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.

………………………….

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	**Signature of HOD Coordinator**
………………………….	………………………….
Mr. Pawan Niroula Lecturer Supervisor Orchid International College Bijayachowk, Gaushala	Er. Dhiraj Kumar Jha Head of Department, Department of IT, 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

Skin conditions are a widely wide-spread global fitness problem, discovered in individuals of every age. The signs and symptoms, which range from great inclusive of acne and eczema to lifestyles-threatening situations which includes melanoma and psoriasis, are often now not identified or misdiagnosed, specifically in which there are not any dermatology capabilities. Traditional diagnostic techniques are plagued with the aid of human bias and lack of ability to scale.

With advances in Artificial Intelligence (AI) technologies, especially deep gaining knowledge of and picture type, it is now feasible to create structures that are capable of detecting diseases from photos with excessive accuracy. This assignment ambitions to utilize such AI technology the usage of CNN, MobileNet, and DenseNet121 fashions to diagnose dermatological snap shots.

The undertaking is to create a supportive AI product that enables early prognosis, reduces healthcare infrastructure overload, and improves access to care in beneath-resourced areas.

1.1. 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.

1.2. 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.

1.3. 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 maynot accurately predict real word cases perfectly along with the limitation of classification of only 9 diseases.

1.4. 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 gadget.

The record is dependent into six chapters:

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

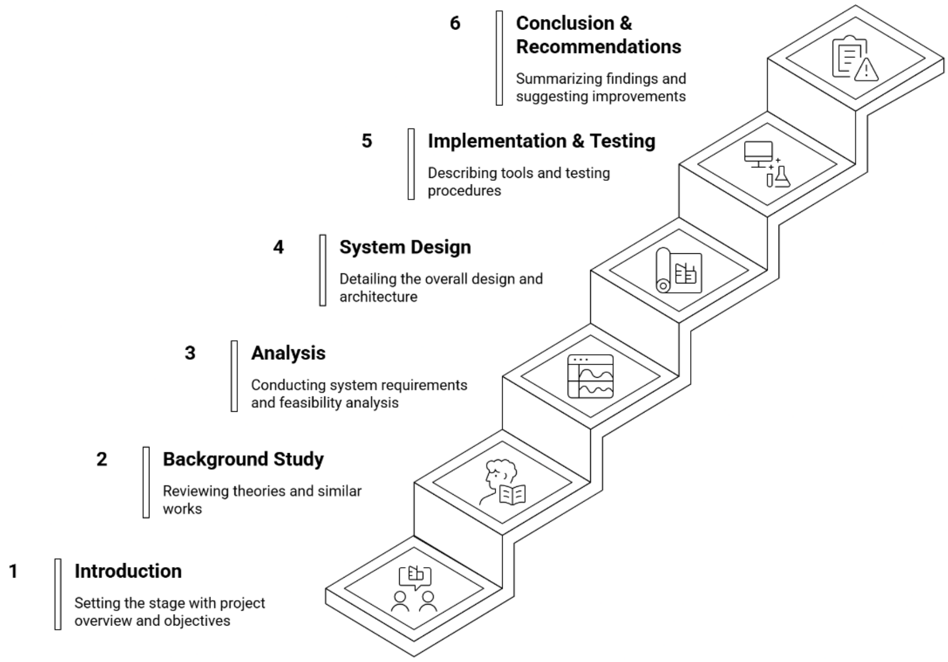


Figure 1.1: Report Organization

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.

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.

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.

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.

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.

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 analysis is the systematic process of gathering, documenting, and understanding the needs and specifications for a system. For the AI-based Skin Disease Detection System, this step looks on defining project's objectives, identifying essential features, specifying data requirements, and analysing functional and non-functional requirements.

3.1.1. Functional Requirements

The functional requirements of the skin disease detection system include:

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

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.

3.1.2. 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 gadget 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.

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

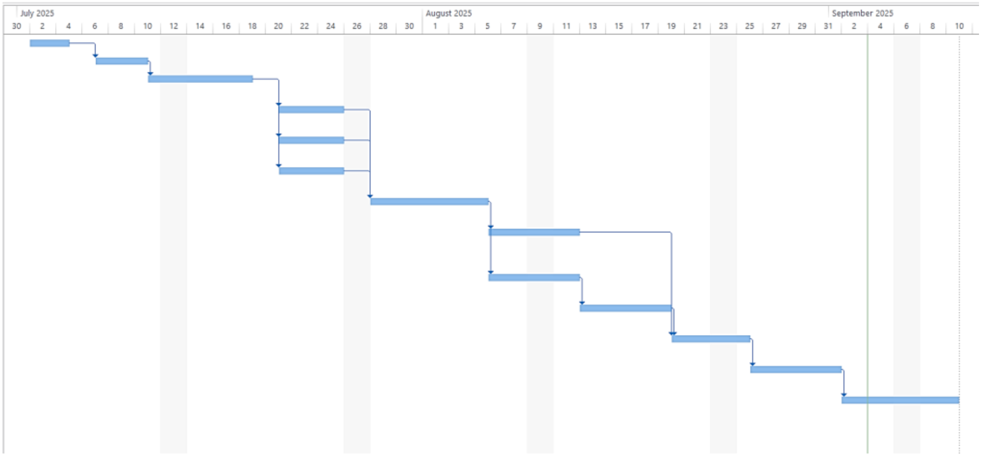


Figure 3.3: Gantt Chart of AI for Skin Disease Detection

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 gadget's object modeling, outlining the important thing instructions, their attributes, functionalities and relationships.

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

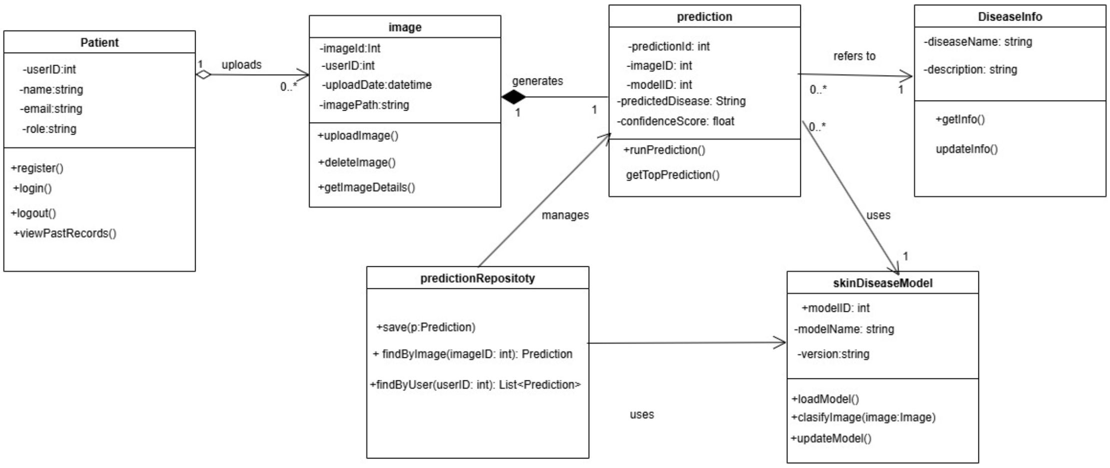


Figure 3.4: Class Diagram of AI for Skin Disease Detection

CHAPTER 4: SYSTEM DESIGN

4.1. Design

The gadget 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. Sequence Diagram

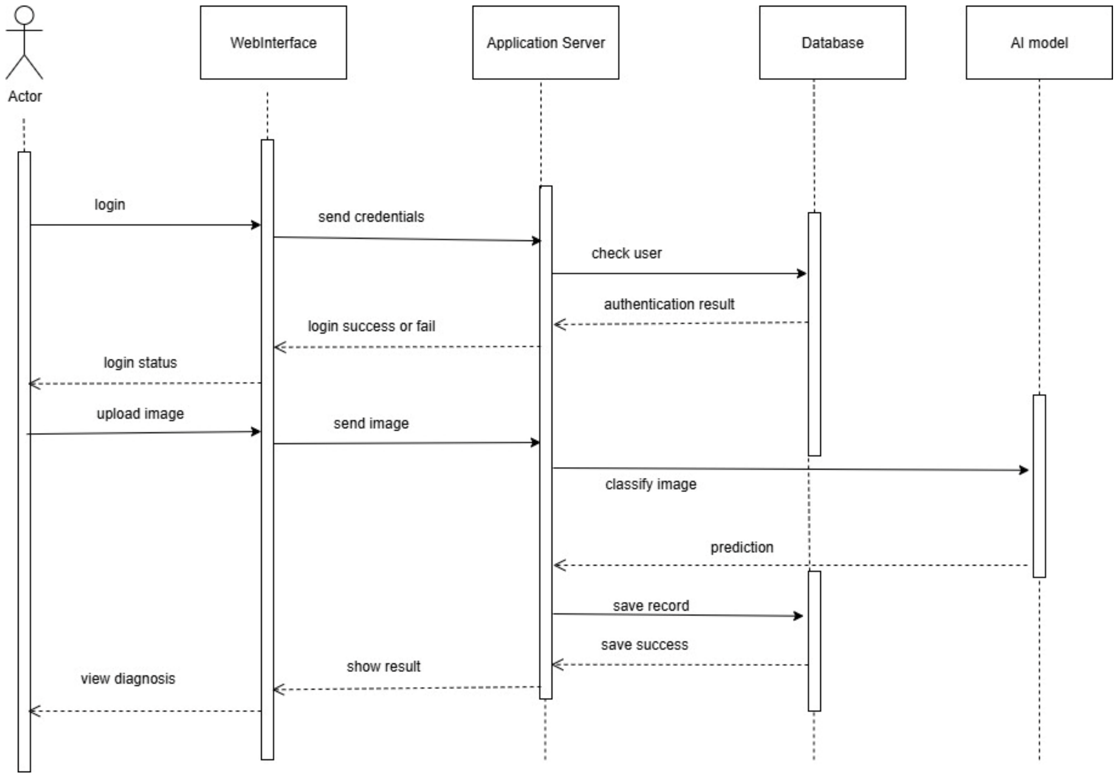


Figure 4.1: Sequence Diagram of AI for Skin Disease Detection

4.1.2. Activity Diagram

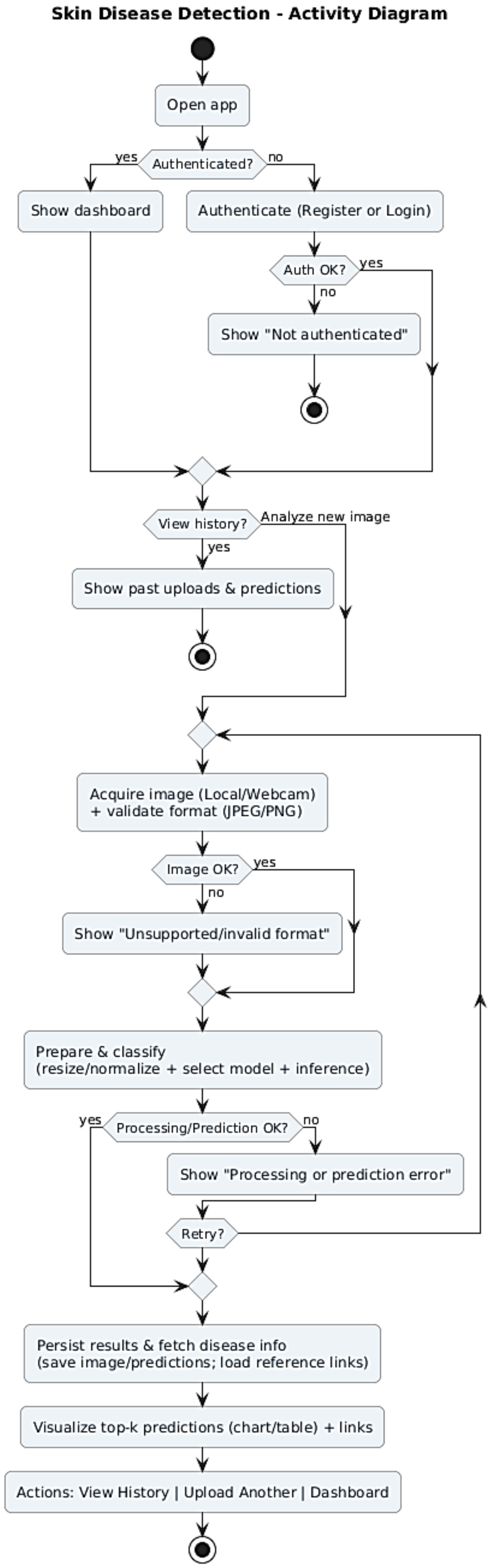


Figure 4.2: Activity Diagram of AI for Skin Disease Detection

4.1.3. Component Diagram

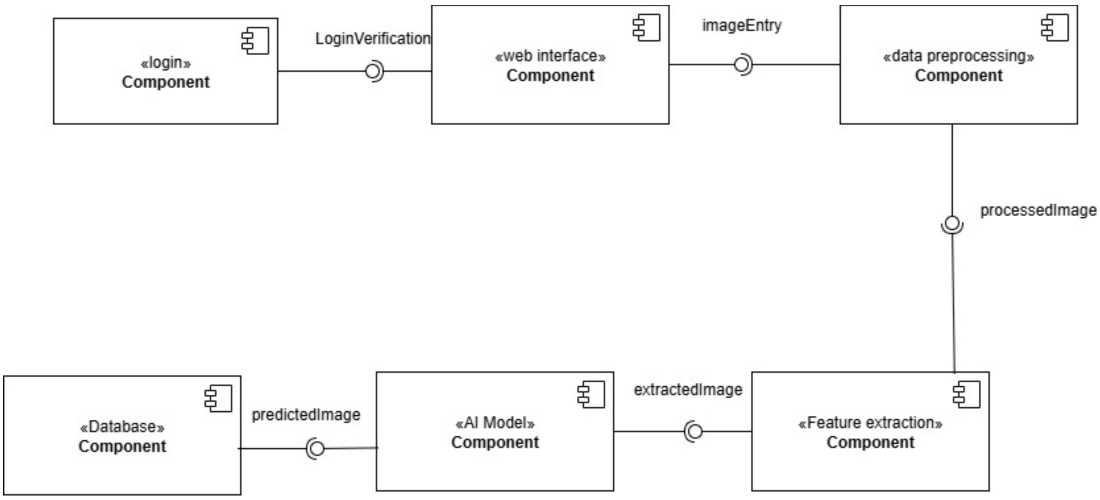


Figure 4.3: Component Diagram of AI for Skin Disease Detection

4.1.4. Deployment Diagram

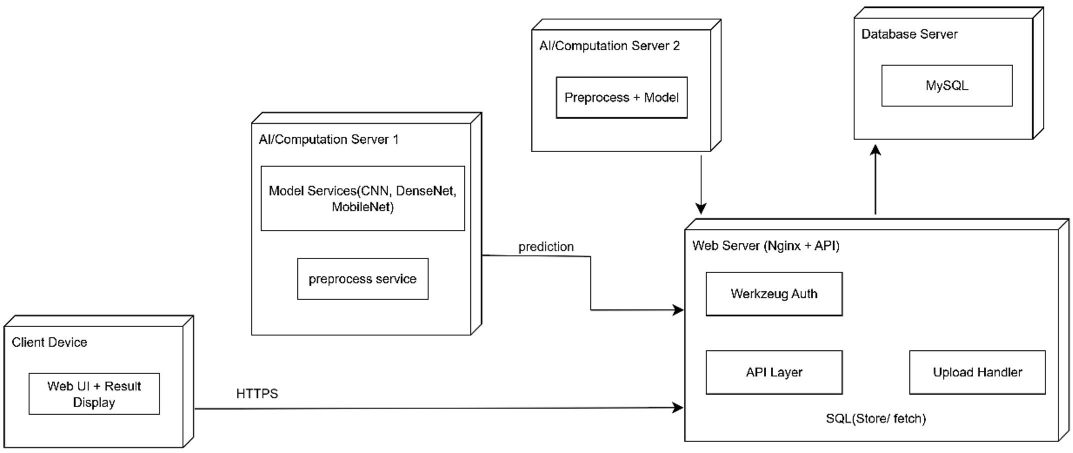


Figure 4.4: Deployment Diagram of AI for Skin Disease Detection

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.

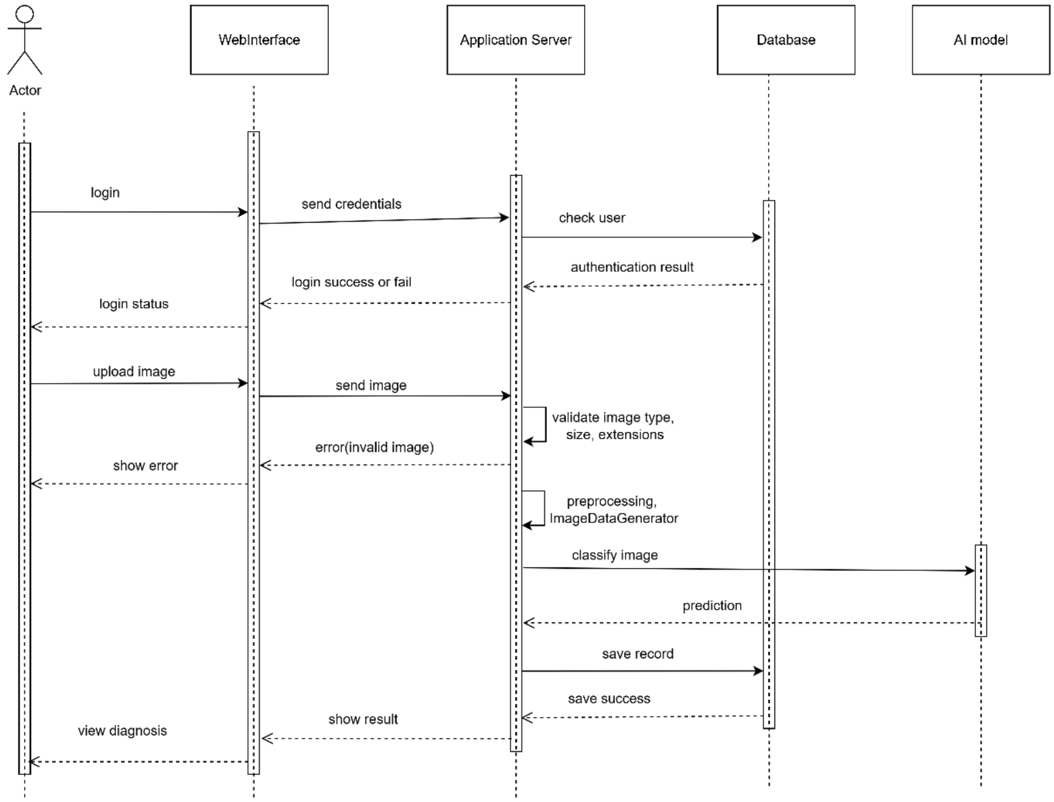


Figure 4.5: Refined Sequence Diagram of AI for Skin Disease Detection

The above refined sequence diagram 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

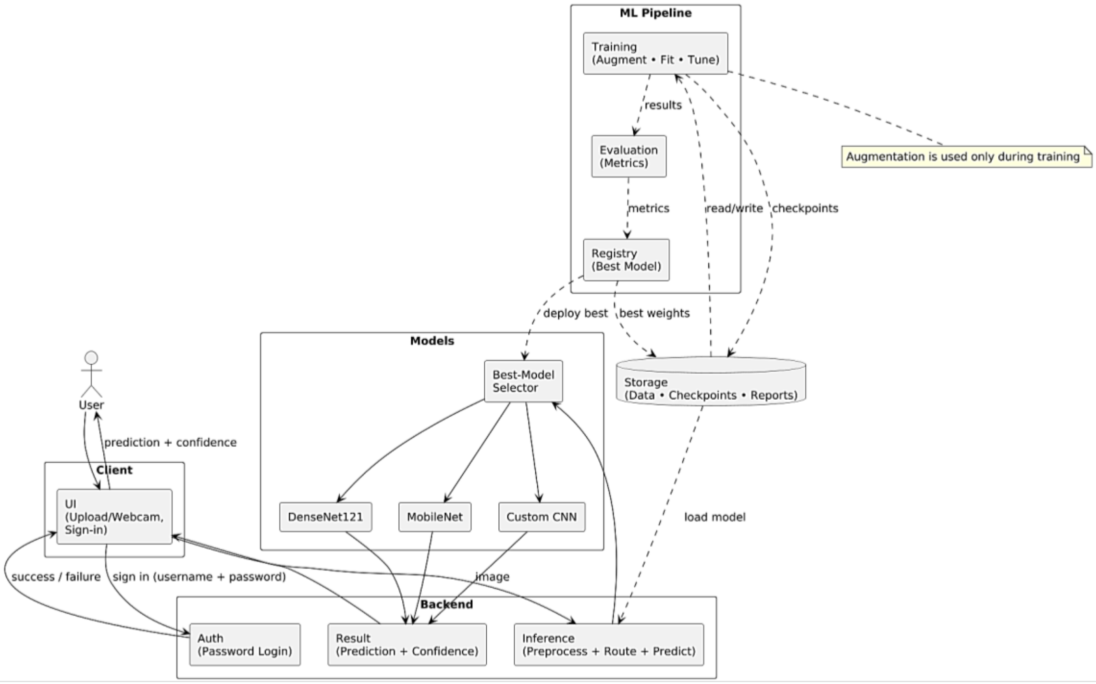


Figure 4.6: Model Architecture of AI for Skin Disease Detection

4.2. Algorithm Details

4.2.1. Convolutional Neural Network (CNN)

- Input: Preprocessed image of skin disease.
- Convolution Layer: Apply a couple of filters to extract spatial capabilities like edges, textures, and styles.
- Activation Function: ReLU applied to introduce non-linearity.
- Pooling Layer: MaxPooling to lessen spatial dimensions and retain critical features.
- Flatten Layer: Converts 2D characteristic maps right into a 1D vector.
- Fully Connected Layers: Dense layers analyze complex patterns from features.
- Output Layer: Softmax activation offers chance distribution across 9 pores and skin disorder instructions.

- Loss Function: Categorical Cross-Entropy for multi-magnificence type

4.2.2. MobileNetV1 (Transfer Learning)

- Input: Preprocessed 224×224 RGB pictures.
- Feature Extraction: Uses depthwise separable convolutions for efficient characteristic extraction.
- Pre-trained Weights: Initialized with ImageNet weights for better convergence.
- Fine-tuning: Last few layers are unfrozen to evolve to skin disease dataset.
- Output Layer: Dense layer with Softmax activation for 9 instructions.
- Ranking/Prediction: Returns top predicted skin sicknesses with self-assurance scores.

4.2.3. DenseNet121 (Transfer Learning)

- Input: Preprocessed pictures of skin lesions.
- Dense Blocks: This algorithm's feature solves vanishing gradient by passing current node's weight to future nodes.
- Transition Layers: Reduce size of feature map along with the complexity among dense blocks.
- Global Average Pooling: Reduces dimensions before absolutely connected layers.
- Fully Connected Layer: Softmax activation outputs chance across trainings.
- Advantage: Improves gradient drift, reduces vanishing gradient hassle, and leverages deep function representations.

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 gadget 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 gadget, 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 gadget. 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 gadgets) are supplied, the model will no longer produce valid classifications.

```python
@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 gadget affords reference hyperlinks (e.g., Wikipedia or authentic medical sources) to assist customers examine greater about the expected sickness.

```
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.

```
@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. Confirm Password 5. Click Register	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
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 credentials for sandesh	User is on login page	1. Enter Username 2. Enter Password 3. Click Login	Username sandesh Password 123	User is logged into dashboard	User logged into 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	Username amisha Password wrong@123	System shows error incorrect username or password	System showed error incorrect username or password

			3. Click Login			
5	Verify user login fails with empty fields	User is on login page	1. Leave Usernam e empty 2. Leave Password empty 3. Click Login	Username (empty) Password (empty)	User is not logged in and error is shown	User was not logged in and error was shown

Table 5.3: Image Upload Test Cases

S.N o	Descriptio n	Prerequisit e	Steps	Input	Expected Result	Actual Result
1	Verify image upload after login	User is logged in and on upload page	1. Click Uploa d 2. Select file 3. Click Submi t	File lesion1.jpg	Image is accepted and sent for processin g	Image accepted and sent for processin g
2	Verify system rejects invalid file type	User is logged in and on upload page	1. Click Uploa d 2. Select file 3. Click	File sample.pdf	System shows error invalid file type	System showed error invalid file type

			Submit			
3	Verify upload blocked when not logged in	User is not logged in	1. Open upload page 2. Try to submit file	File lesion2.jpg	System redirects to login page	System redirected to login page
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 quality image	User has uploaded low quality image	1. Click Predict	Blurred image	Model returns best match with lower confidence or low confidence warning	Model returned best match 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

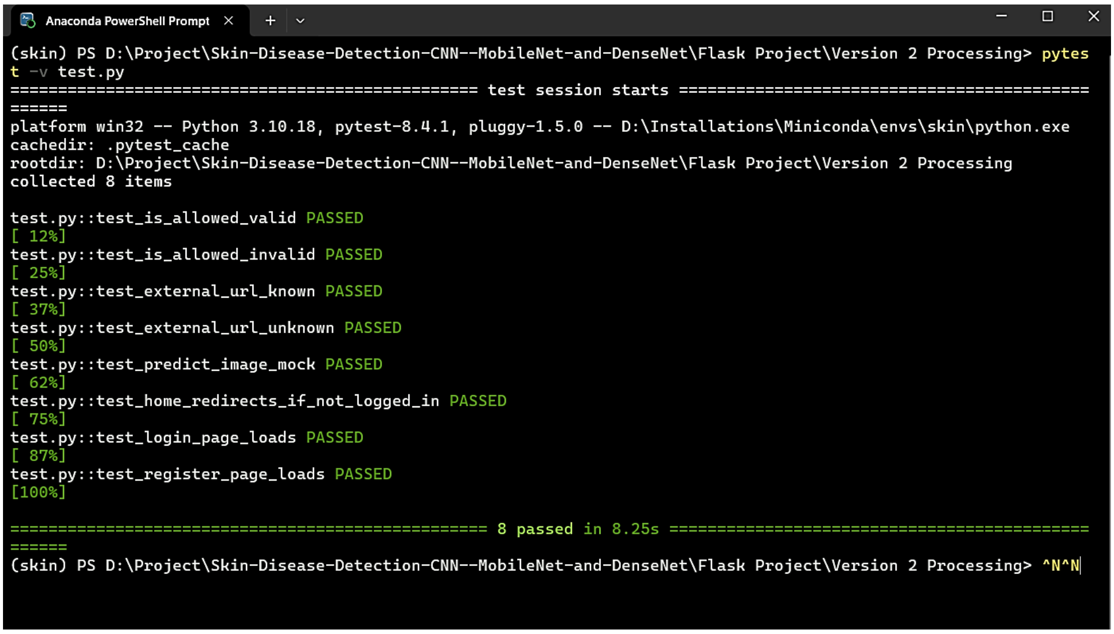


Figure 5.6: Unit Testing Scenarios in AI for Skin Disease Detection

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	User is on upload page	1. Click "Choose	Image of Atopic	Model classifies	Model classified

	upload and prediction		File" 2. Select valid image 3. Click "Predict"	Dermatitis	disease and displays result	image as Atopic Dermatitis
2	Verify error for unsupported file upload	User is on upload page	1. Click "Choose File" 2. Select test.docx 3. Click "Predict"	File test.docx	System shows error message "Invalid file format"	System displayed 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

Evaluation Metrics for AI for Skin Disease Detection

i. **Confusion Matrix**

   The confusion matrix presents a detailed breakdown of the version's predictions as opposed to the actual labels across all skin ailment categories. It lets in us to visualise the performance of our class model, highlighting areas of correct and incorrect predictions for every disorder kind.

ii. The figure five.3 depicts the confusion matrix, which illustrates the category consequences of our version throughout the 9 skin sickness classes in our AI for Skin Disease Detection machine:

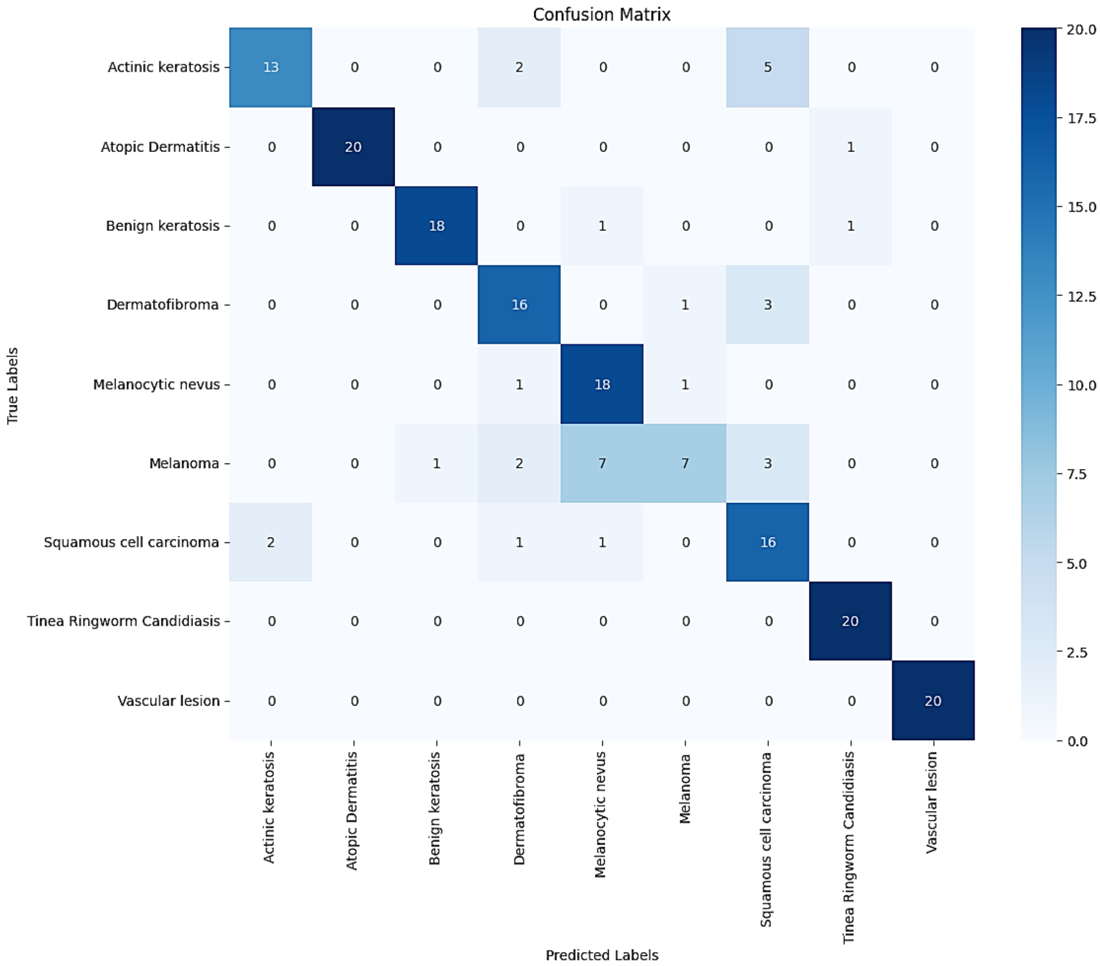


Figure 5.7: Confusion Matrix of AI for Skin Disease Detection

iii. **Accuracy**

The confusion matrix presents a detailed breakdown of the version's predictions as opposed to the actual labels across all skin ailment categories. It lets in us to visualise the performance of our class model, highlighting areas of correct and incorrect predictions for every disorAccuracy is a fundamental metric that measures the overall correctness of our model's predictions throughout all pores and skin sickness classes. It indicates the percentage of correctly labeled snap shots out of the whole pix evaluated. A better accuracy rating signifies higher normal overall performance of the classification model. Based at the evaluation of our model, the accuracy completed is 82%.

iv. Precision

Precision measures the share of correctly predicted images for a particular ailment class out of all pics expected as belonging to that category. In our context, precision shows the version's potential to appropriately perceive a selected pores and skin disorder, minimizing fake positives.

v. Recall

Recall assesses the model's potential to capture all images belonging to a selected sickness category out of all pix that clearly belong to that class. It reflects the version's sensitivity in successfully figuring out diseased cases, thereby minimizing fake negatives.

vi. F1-Score

The F1 score provides a balanced evaluation of the model's overall performance, thinking of both precision and take into account. It offers a complete view of ways nicely our version plays in effectively classifying skin sicknesses. The type record figure beneath provides a detailed review of our model's overall performance, which include precision, keep in mind, F1-score, and support for every pores and skin disease class.

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 5.8: Classification Report of AI for Skin Disease Detection

CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATION

6.1. Conclusion

In cease, the AI for Skin Disease Detection machine is a complete solution designed to help healthcare professionals and individuals in accurately figuring out various skin diseases through computerized picture assessment. By integrating deep learning algorithms along with DenseNet121, MobileNet, and a custom CNN model, the tool efficiently classifies pores and skin disorder pix and offers dependable diagnostic insights.

Throughout the improvement procedure, cautious attention became given to facts preprocessing, version training, and evaluation, making sure the gadget is powerful and performs properly throughout 9 specific skin sickness categories. Extensive sorting out, together with functionality, accuracy, and model widespread overall performance evaluations, validates the machine's reliability, effectiveness, and value in real-worldwide situations.

Overall, the AI for Skin Disease Detection machine represents a treasured device for boosting early analysis, supporting dermatologists, and empowering customers with on hand pores and skin fitness tracking.s

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. Continuous Learning from Feedback: Incorporating mechanisms to learn from user remarks or dermatologist input can help the device adapt and enhance over the years.
3. 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.

4. 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.
5. 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.
6. Regular Model Updates: Continuously updating the model with new information and scientific findings will assist the system remain accurate and relevant as pores and skin ailment patterns evolve.

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.

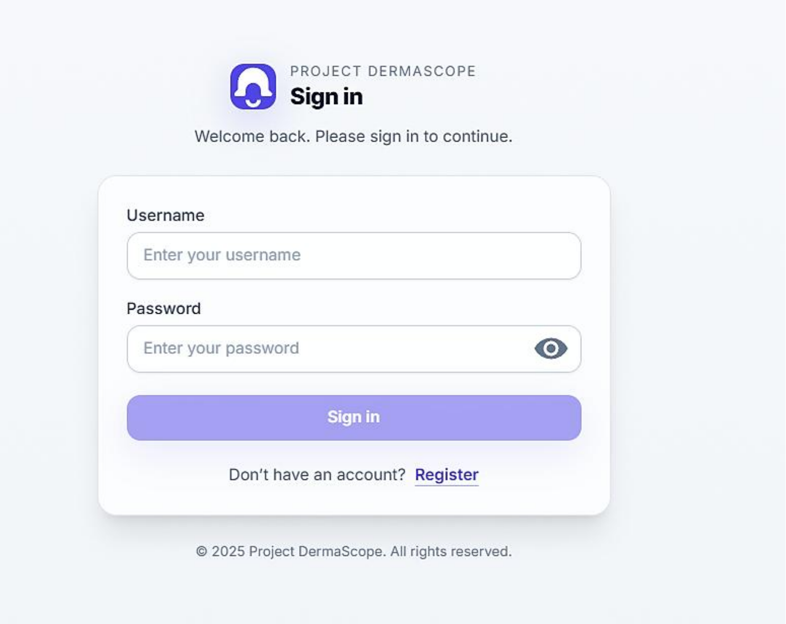
[9] Kassem MA, Hosny KM, Fouad MM. Skin Lesions Classification into Eight Classes for ISIC 2019 Using Deep Convolutional Neural Network and Transfer Learning. IEEE Access. 2020; 8:114822-114832. DOI: 10.1109/ACCESS.2020.3003890.

[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.

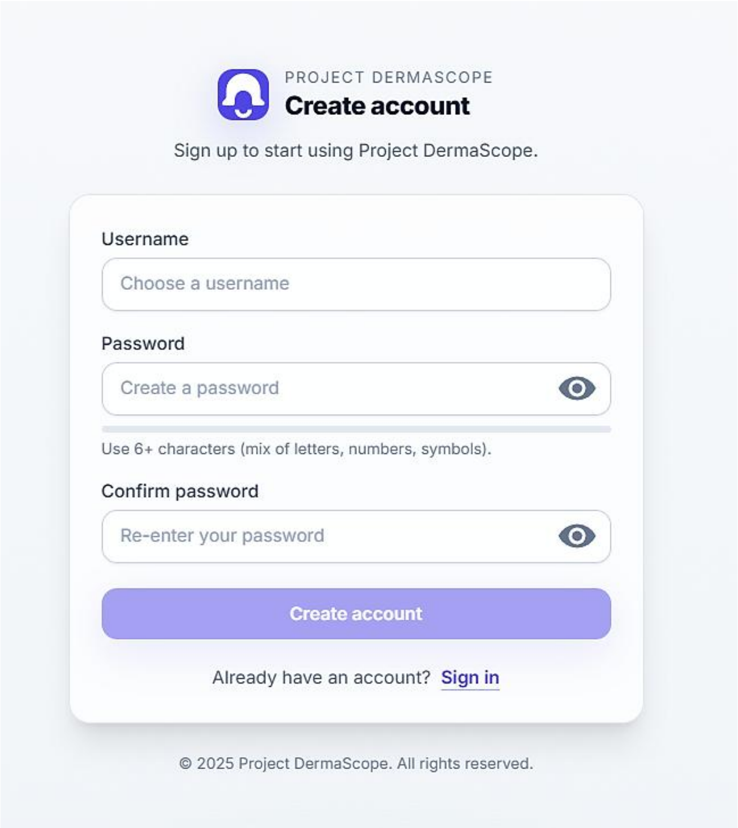
[11] Patil R, Bellary S. Transfer studying based totally system for melanoma type detection. Revue d'Intelligence Artificielle. 2021;35(2):123-a hundred thirty. Https://doi.Org/10.18280/ria.350203

APPENDIX

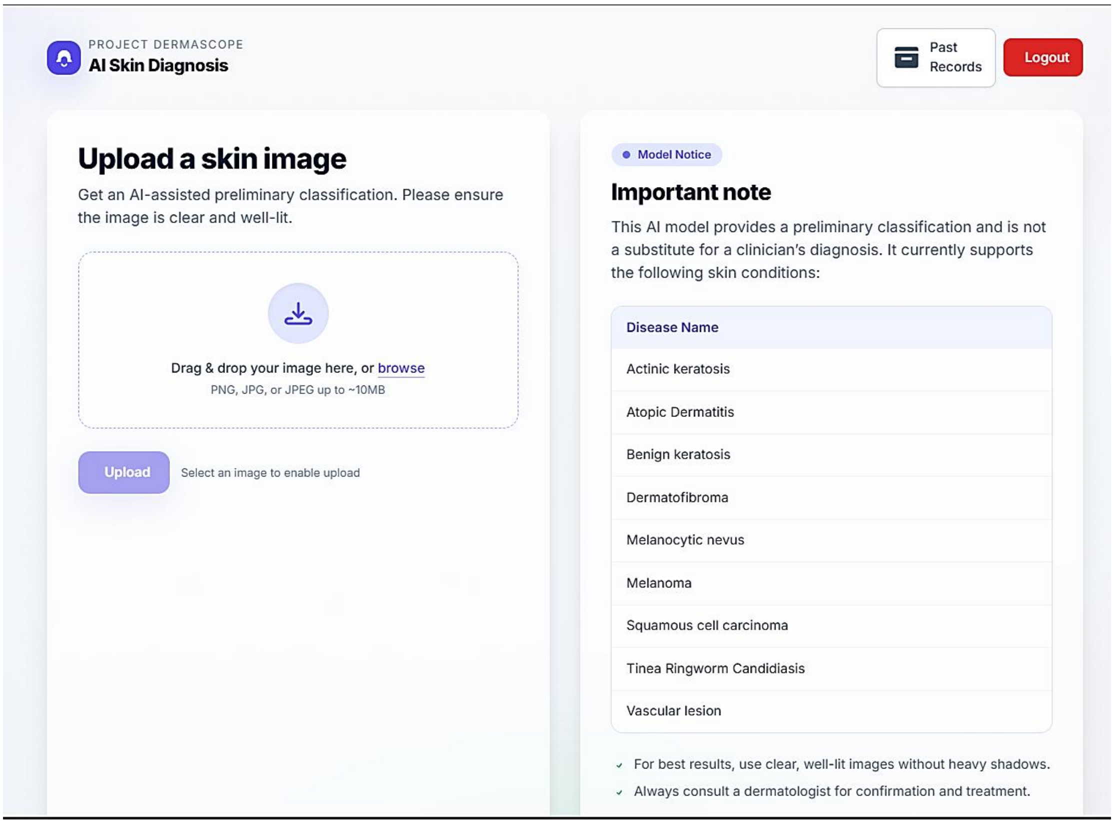
Screenshots:



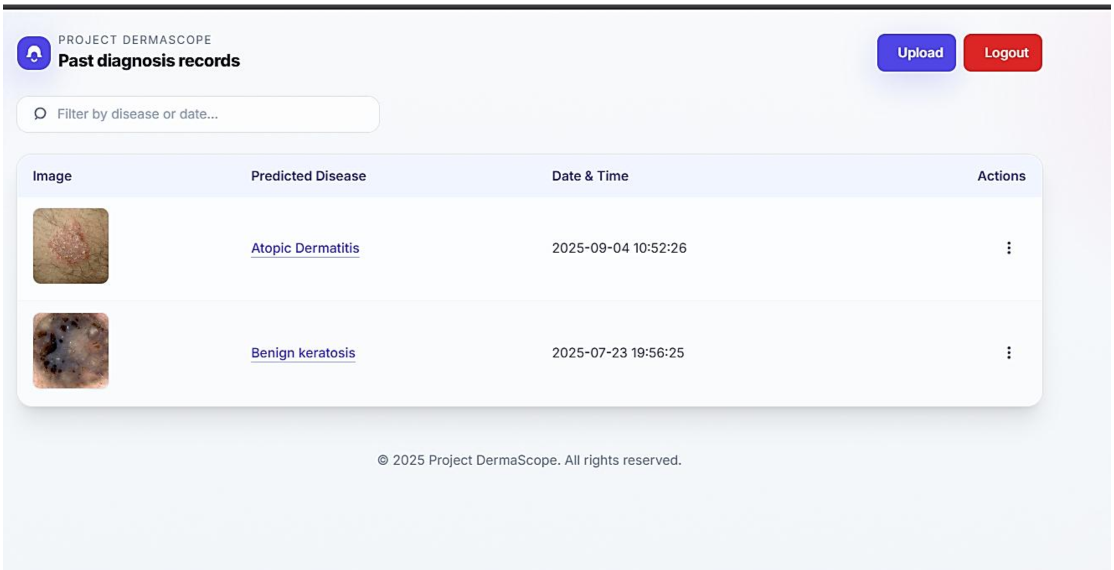
APPENDIX A Sign in Page



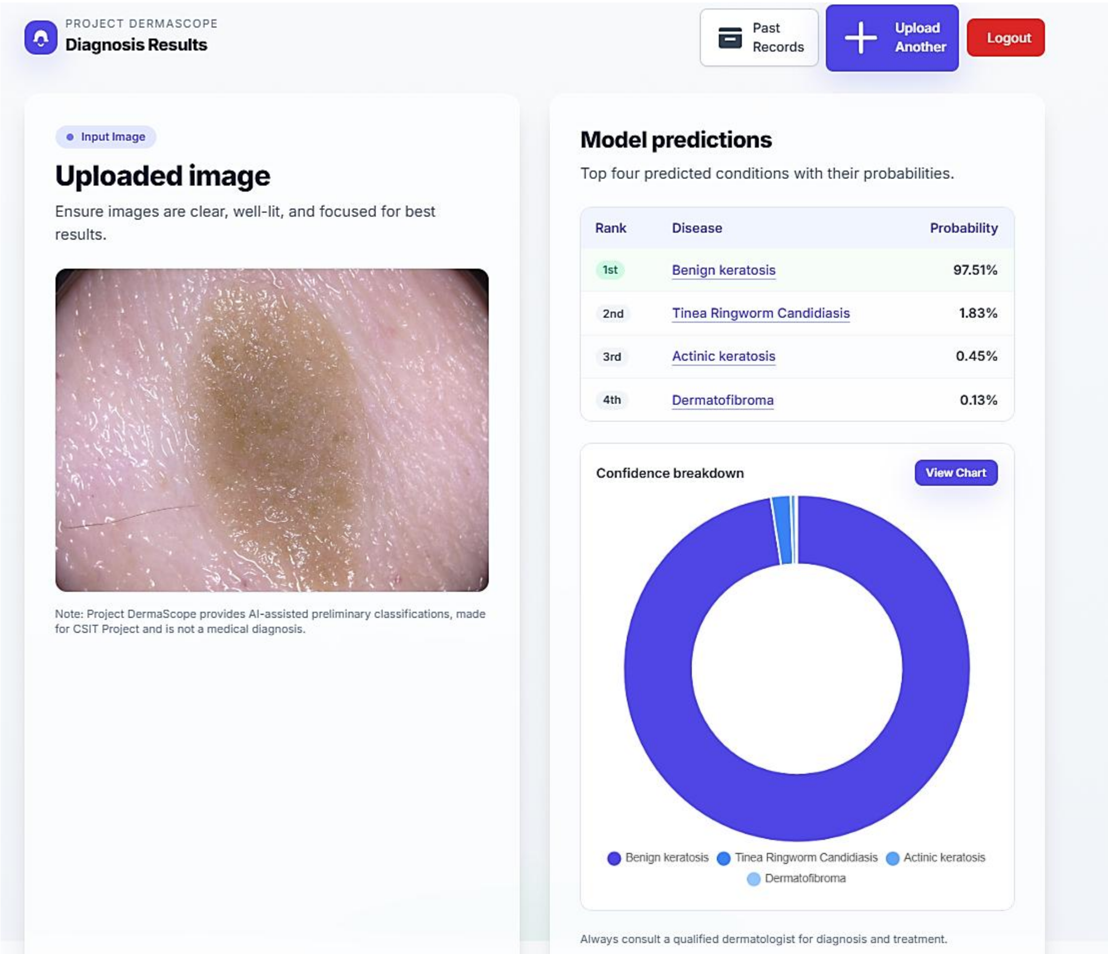
APPENDIX B Register Page



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: [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/Differentiation
Image format/size/
lossy/lossless Color/sharpness/
transformation.

Problems (if any):

orientation of image/lighting.

Tasks for Next Meeting:

20%

Student Name:

Amisha Basnet
Saisa Koirala
Gandesh 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:

3DES/AES

Discussion Topics:

- Use of ImageDataGenerator which resulted 80% accuracy instead of manual image processing
- Use of: Behind PBKDF2-HMAC-SHA256 algorithm for flask

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: Hemu

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: **Supervisor Signature:**

Amisha Basnet

Saisa Basnet

Sandesh Khatiwada

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:

Naresh

08-09-2025