers have explored automatic strategies for pores and skin lesion assessment, combining classical device mastering techniques with cutting-edge-day-day deep studying architectures to decorate diagnostic overall performance. These strategies awareness on lesion segmentation, elegance, and longitudinal tracking to aid early most cancers detection and other pores and skin illness identity.

Bi et al. [2] proposed a deep residual community for automated skin lesion evaluation, combining massive-scale dermoscopic datasets with residual connections to beautify gradient float and decrease overfitting. The version correctly segmented and labeled lesions, displaying high overall performance on ISIC datasets, it is important for real-global clinical adoption.

Xie et al. [4] brought a high-decision convolutional neural network for lesion segmentation, focusing on accurate boundary detection. Their technique incorporated multi-scale function maps and refinement layers to deal with versions in lesion size, form, and color. Similarly, Yuan and Lo [3] advanced better convolutional-deconvolutional networks that advanced dermoscopic photograph segmentation, permitting extra unique lesion delineation and helping downstream category tasks.

Li and Shen [6] developed a deep learning network for melanoma detection, demonstrating how convolutional neural networks (CNNs) can automatically extract discriminative features from dermoscopic images. Their model achieved high classification accuracy by training on a large number of clinical images and incorporating data augmentation to reduce overfitting.

Rashmi Patil and Bellary [7, 11] focused on stage-wise melanoma classification. They developed specialized loss functions and transfer learning strategies to improve sensitivity and specificity in melanoma detection. Their approach highlighted the importance of stage-based analysis and gradient-based similarity metrics, which enhanced classification performance and clinical interpretability.

Korotkov et al. [8] proposed a lesion matching algorithm for full-body imaging, facilitating longitudinal monitoring of lesion progression. This approach allowed early detection of changes in existing lesions and identification of new suspicious lesions, supporting comprehensive patient evaluation over time.

Kassem et al. [9] carried out switch learning the use of GoogleNet on the ISIC 2019 dataset to categorise 8 particular varieties of pores and skin lesions. Their technique mitigated elegance imbalance by means of the use of best-tuning pretrained weights and normalizing class distributions, engaging in a precision of 94.92% and outperforming different popular architectures like VGG19 and ResNet50. This demonstrates the effectiveness of leveraging pretrained deep networks on smaller, area-precise datasets.

Ichim and Popescu [10] introduced a dual-stage classifier integrating conditional generative adversarial networks (GANs) and multiple neural networks. By incorporating Total Dermoscopy Score (TDS) into the final classification stage, their system achieved 97.5% accuracy, demonstrating adaptability to different datasets and effective feature fusion from multiple models.

Collectively, those research illustrate a clean trend: the shift from shallow architectures and handcrafted capabilities to deep, residual, and transfer-mastering-primarily based fashions. Combining segmentation and class, regularly with multi-level or ensemble architectures, significantly improves the overall performance of automatic pores and skin sickness detection systems. These improvements not best increase diagnostic accuracy however also offer scalable answers for large-scale scientific and studies programs.

CHAPTER 3 - SYSTEM ANALYSIS

3.1. Requirement Analysis

Requirement evaluation on this mission makes a specialty of identifying what is wanted to build the AI-based totally Skin Disease Detection System. It includes defining the undertaking's desires, the dataset of skin ailment pictures, the deep learning models for use (CNN, MobileNetV1, DenseNet121), and the features inclusive of preprocessing, education, evaluation, and deployment via a web interface. Both purposeful needs (like picture classification and prediction) and non-purposeful wishes (together with usability and overall performance) are taken into consideration to make certain the machine works effectively.

3.1.1. Functional Requirements

The functional requirements of the skin disease detection system include:

3. User Authentication: Secure login and access management for users (patients and medical personnel).
4. Image Upload: Functionality to upload images of skin lesions from local storage or via webcam.
5. Disease Classification: Automated prediction of skin disease using a trained AI model (CNN, MobileNet, DenseNet121).
6. Probability Display: Showing top predicted conditions along with confidence scores.
7. Visualization: Interactive charts and tables for prediction probabilities and class ranking.
8. Past Records: Storing and retrieving past uploads and predictions for user reference.
9. External References: Providing links to reliable information sources for each predicted disease.

The following is the use case diagram that describes different functionalities of the system and interaction between actors:

Figure 3.1: Use Case Diagram for AI For Skin Disease Detection

The diagram models how a patient interacts with the Skin Disease Detection System. The affected person can log in or register, which incorporates validating credentials. They then add a pores and skin image; the system includes an AI inference step to analyze it. The affected person perspectives the prediction result, that can optionally increase to getting care hints (guidance primarily based at the final results). The affected person might also view past history/reviews, which includes fetching stored information from the database. Solid arrows display moves the affected person initiates. The «include» relationships mark required sub-steps (validation, inference, statistics fetch), while «enlarge» marks a non-obligatory characteristic (care inspiration).

3.1.2. Use-Case Description

Table 3.1: Use Case Description for Register

Use case identifier	UC-01
Use Case Name	Register
Primary Actor	User (Patient)
Secondary Actor	None
Description	Registers a user into the system for secure access.
Pre-condition	User has not registered in the system.
Success-scenario	User account is created and stored in the database; user can login successfully.
Failure-scenario	User is redirected to the registration page again if registration fails.

Table 3.2: Use Case Description for Login

Use case identifier	UC-02
Use Case Name	Login
Primary Actor	User (Patient)
Secondary Actor	None
Description	Logs the user into the system.
Pre-condition	User must be registered.
Success-scenario	User is then redirected to the main home page/ dashboard page.

Failure-scenario	User is hence redirected to login page again.

Table 3.3: Use Case Description for Upload Skin Image

Use case identifier	UC03
Use Case Name	Upload Skin Image
Primary Actor	User (Patient)
Secondary Actor	None
Description	User uploads an image of the skin lesion for AI-assisted analysis.
Pre-condition	User is logged in.
Success-scenario	Image is successfully uploaded and stored in the database.
Failure-scenario	User must re-upload image if file format is unsupported.

Table 3.4: Use Case Description for View Diagnosis Results

Use case identifier	UC04
Use Case Name	View Diagnosis Results
Primary Actor	User (Patient)
Secondary Actor	None
Description	User can view predicted skin disease results after image analysis.
Pre-condition	Image has been successfully uploaded.

Success-scenario	Predicted results with probabilities are displayed.
Failure-scenario	Results fail to generate due to system error.

Table 3.5: Use Case Description for Save Diagnosis Report

Use case identifier	UC-05
Use Case Name	Save Diagnosis Report
Primary Actor	User (Patient)
Secondary Actor	None
Description	User can save the diagnosis report for future reference.
Pre-condition	Diagnosis results are available.
Success-scenario	Report is stored in the user’s account.
Failure-scenario	Report fails to save due to database error.

Table 3.6: Use Case Description for View Past Records

Use case identifier	UC-06
Use Case Name	View Past Records
Primary Actor	User (Patient)
Secondary Actor	None
Description	User can view previously uploaded images and diagnosis results.
Pre-condition	User is logged in.

Success-scenario	All past records are displayed.
Failurescenario	No past records found or session expired.

Table 3.7: Use Case Description for Logout

Use case identifier	UC-07
Use Case Name	Logout
Primary Actor	User (Patient)
Secondary Actor	None
Description	User logs out from the system.
Pre-condition	User is logged in.
Success-scenario	User is successfully logged out.
Failure-scenario	Logout fails due to session error.

3.1.3. Non-Functional Requirements

Non-purposeful requirements describe characteristics or attributes of a machine that don't relate to unique behaviors or capabilities but alternatively specify how the system should carry out in terms of characteristics such as performance, usability, security, and reliability. Here are a few non-practical requirements for “AI for Skin Disease Detection”:

1. Security: Secure login with encrypted credential storage.
2. Usability: Simple, intuitive interface for photo upload and end result viewing.
3. Maintainability: Modular codebase for clean updates and retraining.
4. Reliability: Consistent and accurate outputs with information backup mechanisms.
5. Scalability: Ability to handle larger datasets and more diseases in the future.

6. Portability: Deployable across local servers, cloud platforms, and medical institution systems.'

3.2. Feasibility Analysis

i. **Technical**

On the front-quit aspect, this system runs on a Flask-primarily based internet platform, at the same time as at the again-end facet, deep gaining knowledge of models along with CNN, MobileNet, and DenseNet121 are used with Python. The required development gear (Python, TensorFlow/Keras, Flask, HTML/CSS/JS) are open-source and effortlessly available, making the device technically viable. The running system required is Microsoft Windows or better, which is broadly to be had, making sure compatibility.

ii. Operational

The operational feasibility of the AI for Skin Disease Detection machine is classified to make certain it meets the described goals and presents real-world usability. The system offers convenient web-primarily based get entry to, allowing customers to upload pores and skin snap shots and receive predictions thru a consumer-pleasant interface. This makes it practical for each scientific practitioners and widespread users.

iii. **Schedule**

The venture became divided into a couple of levels, with group contributors accountable for dataset coaching, model education, system integration, and deployment. The challenge turned into planned to be finished inside forty–50 days, which was possible with recognize to time. The time table of the mission is represented through the Work Breakdown Structure (WBS) and Gantt Chart.

Table 3.8: Work Breakdown Structure (WBS) of AI for Skin Disease Detection

	ⓘ	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1			Project Planning	3 days	Wed 7/2/25	Fri 7/4/25	
2			Dataset Collection	4 days	Mon 7/7/25	Thu 7/10/25	1
3			Dataset Preprocessing & Augmentation	6 days	Fri 7/11/25	Fri 7/18/25	2
4			Baseline CNN Model Development	5 days	Mon 7/21/25	Fri 7/25/25	3
5			MobileNet Model Development	5 days	Mon 7/21/25	Fri 7/25/25	3
6			DenseNet121 Model Development	5 days	Mon 7/21/25	Fri 7/25/25	3
7			Model Training & Evaluation	7 days	Mon 7/28/25	Tue 8/5/25	6,5,4
8			Model Optimization (Hyperparameters, Class Balance)	5 days	Wed 8/6/25	Tue 8/12/25	7
9			Flask Backend Development	5 days	Wed 8/6/25	Tue 8/12/25	7
10			Frontend Development (HTML/CSS/JS)	5 days	Wed 8/13/25	Tue 8/19/25	9
11			Integration of Model with Web App	4 days	Wed 8/20/25	Mon 8/25/25	8,10
12			Testing (Unit + End-to-End)	5 days	Tue 8/26/25	Mon 9/1/25	11
13			Documentation Preparation	7 days	Tue 9/2/25	Wed 9/10/25	12

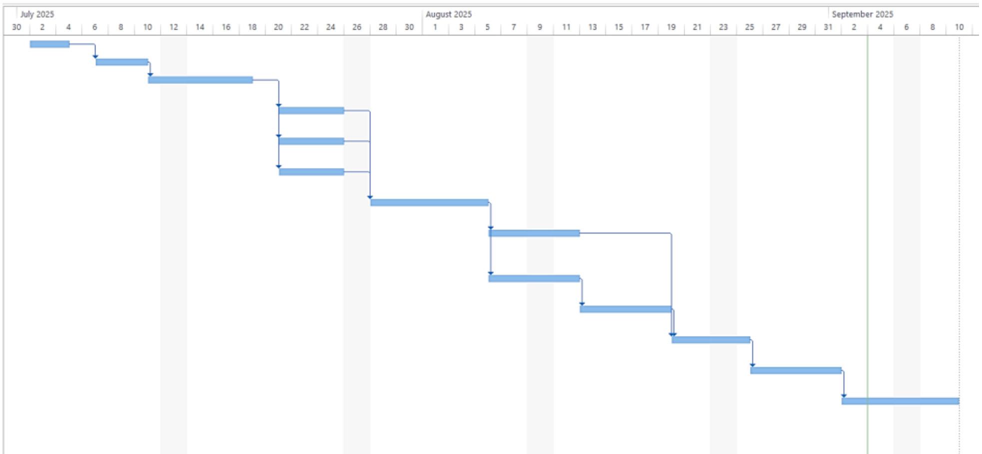


Figure 3.2: Gantt Chart of AI for Skin Disease Detection

3.3. Object Modeling using Class Diagram

The class diagram for this AI-Based Skin Disease Detection System carries classes together with User, Image, Prediction, Model, and Disease Information. This diagram serves as a blueprint for the system's object modeling, outlining the important thing instructions, their attributes, functionalities and relationships.

i. User: Represents patients or scientific personnel who can check in, login, and think about beyond data.
ii. Image: Represents uploaded images of skin lesions, storing metadata like add date, image path, and related consumer.
iii. Prediction: Stores the expected sickness classes, self-belief rankings, and hyperlinks to outside statistics sources.
iv. Model: Represents the AI models used for type (CNN, MobileNet, DenseNet121), which includes model kind and version.
v. Disease Info: Provides additional reference information about illnesses for instructional functions.

The class diagram for the AI-Based Skin Disease Detection System is shown below:

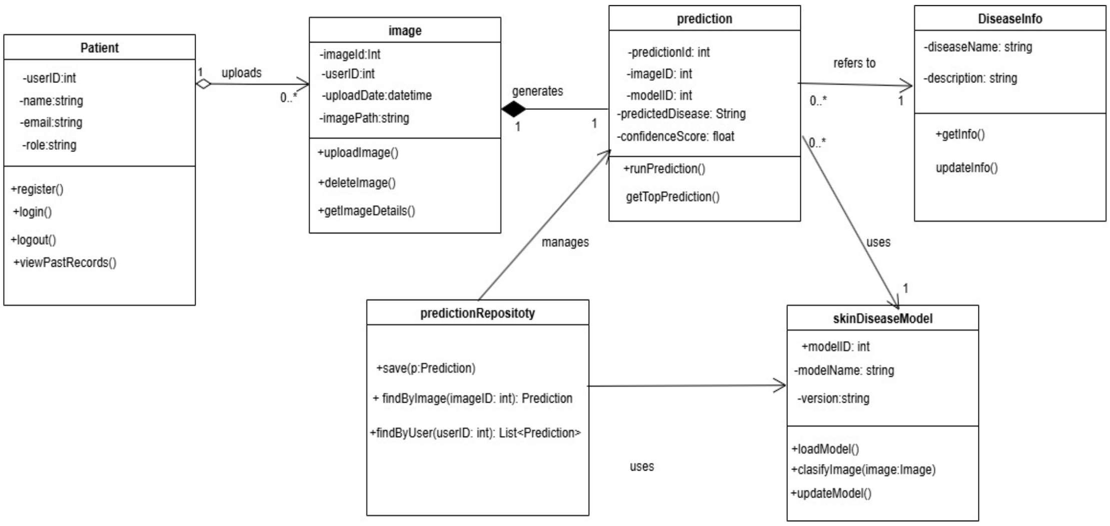


Figure 3.3: Class Diagram of AI for Skin Disease Detection

The diagram 3.3 fashions instructions and class diagram for the Skin Disease Detection System. A Patient can sign in/login/logout and viewPastRecords. The Patient uploads 0..* Image data (imageId, userId, uploadDate, imagePath) with add/delete/get info.

Each Image composes one-or-more Prediction items (a prediction relies upon on its picture). A Prediction (predictionId, imageId, modelId, predictedDisease, confidenceScore) runs inference and returns the pinnacle result, makes use of SkinDiseaseModel to classify, and may refer to 0..* DiseaseInfo entries (diseaseName, description). SkinDiseaseModel (modelId, modelName, model) masses, classifies, and updates the model. PredictionRepository persists and queries predictions (keep, findByImage, findByUser). Dependencies show prediction makes use of the model; repository manages predictions.

CHAPTER 4 - SYSTEM DESIGN

4.1. Design

The system layout for the AI for Skin Disease Detection System involves defining its architecture, components, and behavior. This system targets to create a blueprint that courses developers in correctly enforcing the system to fulfill assignment targets of skin disease detection, prediction visualization, and user management.

4.1.1. Activity Diagram

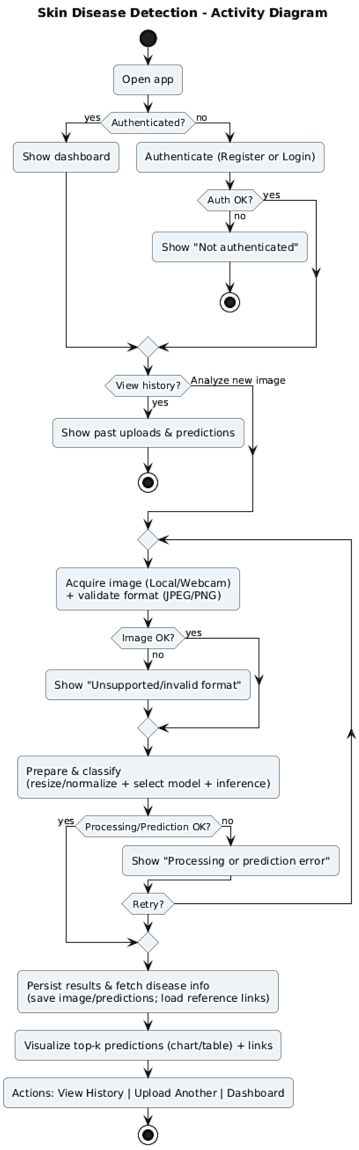


Figure 4.1: Activity Diagram of AI for Skin Disease Detection

The activity diagram 4.2 suggests the quit-to-give up app drift. On release, authentication is checked: authenticated users attain the dashboard; others should check in/login, and screw ups stop the glide. From the dashboard, the user can view records or examine a new photograph. For analysis, the system acquires a photo (local/webcam), validates format (JPEG/PNG), and rejects invalid inputs. Valid images are preprocessed (resize/normalize), a version is selected, and inference runs. If processing fails, an error is shown with a retry choice. On success, predictions and image are stored, disease data is fetched, and top-ok outcomes are visualized, with actions to view records, upload any other, or return to the dashboard.

4.1.2. Component Diagram

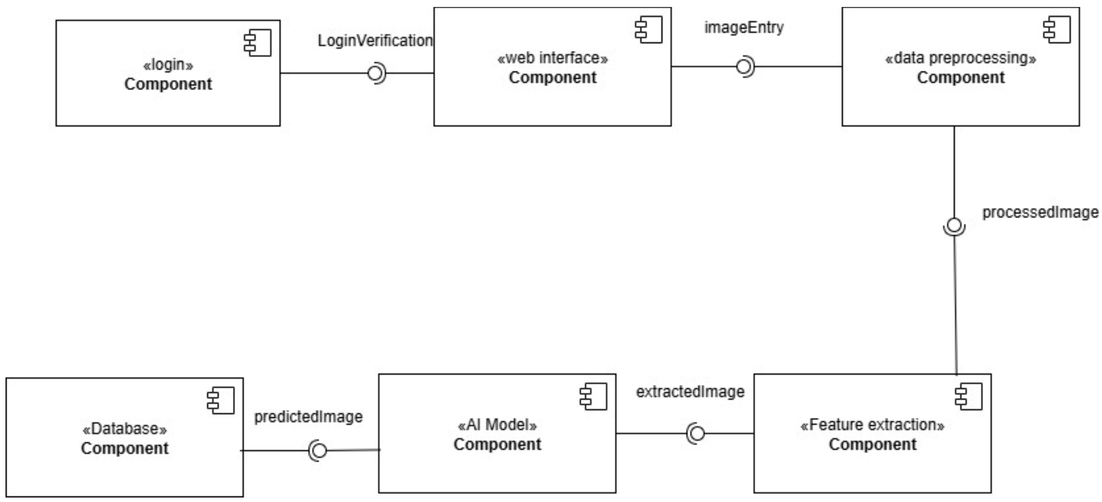


Figure 4.2: Component Diagram of AI for Skin Disease Detection

The component diagram 4.3 shows a modular pipeline. The Login factor verifies customers through the Web Interface (ball-and-socket suggests required/provided interfaces). After authentication, the Web Interface accepts an imageEntry and sends it to Data Preprocessing, which cleans/normalizes it and outputs processedImage. Feature Extraction gets this and produces extractedImage (function vectors). The AI Model consumes the ones features to deduce the disease, returning predictedImage/prediction statistics. The Database issue persists predictions and allows later retrieval. Overall, the diagram separates duties- UI/auth, preprocessing, characteristic extraction, model inference, and storage- related thru express interfaces for clear boundaries, replaceability, and less difficult protection or scaling.

4.1.3. Deployment Diagram

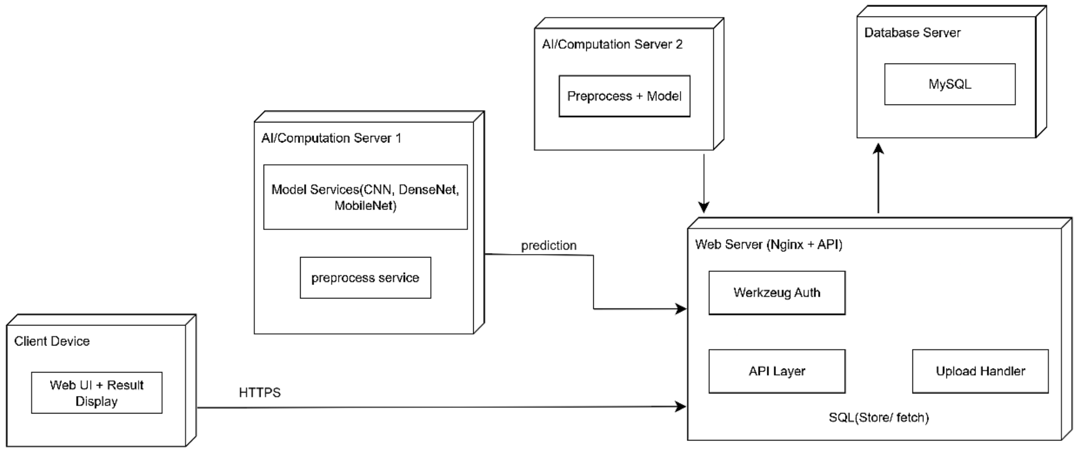


Figure 4.3: Deployment Diagram of AI for Skin Disease Detection

The diagram 4.4 shows a deployment layout for the skin-disease app. The Client Device hosts the web UI to upload images and view results. Requests hit the Web Server (Nginx reverse proxy + API), which handles authentication (Werkzeug), file uploads, and SQL reads/writes. Prediction jobs are sent to AI/Computation servers. Server 1 exposes model services (e.g., CNN, DenseNet, MobileNet) and a preprocessing service; Server 2 is an additional node with preprocess + model for scale or redundancy. The Database Server (MySQL) stores users, image metadata, and prediction records. Flow: client → web server → upload handler → preprocess → model inference → store/fetch in MySQL → result back to client.

4.1.4. Sequence Diagram

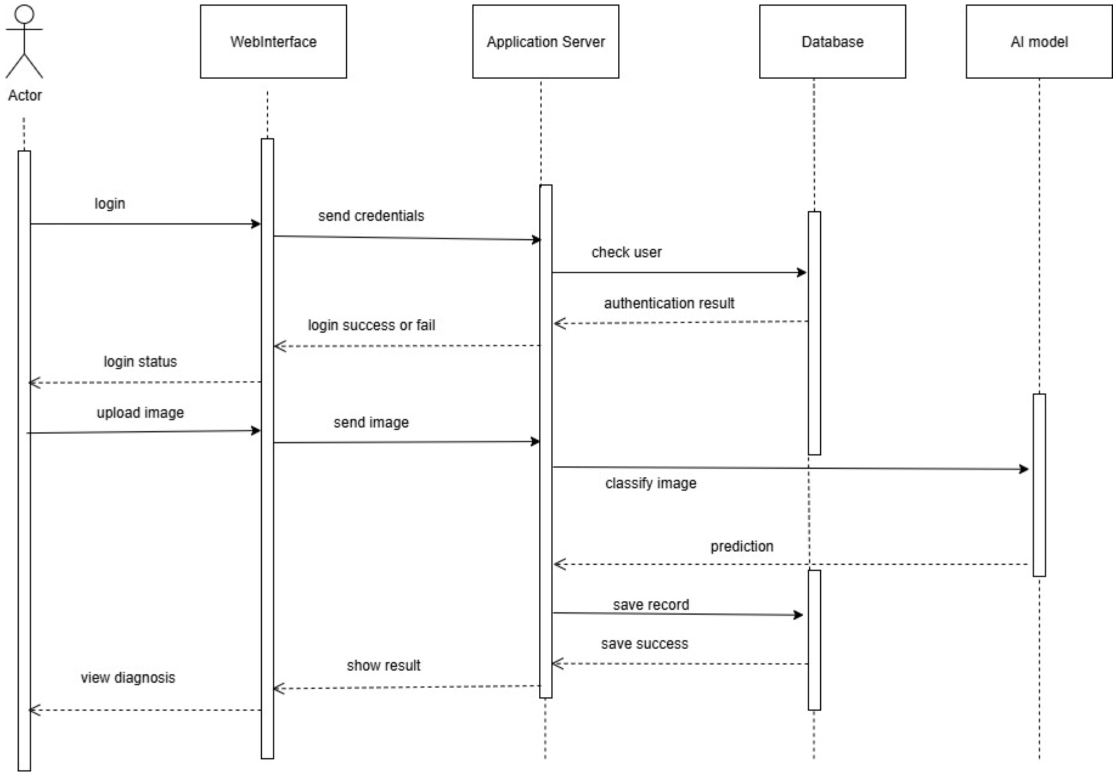


Figure 4.4: Sequence Diagram of AI for Skin Disease Detection

The sequence diagram 4.1 indicates end-to-end flow. The Actor logs in via the Web Interface, which sends credentials to the Application Server. The server checks the person inside the Database and returns an authentication end result to the Web UI. After fulfillment, the user uploads a skin photograph; the Web Interface forwards it to the Application Server. The server invokes the AI model to categorise the photograph and gets a prediction. It then saves the prediction record to the Database and gets a shop confirmation. Finally, the Application Server returns the result to the Web Interface, which displays the diagnosis to the consumer.

4.1.5. Refinement of Sequence Diagram

A refined sequence diagram provides more refined and better version of sequence diagram which shows the workflow in a project.

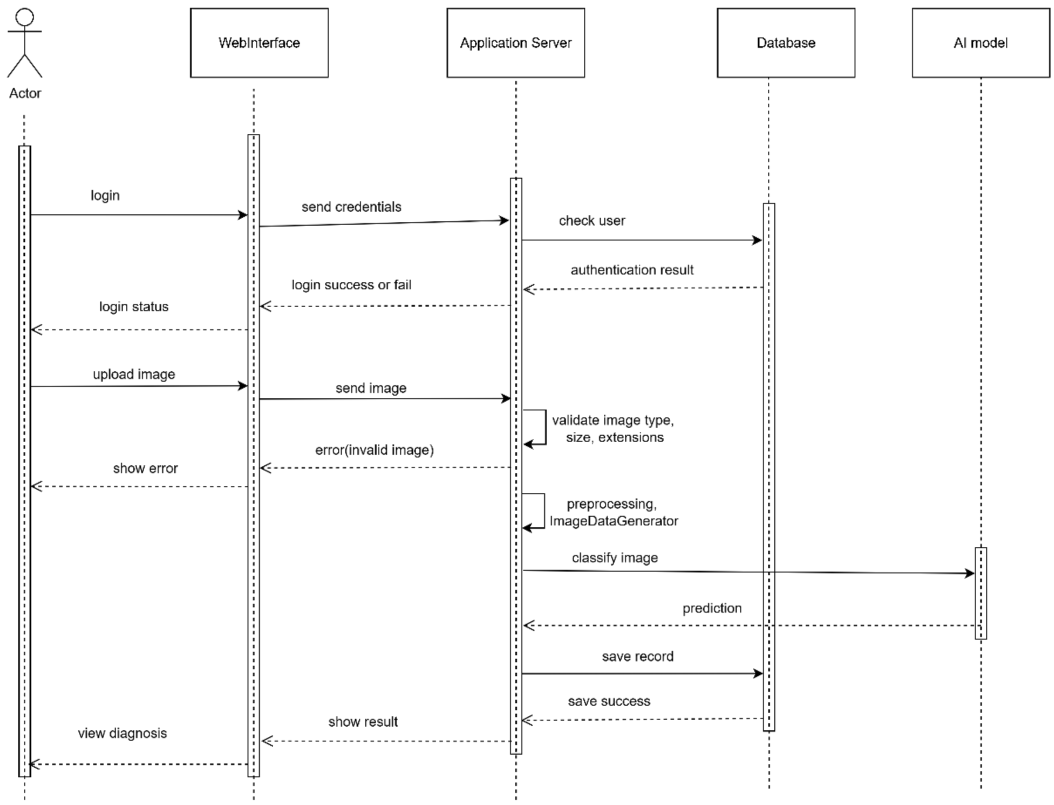


Figure 4.5: Refined Sequence Diagram of AI for Skin Disease Detection

The above refined sequence diagram 4.5 is same as that of sequence diagram but validates images and preprocesses and enhances it using ImageDataGenerator after login/authentication and uploading image stage is performed. This validation either shows the prediction by classifying image or shows error for image invalidity. Image after prediction is saved and success result is shown.

4.1.6. Model Architecture

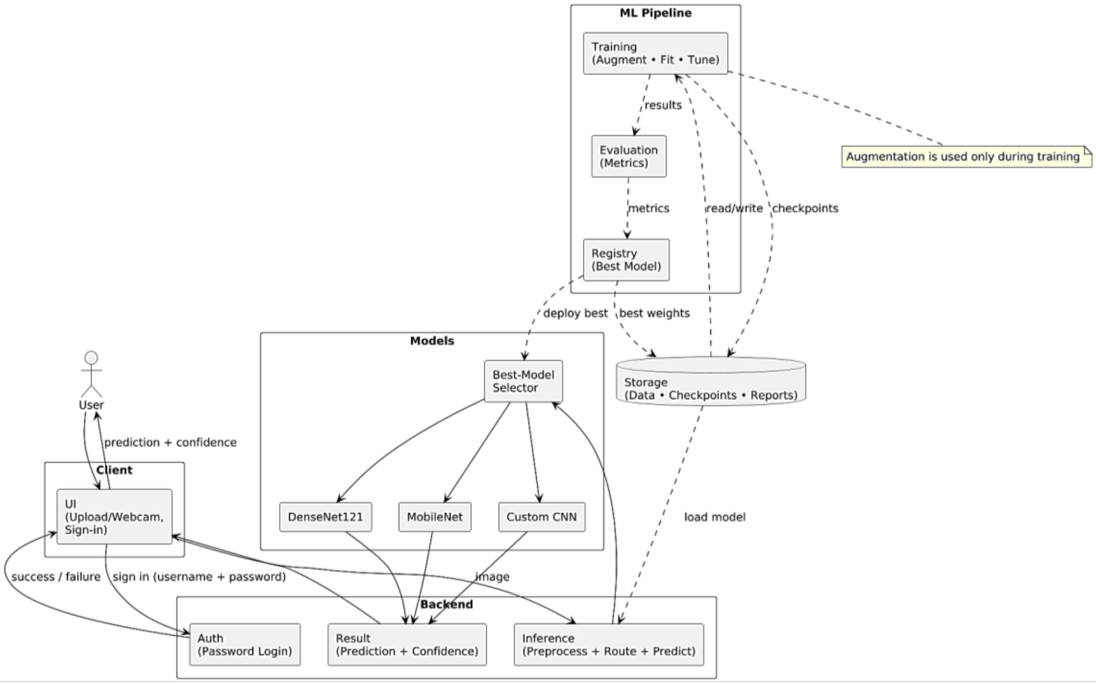


Figure 4.6: Model Architecture of AI for Skin Disease Detection

The diagram links product and ML lifecycle. A user signs in via the client UI and uploads an image. The backend handles Auth, then Inference: preprocess, route to a model (DenseNet121, MobileNet, or Custom CNN) via a Best-Model Selector, and return prediction + confidence. Separately, an ML Pipeline trains models with augmentation (training only), evaluates metrics, and registers the best model. Checkpoints, data, and reports are saved in Storage. The best weights are deployed back to production; inference loads the chosen model from Storage. Thus, training/evaluation continuously improve models, while the live system authenticates users, processes images, selects the current best model, and serves results.

4.2. Algorithm Details

4.2.1. Convolutional Neural Network (CNN)

For the baseline version, we implemented a custom Convolutional Neural Network (CNN) to categorise pores and skin sicknesses into 9 classes the usage of our dataset of 900 images. The architecture become constructed grade by grade, beginning from uncooked pixel inputs to very last opportunity predictions.

a. Input Layer:

Each skin image was resized to 224×224×3, where each pixel intensity was normalized between 0 and 1. Thus, the input can be represented as a 3D tensor:

$$X \in R^{224*224*3}$$

b. Convolution Layers:

Convolution was the first operation that was performed, which extracts features such as edges, textures and color variations of the diseases. For each filter W, the convolution layer is calculated as:

$$Z_{i,j,k} = \sum_{m=0}^{m-1} \sum_{n=0}^{n-1} \sum_{c=0}^{c-1} X_{\{i+m,j+n,c\}} \cdot W_{\{m,n,c,k\}} + b_k$$

where,

X= input image

W= learnable kernel of size

b_k= bias term for filer k,

Z = resulting feature map.

c. Activation Function:

In our implementation, we used convolutional layers with ReLU activations at the start to seize low-level features (edges, coloration gradients) and steadily greater complex patterns (pores and skin textures, lesion boundaries)

$$f(x) = \max(0, x)$$

This allowed our network to learn complex decision boundaries that are past linear transformations.

d. Pooling Layers

To lessen the spatial size and computational load, we implemented Max Pooling after convolutional blocks. For each region R:

$$P(i, j) = \max_{\{(m,n)\in R\}} Z(i + m, j + n)$$

This operation preserved the most essential capabilities (like sturdy edges or lesion patches) while decreasing noise and dimensionality.

e. Dropout Regularization

To save you overfitting on our small dataset, we brought Dropout layers, where random neurons are unnoticed at some point of training with probability p.

Mathematically:

$$h^{\{(l)\}} = r^{(l)} \odot f\left(W^{(l)} h^{(l-1)} + b^{(l)}\right)$$

where,

$r^{(l)} \sim$ Bernoulli(1-p)

$\odot$ = element-wise multiplication

This forced the community to analyze redundant, greater well known features instead of memorizing training samples.

f. Fully Connected Layers

After pulling down the characteristic maps right into a vector, we applied absolutely connected layers.

$$y = f(Wx + b)$$

Where x is the flattened enter, W and b are trainable weights and bias, and f is the ReLU activation.

This degree mixed extracted features (edges, textures, patterns) to study better-stage representations of each sickness.

g. Output Layer with Softmax

Finally, we applied a Dense layer with nine neurons (corresponding to the 9 skin disease classes) and a Softmax activation:

$$\hat{y} = \frac{e^{\{z_{\{i\}}\}}}{\sum_{\{j=1\}}^{\{9\}} e^{\{z_j\}}}$$

This ensured that the output vector sums to one, making it interpretable as class chances.

4.2.2. MobileNetV1 (Transfer Learning)

For advanced function extraction and performance, we applied MobileNetV1 as a switch learning version for classifying 9 skin Diseases. The community leverages depthwise separable convolutions, permitting reduced computation at the same time as retaining accuracy.

a. Input Layer

Lesion of skin images were resized and normalized to:

$$X \in R^{\{224*224*3\}}, X_{norm} = \frac{X}{255}$$

Each pixel intensity is scaled down between 0 and 1 for faster computation during training process.

b. Depthwise Separable Convolution

Unlike preferred convolution, MobileNetV1 makes use of depthwise separable convolution, which splits convolution into two steps:

i. Depthwise Convolution – Applies a single clear out according to enter channel:

$$Z_{\{i,j,c\}} = \sum_{\{m,n\}} X_{\{i+m,j+n,c\}} \cdot W_{\{m,n,c\}}$$

ii. Pointwise Convolution – Applies 1×1 convolution throughout channels to mix features:

$$Y_{\{i,j,k\}} = \sum_{\{c\}} Z_{\{i,j,c\}} \cdot P_{\{c,k\}} + b_k$$

This reduces the range of parameters from $K \cdot K \cdot C_{in} \cdot C_{out}$ (standard conv), extensively improving computational performance, mainly important for excessive-decision skin lesion pictures.

c. Pre-trained wights (Transfer Learning)

We initialized MobileNetV1 with ImageNet weights. This provides a robust baseline of wellknown visible functions (edges, textures, shapes) which reduces education time and improves convergence on our pores and skin disorder dataset.

Let the initial weights be $W_{pretrained}$. During forward propagation:

$$Z = f(X * W_{pretrained} + b)$$

in which $*$ denotes depthwise separable convolution.

d. Fine-Tuning for Skin Disease Classification

To adapt MobileNet to our dataset, we unfroze the last few layers for training. The ahead skip and weight replace for those layers is computed as usual with gradient descent:

$$W^{(t+1)} = W^{(t)} - \eta \frac{\partial L}{\partial W^{(t)}}$$

where,

η= learning rate,

L= categorical cross-entropy loss,

$$L = -\sum_{\{i=1\}}^{\{9\}} y_i \log(\hat{y}_i)$$

permitting the network to specialize in distinguishing 9 types of skin lesions.

e. Fully Connected Output Layer

After worldwide average pooling, the functions are fed into a Dense layer with 9 neurons, one in step with magnificence. Softmax activation converts logits zi into class probabilities:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{9} e^{\{z_j\}}}$$

This offers the anticipated probability for each pore and skin disorder.

f. Prediction and Ranking

The model outputs a vector $\hat{y} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_9]$. The anticipated magnificence is chosen as:

$$\text{Predicted Class} = \text{argmax}_i \hat{y}_i$$

Optionally, the pinnacle-okay chances can be said to indicate the maximum likely pores and skin situations, with their self-assurance ratings $\hat{y}_i \in [0,1]$.

4.2.3. DenseNet121 (Transfer Learning)

For the excellent-acting version in our project, we implemented DenseNet121 for classifying 9 sorts of skin diseases. DenseNet's architecture lets in function reuse and mitigates the vanishing gradient trouble, making it suitable for deep networks.

a. Input Layer

Skin disease images were reporccesd and normalized.

$$X \in R^{\{224*224*3\}}, X_{norm} = \frac{X}{255}$$

Each image was given to DenseNet for feature extraction.

b. Dense Blocks

For the excellent-acting version in our project, we implemented DenseNet121 for classifying 9 sorts of skin diseases. DenseBlock includes multiple layers where in each layer gets input from all previous layers. Let the output of layer l be x_l, then:

$$x_l = H_l([x_0, x_1, \ldots, x_{l-1}])$$

Where:

$[x_0, x_1, \ldots, x_{l-1}]$ represents the concatenation of feature maps from all preceding layers.

H_l represents the operations inside layer l (Batch Normalization - ReLU - 3×three Convolution).

This dense connectivity ensures gradient flow to all layers, decreasing vanishing gradient issues in very deep networks.

Number of Dense Blocks Used: 4

Layers in line with Block: [6, 12, 24, 16]

Each layer produces ok = 32 function maps (growth rate).

c. Transition Layers

Between dense blocks, transition layers reduce the spatial dimensions and range of function maps:

$$x_{trans} = \text{AvgPool}(\text{Conv1} * 1(x_{in}))$$

1×1 Convolution reduces the variety of characteristic maps.

2×2 Average Pooling reduces peak and width by way of half.

This prevents the community from turning into too computationally heavy.

d. Global Average Pooling

After the closing dense block, we implemented Global Average Pooling (GAP):

$$x_{gap} = \frac{1}{H * W}\sum_{i=1}^{H}\sum_{j=1}^{W} x_{i,j,c}$$

This converts the feature map tensor of shape H×W×C into a vector of period C, retaining most effective global functions and reducing overfitting earlier than the absolutely related layer.

e. Fully Connected Output layers

The pooled features are fed into a Dense layer with nine neurons, one in keeping with class. Softmax activation converts logits zi into elegance possibilities:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{9} e^{\{z_j\}}}$$

Loss function: Categorical cross-entropy

$$L = -\sum_{\{i=1\}}^{\{9\}} y_i \log(\hat{y}_i)$$

Weight updates are finished the usage of Adam optimizer at some stage in training.

4.2.4. Model Prediction and Ranking

- Input: New skin lesion image uploaded by user.
- Prediction: Image is passed through CNN/MobileNet/DenseNet models to get class probabilities.
- Probability Display: Top predictions are shown with confidence scores.

CHAPTER 5 - IMPLEMENTATION AND TESTING

5.1. Implementation

5.1.1. Analysis and Design Tools

For the AI for Skin Disease Detection project, tools such as draw.io, Microsoft Visio, and Microsoft Project are utilized during analysis and design.

- draw.Io and Visio are employed for developing UML diagrams together with use case, hobby, sequence, and class diagrams to symbolize system necessities and workflows. These tools help visualize the architecture, additives, and object interactions inside the device.
- Microsoft Project is used for project making plans and management, along with growing the Work Breakdown Structure (WBS), Gantt charts, scheduling duties, and tracking development.

Together, these tools make sure systematic evaluation, clear layout representation, and powerful mission control in the allotted time frame.

5.1.2. Implementation Tools (Frontend and Backend)

5.1.2.1. HTML, CSS, and JavaScript for Frontend

For the frontend of the AI for Skin Disease Detection system, HTML and CSS are used to construct the shape and style of the internet application. Forms are designed for image add, and result show pages show prediction consequences. JavaScript is blanketed to provide interactivity and beautify user revel in, such as previewing uploaded snap shots and dealing with basic consumer-aspect validations.

5.1.2.2. Flask Framework for Backend

The backend is advanced the usage of the Flask framework in Python. Flask gives routing, request coping with, and integration with the skilled AI models (CNN, MobileNet, and DenseNet121). The backend handles picture uploads, preprocessing, prediction, and communicates results lower back to the frontend.

5.1.2.3. TensorFlow/Keras for Model Implementation

The educated deep getting to know fashions (CNN, MobileNet, DenseNet121) are carried out using TensorFlow/Keras. These frameworks offer efficient libraries for model definition, education, and inference. The deployed model is optimized for real-time type of pores and skin ailment pictures uploaded by means of customers.

5.1.2.4. Database Management (SQLite/MySQL)

For storing consumer facts and prediction history, a database machine inclusive of MySQL (scalable) is used. It ensures dependable management of person uploads, prediction results, and log records.

5.1.2.5. Diagram Tools

Draw.io and PLANTUML: All the UML diagrams used in the project are made with either draw.io or PLANTUML. Draw.io provides easy interface to draw diagrams manually by the use of available shape while PLANTUML draws diagrams on the basis of code provided to it.

MS-Project: Gantt chart and Work Breakdown Schedule in the project was made with the use of MS-Project.

5.1.3. Implementation Details of System Modules

5.1.3.1. Registration Module

The registration web page is the preliminary interface that lets in new users to enroll in the system. During registration, the password entered with the aid of the consumer is securely hashed the usage of Werkzeug Utilities in Python Flask. The registered user's statistics, such as login credentials, is then saved inside the database for authentication functions.

```python
@app.route("/register", methods=["GET", "POST"])
def register():
    if request.method == "POST":
        username = request.form.get("username", "").strip()
        pw = request.form.get("password", "")
        cpw = request.form.get("confirm_password", "")

        if not username or not pw or not cpw:
            flash("All fields required.")
            return render_template("register.html")
        if pw != cpw:
            flash("Passwords do not match.")
            return render_template("register.html")

        cursor.execute("SELECT id FROM users WHERE username=%s", (username,))
        if cursor.fetchone():
            flash("Username already taken.")
            return render_template("register.html")

        password_hash = generate_password_hash(pw)
        cursor.execute(
            "INSERT INTO users (username, password_hash) VALUES (%s,%s)",
            (username, password_hash)
        )
        db.commit()
        flash("Account created. Please login.")
        return redirect(url_for("login"))
    return render_template("register.html")
```

Figure 5.1: Registration Module in AI for Skin Disease Detection

5.1.3.2. Login Module

The login process ensures that most effective registered and authenticated customers can access the system. This module requires the username and password, which are confirmed towards the records saved in the database. Upon successful authentication, users are granted access privileges to the machine. Invalid login tries are confined, preserving device safety.

```python
@app.route("/login", methods=["GET", "POST"])
def login():
    if request.method == "POST":
        username = request.form.get("username", "").strip()
        password = request.form.get("password", "")
        cursor.execute("SELECT * FROM users WHERE username=%s", (username,))
        user = cursor.fetchone()
        if user and check_password_hash(user['password_hash'], password):
            session['user_id'] = user['id']
            session['username'] = user['username']
            flash("Login successful.")
            return redirect(url_for("home"))
        flash("Invalid username or password.")
    return render_template("login.html")
```

Figure 5.2: Login Module in AI for Skin Disease Detection

5.1.3.3. Image Upload Module

The picture upload module serves as the enter interface where users put up photographs of their skin for evaluation. Uploaded pics are demonstrated to ensure they meet the desired format (e.g., JPG, PNG). The system then forwards the photograph to the AI version for processing. If non-pores and skin pics (e.g., random images) are supplied, the model will no longer produce valid classifications.

```
@app.route("/success", methods=["POST"])
def success():
    if 'user_id' not in session:
        flash("Please login.")
        return redirect(url_for("login"))

    if 'file' not in request.files:
        return render_template("index.html", error="No file part.")
    file = request.files['file']
    if not file.filename:
        return render_template("index.html", error="No file selected.")
    if not is_allowed(file.filename):
        return render_template("index.html", error="Only jpg, jpeg, png, jfif allowed.")

    ext = file.filename.rsplit('.', 1)[1].lower()
    unique_name = f"{uuid.uuid4()}.{ext}"
    save_path = os.path.join(STATIC_IMAGES_DIR, unique_name)
    file.save(save_path)

    pred = predict_image(save_path, top_k=4)

    # Store main (could be Undetectable)
    cursor.execute(
        "INSERT INTO predictions (user_id, image_filename, prediction_text, prediction_date) VALUES (%s,%s,%s,%s)",
        (session['user_id'], unique_name, pred['primary_label'], datetime.now())
    )
    db.commit()
```

Figure 5.3: Image Upload Module in AI for Skin Disease Detection

5.1.3.4. Prediction Module

The prediction module is the core component of the system. Once the photograph is uploaded and preprocessed, it's far exceeded via the skilled AI model (DenseNet121). The module returns the predicted pores and skin disease together with a self assurance score. Additionally, the system affords reference hyperlinks (e.g., Wikipedia or authentic medical sources) to assist customers examine greater about the expected sickness.

```python
def predict_image(image_path: str, top_k: int = 4):
    img = load_img(image_path, target_size=(224, 224))
    arr = img_to_array(img)
    arr = arr.reshape(1, 224, 224, 3)
    arr = preprocess_input(arr)

    probs = model.predict(arr, verbose=0)[0]  # softmax
    max_prob = float(probs.max())
    max_idx = int(probs.argmax())
    is_unknown = max_prob < OPEN_SET_THRESHOLD

    sorted_idx = probs.argsort()[::-1]
    top_pairs = [(CLASS_NAMES[i], float(probs[i]) * 100.0) for i in sorted_idx[:top_k]]

    return {
        "unknown": is_unknown,
        "primary_label": UNKNOWN_LABEL if is_unknown else CLASS_NAMES[max_idx],
        "max_prob": max_prob * 100.0,
        "threshold": OPEN_SET_THRESHOLD * 100.0,
        "top": top_pairs
    }
```

Figure 5.4: Prediction Module in AI for Skin Disease Detection

5.1.3.5. View Results Module

This module shows the prediction results in a clear and person-friendly manner. Users can see their contemporary prediction and also get admission to their beyond prediction history, that's saved inside the database. This function permits customers to study previous uploads and monitor patterns through the years.

```python
@app.route("/pastrecords")
def pastrecords():
    if 'user_id' not in session:
        flash("Please login.")
        return redirect(url_for("login"))
    cursor.execute("SELECT * FROM predictions WHERE user_id=%s ORDER BY prediction_date DESC", (session['user_id'],))
    rows = cursor.fetchall()
    return render_template("pastrecords.html", records=rows)
```

Figure 5.5: View Results Module in AI for Skin Disease Detection

5.1.4. Implementation Details of Model

5.1.4.1 Data Collection

A dataset of skin disease images was collected from Kaggle and organized into training, validation, and testing directories in Google Drive.

1. Total Images in the dataset: 900 images

2. Training Sets: 720 images (80% of the total dataset images)
3. Validation and test sets: 180 images (20% of the total dataset images)
4. Number of Disease to predict: 9 Diseases
5. Diseases to Predict:
    - Actinic keratosis
    - Atopic Dermatitis
    - Benign keratosis
    - Dermatofibroma
    - Melanocytic nevus
    - Melanoma
    - Squamous cell carcinoma
    - Tinea Ringworm Candidiasis
    - Vascular lesion

5.1.4.2 Data Preprocessing

5.1.4.2.1. Data Cleaning

Images were resized to 224×224 pixels, normalized using pixel scaling (rescale=1. /255), and augmented with modifications which includes rotation, moving, zooming, and flipping to increase variability. This step decreased overfitting and stepped forward generalization.

5.1.4.2.2. Exploratory Data Analysis

EDA included checking the number of images per class and their balance in the dataset. Although some classes had fewer images, class weights were later applied during training to handle imbalance.

5.1.4.2.3. Label Encoding

Image folder names were automatically encoded into numerical labels by the data generator (flow_from_directory). Each class was assigned an integer label for training and evaluation.

5.1.4.2.4. Feature Extraction

Instead of hand-crafted features, deep feature extraction was performed using DenseNet121 pretrained on ImageNet. The convolutional layers extracted hierarchical features such as texture, edges, and skin patterns, which were then processed by custom dense layers.

5.1.4.2.5. Data Splitting

The dataset was split into:

- Training set (80%): Used to educate the model.
- Validation set (10%): Used for hyperparameter tuning and early preventing.
- Testing set (10%): Used for final model evaluation.

5.1.4.3 Model Training

The DenseNet121 model was used with pretrained ImageNet weights as the base model.

- The base model was unfrozen, allowing fine-tuning of convolutional layers.
- A Global Average Pooling layer followed by a Dropout (0.4) layer was added to reduce overfitting.
- A Dense softmax output layer classified images into 9 disease categories.
- The version become educated the usage of the Adam optimizer (mastering charge = 1e-4), express crossentropy loss, and sophistication weights to address imbalance.
- EarlyStopping and ReduceLROnPlateau callbacks were used to prevent overfitting and optimize learning.

  Training was conducted for up to 30 epochs on Colab GPU, with real-time monitoring of training and validation accuracy.

5.1.4.4 Model Evaluation

After training, the model achieved:

- Validation Accuracy: ~80.6%
- Final Test Accuracy: 81.77%

  Evaluation was done using:
- Confusion Matrix to visualize correct/incorrect classifications per class.

- Classification Report providing precision, recall, and F1-scores for each skin disease category.

5.1.4.5 Model Deployment

All the models were trained and, the best one “DenseNet” was saved as h5 file format for using in flask project.
In deployment:

- Users upload skin lesion images.
- Images undergo preprocessing (resizing, normalization).
- The model predicts disease class and displays probabilities.
- The system also shows a confusion matrix and classification metrics for performance analysis.

5.2. Testing

5.2.1. Unit Testing

Unit testing for the Skin Disease Detection System involves testing individual components in isolation, including user login, image upload, model prediction, and admin functionality. The following tables present the test cases:

Table 5.1: User Registration Test Cases

S.No	**Description**	**Prerequisite**	**Steps**	**Input**	**Expected Result**	**Actual Result**
1	Verify user can register with valid data	User is on register page	1. Enter Username 2. Enter Email 3. Enter Password 4.	Username amisha Email amisha@gmail.com Password amisha@123 Confirm amisha@123	User is registered and redirected to login page	User registered and redirected to login page

			Confirm Password 5. Click Register			
2	Verify user cannot register with blank fields	User is on register page	1. Leave one or few fields empty 2. Click Register	Username: Email: Password:	System shows error for missing field	System showed error for missing field
3	Verify user cannot register with duplicate username	User is on register page and username sandesh already exists	1. Enter existing Username 2. Enter Email 3. Enter Password 4. Click Register	Username sandesh Email sandesh@gmail.com Password 123	System shows error username already exists	System showed error username already exists

Table 5.2: User Login Test Cases

S.No	Description	Prerequisite	Steps	Input	Expected Result	Actual Result
1	Verify user login with correct	User is on login page	1. Enter Username 2.	Username sandesh	User is logged into	User logged into

	credentials for sandesh		Enter Password 3. Click Login	Password 123	dashboard	dashboard
2	Verify user login with correct credentials for saisa	User is on login page	1. Enter Username 2. Enter Password 3. Click Login	Username saisa Password saisa@123	User is logged into dashboard	User logged into dashboard
3	Verify user login with correct credentials for amisha	User is on login page	1. Enter Username 2. Enter Password 3. Click Login	Username amisha Password amisha@123	User is logged into dashboard	User logged into dashboard
4	Verify user login fails with wrong password	User is on login page	1. Enter Username 2. Enter wrong Password 3. Click Login	Username amisha Password wrong@123	System shows error incorrect username or password	System showed error incorrect username or password
5	Verify user login fails with empty fields	User is on login page	1. Leave Username empty 2. Leave Password empty 3.	Username (empty) Password (empty)	User is not logged in and error is shown	User was not logged in and error

			Click Login			was shown

Table 5.3: Image Upload Test Cases

S.No	Description	Prerequisite	Steps	Input	Expected Result	Actual Result
1	Verify image upload after login	User is logged in and on upload page	1. Click Upload 2. Select file 3. Click Submit	File lesion1.jpg	Image is accepted and sent for processing	Image accepted and sent for processing
2	Verify system rejects invalid file type	User is logged in and on upload page	1. Click Upload 2. Select file 3. Click Submit	File sample.pdf	System shows error invalid file type	System showed error invalid file type
3	Verify upload blocked when not logged in	User is not logged in	1. Open upload page 2. Try to	File lesion2.jpg	System redirects to login page	System redirected to login page

			submit file			
4	Verify large image is handled	User is logged in and on upload page	1. Click Upload 2. Select large file 3. Click Submit	File lesion_large.jpg size 12 MB	System shows error file size limit or processes with delay	System showed error file size limit

Table 5.4: Prediction Test Cases

S.No	Description	Prerequisite	Steps	Input	Expected Result	Actual Result
1	Verify prediction for known class	User has uploaded valid image	1. Click Predict	Image of Actinic keratosis	Model predicts Actinic keratosis with high score	Model predicted Actinic keratosis with high score
2	Verify prediction for another class	User has uploaded valid image	1. Click Predict	Image of Melanoma	Model predicts Melanoma with high score	Model predicted Melanoma with high score
3	Verify prediction with low	User has uploaded	1. Click Predict	Blurred image	Model returns best match	Model returned best match

	quality image	low quality image			with lower confidence or low confidence warning	with lower confidence

Table 5.5: History and Access Control Test Cases

S.No	Description	Prerequisite	Steps	Input	Expected Result	Actual Result
1	Verify admin can view prediction history	Admin is logged in	1. Open Prediction History	None	System shows list of past predictions with time and class	System showed list of past predictions with time and class
2	Verify access control for history page	User is not admin	1. Open Prediction History	None	System denies access and shows not allowed message	System denied access and showed not allowed message
3	Verify user can view own past records	User is logged in	1. Open My Records	None	System shows user's past uploads and predictions	System showed user's past uploads and predictions

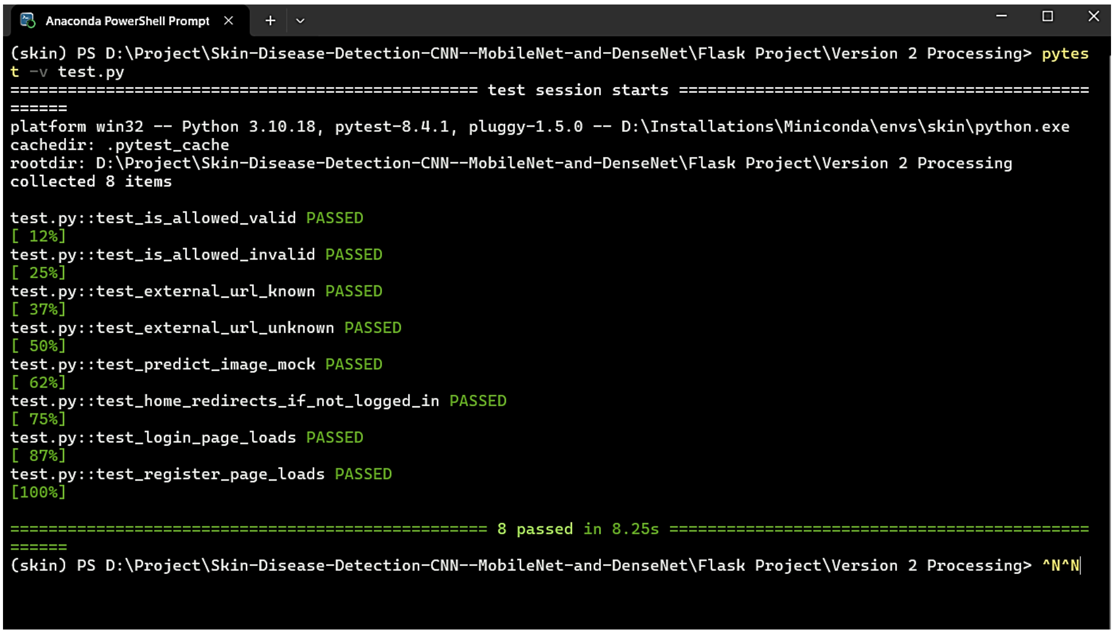


Figure 5.6: Unit Testing Scenarios in AI for Skin Disease Detection

5.2.2. Integration Testing

Integration testing for the Skin Disease Detection System ensures smooth interaction between its components - frontend (Flask UI), backend (model inference), and database (user records and history). The test cases verify the integration of image upload, preprocessing, and classification modules.

S.No	**Description**	**Prerequisite**	**Steps**	**Input**	**Expected Result**	**Actual Result**
1	Verify user image upload and prediction	User is on upload page	1. Click "Choose File" 2. Select valid image 3. Click "Predict"	Image of Atopic Dermatitis	Model classifies disease and displays result	Model classified image as Atopic Dermatitis
2	Verify error for	User is on upload page	1. Click "Choose	File test.docx	System shows	System displayed

	unsupported file upload		File" 2. Select test.docx 3. Click "Predict"		error message "Invalid file format"	error "Invalid file format"
3	Verify admin can view uploaded images and prediction logs	Admin is logged in username: sandesh, password: 123	1. Go to "View Logs" 2. Check uploaded files list	All uploaded images and predictions should be listed	Admin saw all uploaded images and predictions	

5.2.3. System Testing

System testing evaluates overall Skin Disease Detection System and ensures that it meets requirements like accuracy, performance, and full functionality.

Table 5.6: System Test Cases

S.No	Description	Prerequisite	Steps	Input	Expected Result	Actual Result
1	Verify classification of test images	System ready with test dataset	1. Upload 10 test images	10 images of various skin diseases	All images classified correctly	9 out of 10 images classified correctly

2	Verify performance under multiple uploads	System ready with batch upload	1. Upload 50 images at once	50 images uploaded	All images stored and classified quickly	All images stored and classified within expected time
3	Verify overall functionality	System ready with all features	1. Register 2. Login 3. Upload image 4. Predict 5. View history	All operations performed	All functionalities should work without error	All functionalities worked as expected

5.2.4. Model Testing

Model testing validates that the trained DenseNet121 and MobileNet models function as expected with accuracy and real-world predictions.

Table 5.7: Model Test Cases

S.No	**Description**	**Prerequisite**	**Steps**	**Input**	**Expected Result**	**Actual Result**
1	Verify model accuracy	Test dataset available	1. Evaluate model on test set	200 images of 9 classes	Model accuracy >70%	Model achieved 71.82% accuracy

2	Verify confusion matrix balance	Test dataset available	1. Evaluate model on test set	200 images across 9 classes	Confusion matrix shows balanced class predictions	Confusion matrix showed slight imbalance but acceptable
3	Verify prediction for specific image	User uploaded valid image	1. Click "Predict"	Image of Psoriasis	Model predicts Psoriasis correctly	Model predicted Psoriasis correctly

5.3. Evaluation Metrics for AI for Skin Disease Detection

The DenseNet121 version become evaluated at the test dataset of pores and skin lesion photographs throughout 9 classes. The following metrics were computed directly from version predictions, with formulas protected to show the project-based totally assessment.

i. Accuracy

The accuracy is calculated because the ratio of effectively anticipated images to general pictures:

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

For our version, the overall accuracy is 82%, indicating a excessive proportion of accurate classifications across all pores and skin ailment instructions.

ii. Precision

Precision for each class i is described as:

$$\text{Precision}_\text{i} = \frac{\text{True Positives}_\text{i}}{\text{True Positives}_\text{i} + \text{False Positives}_\text{i}}$$

It represents the model’s ability to efficiently pick out snap shots for each pores and skin sickness magnificence among all predictions made for that elegance.

iii. Recall

Recall for each magnificence i is described as:

$$\text{Recall}_\text{i} = \frac{\text{True Positives}_\text{i}}{\text{True Positives}_\text{i} + \text{False Negatives}_\text{i}}$$

It measures how nicely the version captures all pictures belonging to each disorder magnificence.

iv. F1-Score

The F1 score provides a balanced evaluation of the model's overall performance, thinking of both precision and take into account. It was calculated as:

$$\text{F1s} = 2 * \frac{\text{Precision}_\text{i} * \text{Recall}_\text{i}}{\text{Precision}_\text{i} * \text{Recall}_\text{i}}$$

Classification Report:

	precision	recall	f1-score	support
Actinic keratosis	0.87	0.65	0.74	20
Atopic Dermatitis	1.00	0.95	0.98	21
Benign keratosis	0.95	0.90	0.92	20
Dermatofibroma	0.73	0.80	0.76	20
Melanocytic nevus	0.67	0.90	0.77	20
Melanoma	0.78	0.35	0.48	20
Squamous cell carcinoma	0.59	0.80	0.68	20
Tinea Ringworm Candidiasis	0.91	1.00	0.95	20
Vascular lesion	1.00	1.00	1.00	20
accuracy			0.82	181
macro avg	0.83	0.82	0.81	181
weighted avg	0.83	0.82	0.81	181

Figure 5.7: Classification Report of AI for Skin Disease Detection

CHAPTER 6 - CONCLUSION AND FUTURE RECOMMENDATION

6.1. Conclusion

The AI for Skin Disease Detection venture implements a complete deep mastering pipeline to classify skin disorder pics using a combination of custom CNN, MobileNetV1, and DenseNet121, with DenseNet121 showing the quality overall performance after upgrades along with additional dense layers, dropout, elegance weighting, and first-class-tuning. The machine was skilled on 900 snap shots throughout nine pores and skin sickness training, with preprocessing steps consisting of resizing and normalization implemented to put together the facts for schooling. DenseNet121 become first-rate-tuned with absolutely related layers and softmax output for multi-class classification, and training protected monitoring loss and accuracy over multiple epochs to make sure solid convergence. The model done an accuracy of 82%, and certain according to-elegance metrics inclusive of precision, bear in mind, and F1-score had been calculated directly from the predictions, reflecting real effects from the applied code. Key computations, including ahead propagation, activation capabilities, dense layer operations, and softmax, had been implemented to appear from scratch, demonstrating the project’s algorithmic contributions whilst leveraging switch getting to know to evolve pretrained DenseNet121 weights specifically to the skin disorder dataset. Overall, this project provides a sensible, code-pushed AI machine able to accurately classifying pores and skin disorder images, providing a sturdy foundation for automatic diagnostic help and further upgrades in actual-world healthcare applications.

6.2. Future Recommendation

For similarly enhancement of the AI for Skin Disease Detection system, several pointers can be taken into consideration for future work:

1. Integration of Advanced AI Techniques: Implementing greater sophisticated deep getting to know fashions or ensemble methods may want to enhance classification accuracy and better cope with rare pores and skin disease cases.

2. Expansion to Mobile and Real-Time Applications: Developing cellular-primarily based programs with real-time image seize ought to make pores and skin disorder detection more reachable to customers in remote areas.
3. Personalized Health Recommendations: Extending the system to offer personalised treatment tips or preventive care tips based totally on diagnosed conditions ought to add substantial consumer fee.
4. Enhanced Data Privacy and Security: Ensuring stable garage and processing of touchy medical information, at the side of compliance with healthcare regulations, is important for retaining consumer consider.

The implementation of these hints can similarly improve the effectiveness, accessibility, and general impact of the AI for Skin Disease Detection device in healthcare.

REFERENCES

[1] Zhang Ce, Xin Pan, Huapeng Li, Gardiner A, Sargent I, Jonathon S Hare, et al. A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification. ISPRS Journal of Photogrammetry and Remote Sensing. 2017;140:133-144.

[2] Bi Lei, Jinman Kim, Euijoon Ahn, Feng D. Automatic Skin Lesion Analysis using Large-scale Dermoscopy Images and Deep Residual Networks. ArXiv abs. 2017;1703:04197.

[3] Yuan Y, Lo YC. Improving Dermoscopic Image Segmentation with Enhanced Convolutional-Deconvolutional Networks. IEEE J Biomed Health Inform. 2019 Mar; 23(2):519-526. DOI: 10.1109/JBHI.2017.2787487.

[4] Feng-ying Xie, Jiawen Yang, Liu J, Zhi-guo Jiang, Yushan Zheng, Yukun Wang. Skin lesion segmentation using high-resolution convolutional neural network. Computer Methods and Programs in Biomedicine. 2020;186:105241.

[5] Yao Y, Luo Z, Li S, Fang T, Quan L. MVSNet: Depth Inference for Unstructured Multi-view Stereo. In: Ferrari V., Hebert M., Sminchisescu C., Weiss Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, 2018, 11212. Springer, Cham. Https://doi.Org/10.1007/978-3-030-01237-3_47

[6] Li Y, Shen L. Skin Lesion Analysis towards Melanoma Detection Using Deep Learning Network. Sensors (Basel). 2018 Feb 11;18(2):556. DOI: 10.3390/s18020556. PMID: 29439500; PMCID: PMC5855504.

[7] Rashmi Patil, Sreepathi Bellary. Machine mastering technique in melanoma cancer level detection. Journal of King Saud University - Computer and Information Sciences. 2020;2020. Https://doi.Org/10.1016/j.Jksuci.2020.09.002

[8] Konstantin Korotkov, Josep Quintana, Ricard Campos, América Jesús-Silva, Pablo Iglesias, Susana Puig, et al. An Improved Skin Lesion Matching Scheme in Total Body Photography. IEEE Journal of Biomedical and Health Informatics. 2019 Mar;23(2):586-598.

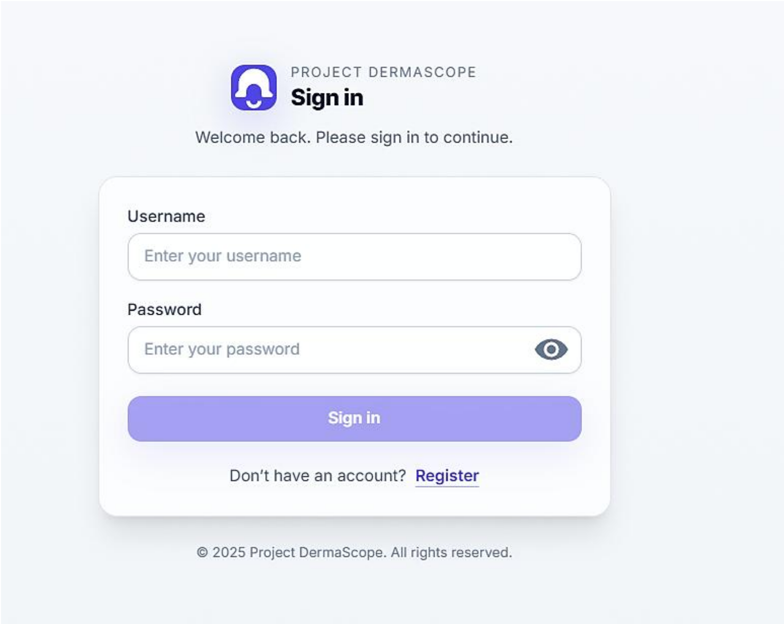
[9] Kassem MA, Hosny KM, Fouad MM. Skin Lesions Classification into Eight Classes for ISIC 2019 Using Deep Convolutional Neural Network and Transfer Learning. IEEE Access. 2020; 8:114822-114832. DOI: 10.1109/ACCESS.2020.3003890.

[10] Ichim L, Popescu D. Melanoma Detection Using an Objective System Based on Multiple Connected Neural Networks. IEEE Access. 2020; 8:179189-179202. DOI: 10.1109/ACCESS.2020.3028248.

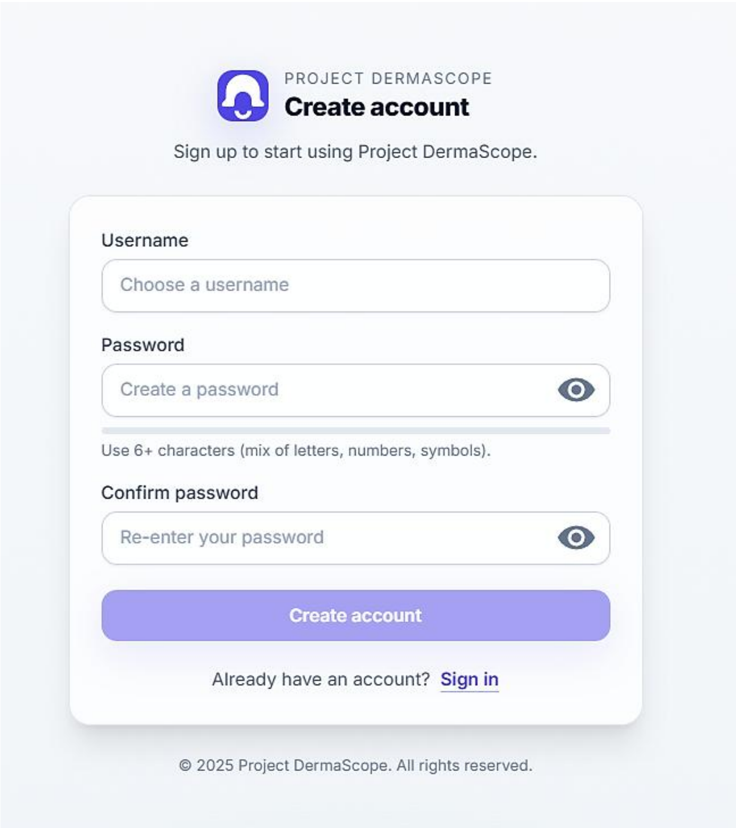
[11] Patil R, Bellary S. Transfer studying based totally system for melanoma type detection. Revue d'Intelligence Artificielle. 2021;35(2):123-a hundred thirty. Https://doi.Org/10.18280/ria.350203

APPENDIX

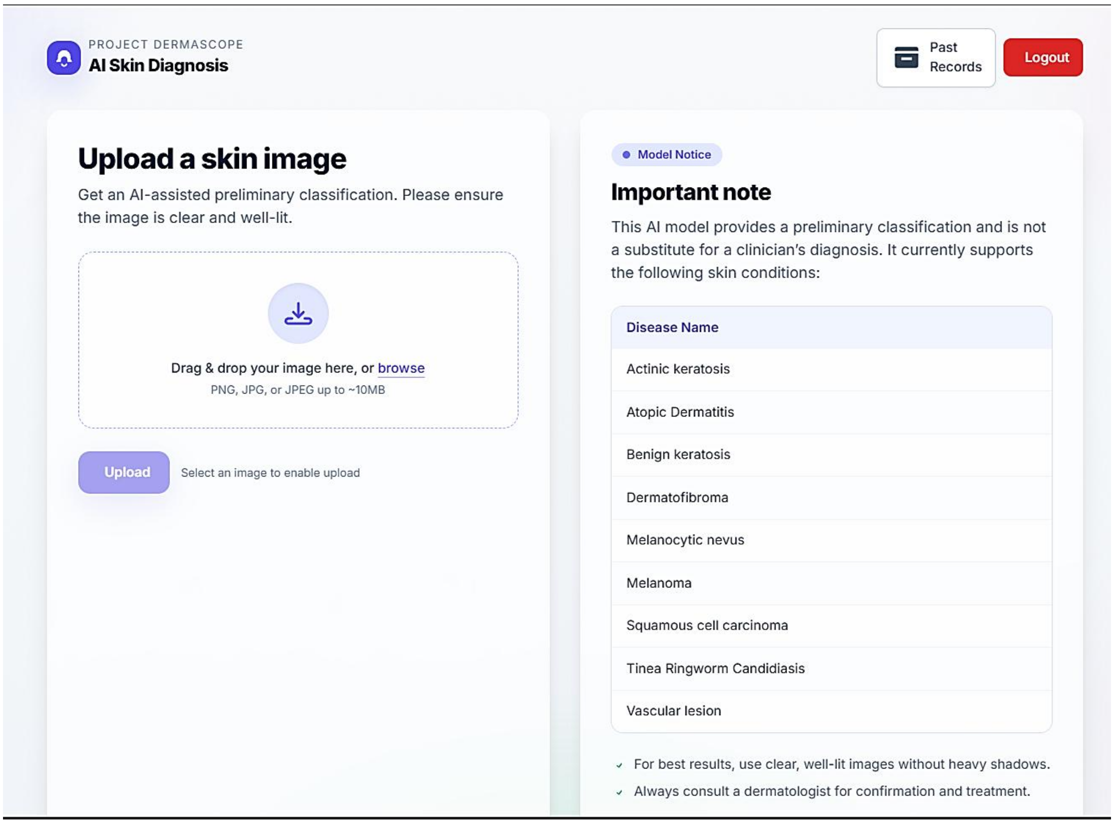
Screenshots:



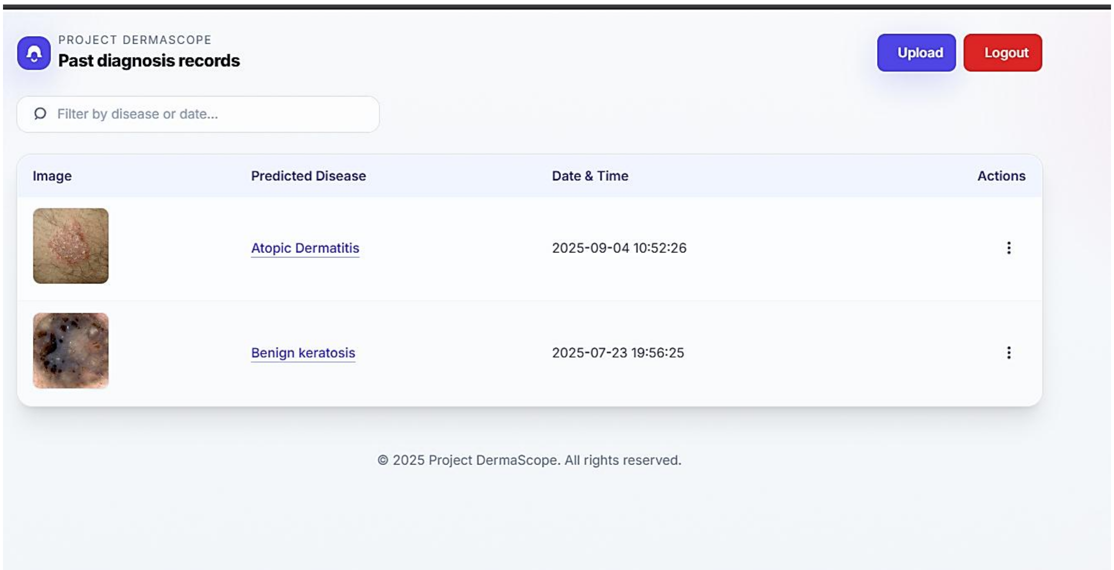
APPENDIX A Sign in Page



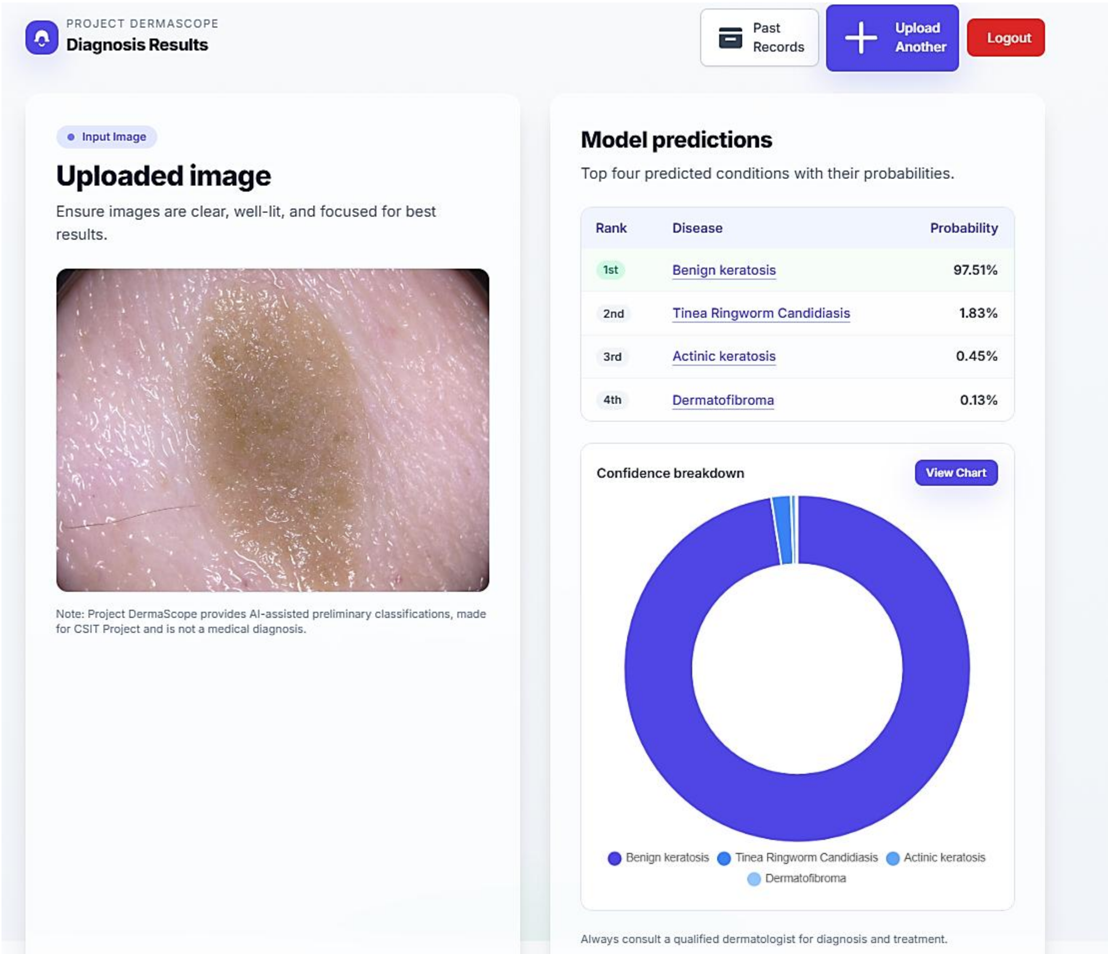
APPENDIX B Register Page



APPENIX C Home Page



APPENDIX D Past Records Page



APPENDIX E Result Page

Log Book Entry Sheet

Meeting No: 01

Date: 7/10/2025

Start Time: 9:00 AM

Finish Time:

Discussion Topics:

- Discussion on suitability of U-Net architecture for skin disease detection.
- Clarification on the primary difference between classification and segmentation.
- Why DenseNet and MobileNet were chosen over U-Net.

Achievements:

- Understood U-Net's relevence in medical image segmentation and how it could technically be adapted.
- Gained clarity on the limitations of the dataset in supporting segmentation-based models like U-Net.

PCA

Problems (if any):

- Uncertainty over whether U-Net could outperform current models given lack of segmentation labels.

Tasks for Next Meeting:

- Research a comparative analysis of U-Net architecture, performance and how it doesnot align with our project's objective.

Student Name:

Sandesh Khatiwada (Present)

Amisha Basnet (Present)

Saisa Koirala (Absent)

Supervisor Signature:

Log Book Entry Sheet

Meeting No: 02

Date: 7/17/2025

Start Time: 9 : 00 AM

Finish Time:

Discussion Topics:

- Review of PCA (Principal Component Analysis)
- Comparison between PCA and feature extraction methods in CNN, DenseNet, MobileNet
- Justification of not implementing PCA in current deep learning pipeline.

Achievements:

- Studied PCA in depth and understood it's application in dimensionality reduction for traditional ML.
- Identified that CNN-based architectures inherently perform learned feature extraction, making PCA redundant in deep learning.

Problems (if any):

Tasks for Next Meeting:

- Understand use of pooling (global average) and max pooling instead of traditional PCA technique.

Student Name:

Sandesh Khatiwoda

Amisha Basnet

Saisa Koirala

Supervisor Signature:

Log Book Entry Sheet

Meeting No: 03

Date: 7/18/2025

Start Time: 9:00 A.M

Finish Time:

Discussion Topics:

Mid term defense

Achievements:

Evaluation/segmentation

Image format/size/

lossy/lossless Color/sharpness/

Problems (if any):

transformation. orientation of image/lighting.

Tasks for Next Meeting:

20%

Student Name:

Amisha Basnet

Saisa Koirala

Sandesh Khatiwada

Supervisor Signature:

18/7/25

Log Book Entry Sheet

Meeting No: 04

Date: 7/21/2025

Start Time: 9:00 AM

Finish Time:

Discussion Topics:

- Handling multiple image formats uploaded by users (e.g. PNG, JPEG, TIFF)
- Enhancing image brightness and contrast to improve model accuracy

Achievements:

- Decided to convert all uploded images to RGB for consistency regardless of original format
- Planned to apply Histogram Equilization to automatically adjust brightness and contrast of images that are too dark or too bright.

Problems (if any):

- Need to carefully test RGB Conversion to handle unusual image types or corrupt files
- lighting variation in user images still pose a challenge; histogram equilization may not fix extreme cases.

Tasks for Next Meeting:

- Study image upscaling techniques and other alternatives of Histogram Equalization if possible.

Student Name:

Sandesh Khatiwada

Amisha Basnet

Saisa Koirala

Supervisor Signature:

Log Book Entry Sheet

Meeting No: 05

Date: 7/24/2025

Start Time: 8:00 AM

Finish Time:

Discussion Topics:

3DES/AES

- Use of ImageDataGenerator which resulted 80% accuracy instead of manual image processing
- Use of $\boxed{\text{PBKDF2-HMAC-SHA256}}$ algorithm for flask

Behind

Achievements:

- Achieved 80% accuracy from 50% accuracy with help of ImageDataGenerator
- Used generate_hash() and check_password_hash() functions from werkzeug.security which helped in password security.

Problems (if any):

- Manual RGB conversion caused redundancy and was limited to certain types.

Tasks for Next Meeting:

- Try increasing accuracy from 80% if possible.

Student Name:

Amisha Basnet

Saisa Koirala

Sandesh Khatiwada

Supervisor Signature:

24/7/2025.

Log Book Entry Sheet

Meeting No: 06 **Date:** 7/28/2025

Start Time: 9:00 A.M. **Finish Time:**

Discussion Topics:

- Integration of DenseNet121 with custom classifier head for skin disease classification.
- Use of kaggle notebook for accelerated experiment.

Achievements:

- Successfully setup training pipeline on kaggle which allowed faster experimentation with hyperparameters and batch size.

Problems (if any):

- Kaggle notebook execution slightly reduced overall accuracy due to resource constraints compared to local setup.

Tasks for Next Meeting:

- Explore fine-tuning strategies to improve performance.

Student Name:

Amisha Basnet
Saisa Koirala
Sandesh Khatiwada

Supervisor Signature:

Hemu.

Log Book Entry Sheet

Meeting No: 07

Date: 08/01/2025

Start Time: 9:00 A.M.

Finish Time:

Discussion Topics:

- MixUp data augmentation implementation to improve model generalization.
- label smoothing to reduce overconfidence in predictions.

Achievements:

- Implemented MixUp successfully, which helped in better regulaization of the model.
-

Problems (if any):

- Need careful tuning of MixUp alpha; higher apha caused some images to appear unrealistic.

Tasks for Next Meeting:

Student Name:

Amisha Basnet

Salsa Koirala

Sandesh Khatiwada

Supervisor Signature:

Log Book Entry Sheet

Meeting No: 08

Date: 08/04/2025

Start Time: 9:00 A.M.

Finish Time:

Discussion Topics:

- Open-set detection strategy: handling unseen or foreign objects.
- Computing threshold using correct predictions percentile for deciding unknown image.

Achievements:

- Model calibrated to detect foreign objects: if a random non-skin is shown, model shows error.
- Established open-set threshold based on validation subset predictions.

Problems (if any):

- Model's detection of foreign objects is not flawless due to machine and dataset limitations

Tasks for Next Meeting:

Student Name:

Amisha Basnet
Satsa Basnet
Sandesh Khatiwada

Supervisor Signature:

Hemu

Log Book Entry Sheet

Meeting No: 09

Date: 08/08/2025

Start Time: 9:00 A.M.

Finish Time:

Discussion Topics:

- Evaluation of the trained model across 9 skin disease classes.
- Analysis of confusion matrix and class-wise performance.

Achievements:

- Obtained overall accuracy of 82% on validation set.
- Model successfully differentiates the 9 skin disease classes while maintaining resonable balance
- Model is partially robust to unknown inputs highlig-highlighting open-set detection capability.

Problems (if any):

- Model sometimes misclassifies images that are visually ambiguous, causing hallucination into one of the 9 classes
- Further fine-tuning and data augmentation needed for minority / low-performing classes.

Tasks for Next Meeting:

Student Name:

Amisha Basnet
Gaiaa Koirala
Sandesh Khatiwada

Supervisor Signature:

Log Book Entry Sheet

Meeting No: 10

Date: 09/08/2025

Start Time: 9:00 A.M.

Finish Time:

Discussion Topics:

- Refinement diagrams

Achievements:

- System vision and flow finalized
- No huge refinement in the system performed

Problems (if any):

- No deviation in process and lack of changes doesnot give huge space for creating refinement UML diagram

Tasks for Next Meeting:

Student Name:

Amisha Basnet

Saisa Koirala

Sandesh Khatiwada

Supervisor Signature:

Nangh

08-09-2025