1. INTRODUCTION

Skin diseases are among the most common health conditions affecting people of all age groups across the world. Conditions such as acne, eczema, psoriasis, vitiligo, and skin cancer can significantly impact a person's physical health and quality of life. Early and accurate diagnosis of skin diseases is crucial for effective treatment, prevention of disease progression, and reduction of long-term complications. However, conventional dermatological diagnosis relies primarily on manual visual examination by dermatologists, which can be time-consuming and may vary depending on individual clinical expertise and experience.

With the rapid growth of medical imaging and artificial intelligence, deep learning has emerged as a powerful tool for automated image analysis. Convolutional Neural Networks (CNNs) have demonstrated remarkable performance in extracting meaningful patterns from images and have been widely applied in medical image classification tasks. In dermatology, CNN-based systems can assist clinicians by analyzing skin images and providing preliminary diagnostic insights, thereby improving diagnostic consistency and reducing dependency on subjective visual assessment.

This project focuses on the development of an intelligent skin disease detection system using deep learning techniques. A DenseNet-based convolutional neural network is employed to classify multiple skin disease categories from dermatological images. The model is trained and evaluated using a large, publicly available skin disease image dataset, enabling robust multi-class classification. The system is further integrated into a web-based platform that supports clinical workflows, where dermatologists can upload patient skin images for analysis and patients can book appointments through the system.

By combining deep learning with web-based clinical support, the proposed system aims to enhance the efficiency and reliability of skin disease diagnosis. The project demonstrates the practical application of artificial intelligence in healthcare and highlights its potential to support dermatologists in delivering accurate, consistent, and timely diagnostic decisions.

2. LITERATURE REVIEW

2.1 Literature Review

Paper 1 -Abedin MJ, Nisha JR, Faruk R, Akter S, Hridi TE, Nishi AR. A Study of Images Based Skin Disease Recognition Using Deep Learning. In2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN) 2025 Jul 31 (pp. 1-6). IEEE.

The paper "A Study of Images Based Skin Disease Recognition Using Deep Learning" (2025) presents a comprehensive investigation into the use of deep learning techniques for automated skin disease detection through medical image analysis. The authors highlight that skin diseases are among the most common health issues worldwide and that traditional diagnostic approaches rely heavily on dermatologist expertise, which is often limited in rural and resource-constrained regions. This creates a strong motivation for developing intelligent, automated diagnostic systems capable of providing early and accurate detection to assist healthcare professionals.

To address these challenges, the study evaluates several state-of-the-art Convolutional Neural Network (CNN) architectures, including DenseNet, NASNetMobile, MobileNetV2, EfficientNetB0, InceptionV3, VGG16, and VGG19. Each model was adapted for multi-class classification by modifying the top layers and utilizing transfer learning with pre-trained weights from ImageNet to improve feature extraction. The proposed framework automatically learns discriminative features directly from image data, incorporating custom layers such as Global Average Pooling and Dropout to prevent overfitting.

The experimental phase utilized a curated dataset of approximately 8,000 dermatological images representing ten different skin disease categories along with normal skin, specifically targeting the disease landscape in Bangladesh. Extensive preprocessing techniques, such as image normalization, resizing to 224x224 pixels, and data augmentation—including rotation, flipping, and zooming—were applied to improve data quality and address class imbalance. Among the implemented models, DenseNet achieved the best performance with an average test accuracy of 91.87%, demonstrating superior feature reuse and efficient gradient flow due to its dense connectivity.

The study concludes that image-based deep learning systems can serve as reliable, cost-effective, and scalable diagnostic tools, particularly in regions with limited access to medical specialists. The results indicate that these models significantly enhance diagnostic accuracy compared to traditional methods and can effectively support teledermatology applications. The authors suggest that future work will focus on optimizing these models for real-time inference on mobile devices and collaborating with dermatologists for further clinical validation.

Paper 2 - Chowdary ND, Inturu S, Katta J, Yashwanth C, Kanaparthi NS, Voore S. Skin Disease Detection and Recommendation System Using Deep Learning and Cloud Computing. In2023 8th International Conference on Communication and Electronics Systems (ICCES) 2023 Jun 1 (pp. 1064-1068). IEEE.

The paper "Skin Disease Detection and Recommendation System using Deep Learning and Cloud Computing" (2023) presents an integrated framework designed to automate the diagnosis of skin conditions and provide immediate treatment recommendations. The authors emphasize that skin diseases are a global health priority, often exacerbated by environmental factors and a lack of early detection, particularly in regions with high patient-to-dermatologist ratios. To bridge this gap, the study proposes a system that combines the high accuracy of deep learning with the accessibility and storage capabilities of cloud computing.

To achieve this, the researchers developed a multi-layered methodology centered on a Convolutional Neural Network (CNN) for image classification. The process begins with extensive data collection and preprocessing, including image resizing and data augmentation techniques like rotation and flipping to ensure the model remains robust against variations in lighting and orientation. The architecture incorporates Transfer Learning using pre-trained weights from the ImageNet dataset, which allows the model to leverage existing feature extraction capabilities for more precise identification of lesion patterns. Furthermore, the system is integrated with cloud infrastructure to facilitate real-time processing and a user-friendly recommendation engine that suggests appropriate medications or actions based on the detected disease.

The experimental results demonstrate the effectiveness of the proposed CNN model, which achieved a high classification accuracy of 95%. This performance was evaluated using standard metrics such as precision, recall, and F1-score, confirming the system's ability to reliably distinguish between various bacterial, viral, and fungal skin infections. The model’s success is attributed to its ability to automatically extract complex features from RGB images, surpassing the limitations of traditional machine learning methods that rely on manual feature engineering.

The study concludes that the integration of deep learning with cloud-based services offers a scalable and cost-effective solution for dermatological screening. By providing an automated diagnostic tool that is accessible via mobile or web applications, the system has the potential to reduce diagnostic delays and lower the overall burden on healthcare systems. The authors highlight that this technology is particularly suited for rural and underserved populations, where it can serve as a vital preliminary screening tool for early disease management.

Paper 3- Alruwaili M, Mohamed M. An Integrated Deep Learning Model with EfficientNet and ResNet for Accurate Multi-Class Skin Disease Classification. Diagnostics. 2025 Feb 25;15(5):551.

The paper "An Integrated Deep Learning Model with EfficientNet and ResNet for Accurate Multi-Class Skin Disease Classification" (2025) presents a robust computational approach to automate the detection of various dermatological conditions. The researchers identify skin diseases as a global health concern where traditional diagnostic methods are often hindered by a shortage of specialized dermatologists, particularly in rural or underdeveloped regions. This disparity in healthcare access often leads to delayed diagnoses for critical conditions like melanoma, which has significantly lower survival rates if not caught early. Consequently, the study seeks to develop a high-performance, automated diagnostic system to support clinical decision-making and teledermatology.

To address these challenges, the authors propose a fusion-level deep learning architecture that integrates three distinct convolutional neural networks (CNNs): EfficientNet-B0, EfficientNet-B2, and ResNet50. Each of these models operates independently as a feature extractor, allowing the system to capture a wide variety of detailed patterns from medical images. ResNet50 was chosen for its ability to manage deep network training through residual learning, while the EfficientNet variants were selected for their high parameter efficiency and balanced network scaling. These extracted features are then concatenated into a single, enriched feature vector. The system further incorporates batch normalization to stabilize learning and dropout layers to prevent overfitting, ensuring the model generalizes well to new data.

The experimental phase utilized a large dataset from Kaggle consisting of 27,153 images representing ten different skin disease categories, including eczema, melanoma, and basal cell carcinoma. Before training, the researchers employed extensive data augmentation techniques—such as rotation, flipping, and brightness adjustments—to improve model robustness and address class imbalance issues common in medical datasets. The proposed fusion model demonstrated exceptional performance, achieving an overall accuracy of 99.14%. This result significantly outperformed several standalone state-of-the-art models, such as MobileNet-V3 (85%), EfficientNet-B2 (84%), and InceptionNet-V3 (81%).

The study concludes that the integrated fusion approach is a highly effective tool for multi-class skin disease classification, providing nearly perfect precision and recall across all ten categories. By combining the complementary strengths of different neural network architectures, the model successfully overcomes the limitations of single-model systems, such as misclassification due to visual similarities between different diseases. The authors highlight that while the fusion model requires slightly more execution time than some simpler architectures, its superior accuracy and reliability make it an ideal candidate for real-world clinical applications and automated dermatological diagnosis.

2.2 Literature Review Summary

No	Title	Findings
1	A Study of Images-Based Skin Disease Recognition Using Deep Learning	Proposed a CNN-based image classification system using multiple deep learning architectures. DenseNet achieved the highest accuracy of 91.87%, demonstrating that deep CNN models effectively capture complex visual features of skin lesions and can support early and accurate dermatological diagnosis.
2	Skin Disease Detection and Recommendation System Using Deep Learning and Cloud Computing	Implemented an ensemble CNN model combined with GAN-based data augmentation and cloud-based recommendations. Achieved an overall accuracy of about 97% with prediction capability for up to 40 skin diseases, highlighting the effectiveness of integrating deep learning with cloud computing for scalable diagnosis and personalized treatment guidance.

3	An Integrated Deep Learning Model with EfficientNet and ResNet for Accurate Multi-Class Skin Disease Classification	The study implemented a fusion CNN model combining EfficientNet-B0, EfficientNet-B2, and ResNet50 for multi-class skin disease classification. Trained on 27,153 images across 10 disease categories, the model achieved 99.14% accuracy, outperforming several standalone CNNs. It demonstrates high precision and reliability, making it suitable for automated dermatological diagnosis in clinical applications.

2.3 Findings and Proposals

2.3.1 Findings

After reviewing existing research papers and related works in the field of skin disease detection using deep learning, the following findings were observed:

- Deep learning and computer vision techniques, particularly Convolutional Neural Networks (CNNs), have demonstrated high effectiveness in analyzing dermatological images for automated skin disease classification.
- Several studies confirm that image-based deep learning models can accurately identify and differentiate multiple skin disease categories by learning complex visual features such as texture, color variation, and lesion patterns.
- Transfer learning using pre-trained deep learning architectures significantly improves classification performance, especially when training data is limited, by leveraging knowledge learned from large-scale image datasets.
- Existing research shows that automated skin disease detection systems can assist dermatologists by providing consistent preliminary predictions, thereby reducing dependency on subjective visual examination.
- Although multiple CNN architectures such as VGG, ResNet, MobileNet, and EfficientNet have been explored, comparative analysis indicates that DenseNet offers superior feature reuse and improved gradient flow, leading to better performance in multi-class skin disease classification.

- Most existing systems focus primarily on model accuracy and experimental evaluation, with limited integration into practical clinical workflows, such as structured diagnostic support and scalable deployment.

2.3.2 Proposal

Based on the findings from the literature survey, the proposed system — Deep Learning–Based Skin Disease Detection and Clinical Support System — is designed to address the identified gaps and extend existing research as follows:

- The proposed system will utilize a DenseNet-based deep learning model to automatically classify skin diseases from dermatological images with high accuracy and consistency.
- A multi-class skin disease image dataset will be used to train and evaluate the model, enabling robust and reliable classification across diverse dermatological conditions.
- Image preprocessing and data augmentation techniques will be applied to improve generalization and reduce overfitting during model training.
- The trained DenseNet model will be deployed on a cloud platform, allowing scalable and remote access to the prediction service.
- A Django-based web application will serve as the front-end interface, enabling dermatologists to upload skin images and view classification results, and allowing patients to book dermatology appointments.
- Patient details and appointment information will be managed using a local SQLite database, whereas uploaded skin images and diagnostic prediction results will be securely stored in cloud storage to ensure scalability and long-term accessibility.
- The cloud-stored prediction results and patient records will be managed through an integrated database system to support retrieval, monitoring, and future analysis.
- The performance of the proposed system will be evaluated using standard metrics such as accuracy, precision, recall, and F1-score to validate its effectiveness.
- By integrating deep learning inference, web-based interaction, and cloud storage, the proposed system provides an end-to-end clinical decision support solution for automated skin disease detection.

3.SYSTEM ANALYSIS

3.1 Analysis of dataset

3.1.1 About the Dataset:

The skin disease image dataset used in this study was obtained from Kaggle and consisting of dermatological images organized into 10 distinct skin disease classes. The dataset contained a total of 27,153 images, making it suitable for training and evaluating a multi-class deep learning model.The dataset is approximately **5.58 GB** in size and is organized into separate folders for each disease class, making it suitable for supervised multi-class image classification.

URL:https://www.kaggle.com/datasets/ismailpromus/skin-diseases-image-dataset/data

3.1.2 Explore the dataset

The dataset consists of 10 distinct skin disease categories, with the number of images per class as follows:

- Eczema – 1,677 images
- Atopic Dermatitis – 1,257 images
- Melanoma – 3,140 images
- Basal Cell Carcinoma (BCC) – 3,323 images
- Melanocytic Nevi (NV) – 7,970 images
- Benign Keratosis-like Lesions (BKL) – 2,079 images
- Psoriasis, Lichen Planus, and Related Diseases – 2,055 images
- Seborrheic Keratoses and Other Benign Tumors – 1,847 images
- Tinea, Ringworm, Candidiasis, and Other Fungal Infections – 1,702 images
- Warts, Molluscum, and Other Viral Infections – 2,103 images

Image resolution varied across classes, with widths ranging from 294 to 1024 pixels and heights from 222 to 1128 pixels. Certain classes such as Melanoma contained uniformly sized images (512×512), while others exhibited diverse dimensions, reflecting real-world image acquisition conditions.

Duplicate image analysis revealed a total of 624 duplicate images across all classes, with the highest number observed in the Warts and Viral Infections category. These duplicates were identified and removed during preprocessing to improve data quality and prevent bias during model training.

```
TOTAL DUPLICATE IMAGES: 624

DUPLICATES PER CLASS:
1. Eczema 1677: 104
3. Atopic Dermatitis - 1.25k: 108
9. Tinea Ringworm Candidiasis and other Fungal Infections - 1.7k: 71
10. Warts Molluscum and other Viral Infections - 2103: 233
7. Psoriasis pictures Lichen Planus and related diseases - 2k: 57
8. Seborrheic Keratoses and other Benign Tumors - 1.8k: 23
4. Basal Cell Carcinoma (BCC) 3323: 7
5. Melanocytic Nevi (NV) - 7970: 3
6. Benign Keratosis-like Lesions (BKL) 2624: 14
2. Melanoma 15.75k: 4
```

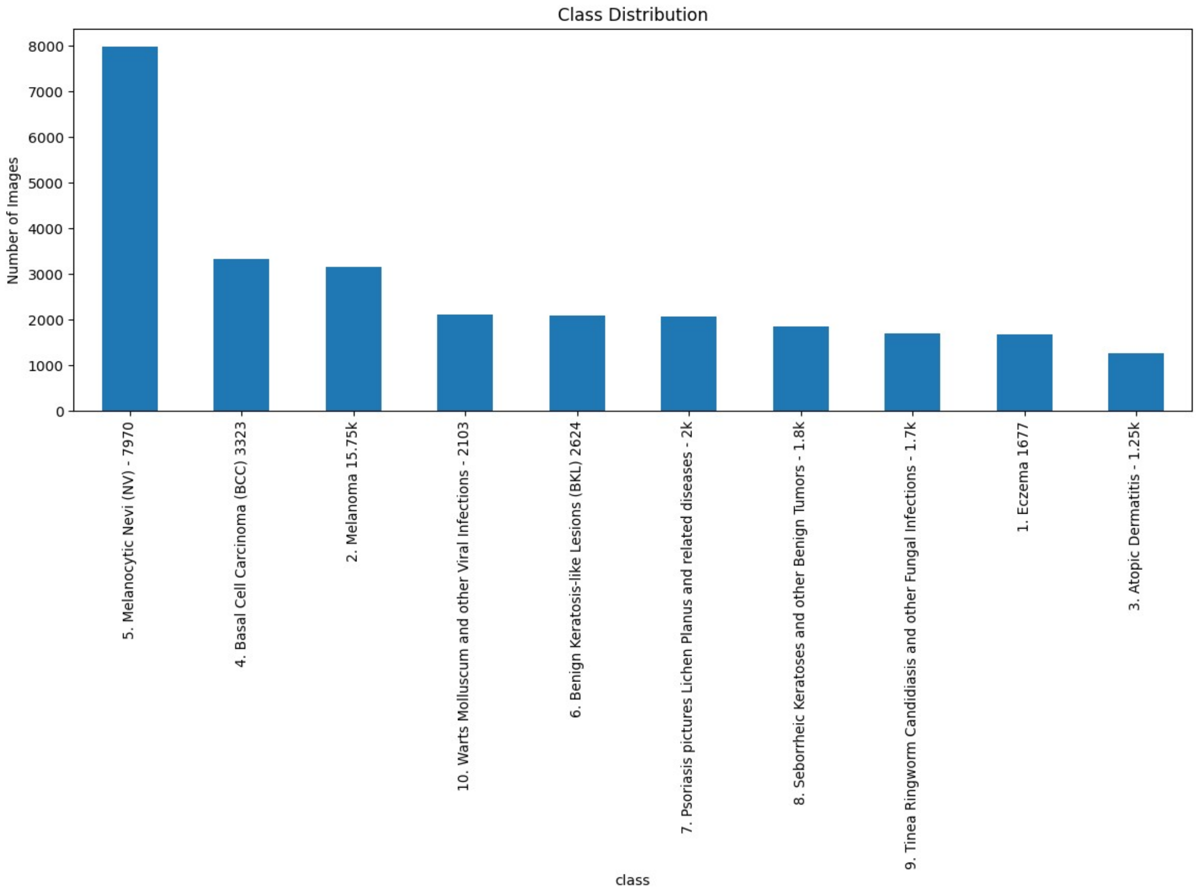
3.1 No. of duplicate images per class

Analysis of the class-wise image distribution revealed a significant imbalance among the skin disease categories in the dataset. Certain classes such as Melanocytic Nevi and Basal Cell Carcinoma contained a substantially higher number of images, while classes like Atopic Dermatitis and Eczema were comparatively underrepresented. The imbalance ratio between the most frequent and least frequent classes was observed to be approximately 6.34, indicating a moderate to high level of class imbalance.

```
IMBALANCE RATIO (max/min): 6.34

CLASS DISTRIBUTION (%)
class
5. Melanocytic Nevi (NV) - 7970                                     29.35
4. Basal Cell Carcinoma (BCC) 3323                                  12.24
2. Melanoma 15.75k                                                  11.56
10. Warts Molluscum and other Viral Infections - 2103                7.75
6. Benign Keratosis-like Lesions (BKL) 2624                          7.66
7. Psoriasis pictures Lichen Planus and related diseases - 2k        7.57
8. Seborrheic Keratoses and other Benign Tumors - 1.8k               6.80
9. Tinea Ringworm Candidiasis and other Fungal Infections - 1.7k     6.27
1. Eczema 1677                                                       6.18
3. Atopic Dermatitis - 1.25k                                         4.63
Name: count, dtype: float64
```

3.2 Class Imbalance



3.3 Class Imbalance visualization

Overall, the dataset's size, diversity in image resolution, and wide range of dermatological conditions provided a strong foundation for training a DenseNet-based deep learning model. Appropriate preprocessing steps, including duplicate removal, resizing, normalization, and class balancing techniques, were applied to enhance model robustness and generalization performance.

3.2 Data Preprocessing

Data preprocessing plays a vital role in deep learning–based skin disease classification, as the quality and balance of dermatological images directly affect model learning and performance. In this project, a systematic preprocessing pipeline was implemented to prepare the skin disease image dataset before training the DenseNet-based convolutional neural network. The preprocessing steps include dataset organization, duplicate image removal, image resizing, dataset splitting, and class imbalance handling.

3.2.1 Data cleaning

```
IMAGE COUNT PER CLASS
class
5. Melanocytic Nevi (NV) - 7970                                  7970
4. Basal Cell Carcinoma (BCC) 3323                               3323
2. Melanoma 15.75k                                               3140
10. Warts Molluscum and other Viral Infections - 2103            2103
6. Benign Keratosis-like Lesions (BKL) 2624                      2079
7. Psoriasis pictures Lichen Planus and related diseases - 2k    2055
8. Seborrheic Keratoses and other Benign Tumors - 1.8k           1847
9. Tinea Ringworm Candidiasis and other Fungal Infections - 1.7k 1702
1. Eczema 1677                                                   1677
3. Atopic Dermatitis - 1.25k                                     1257
Name: count, dtype: int64
```

3.4. No.of Images Per class

Duplicate Image Removal

Duplicate images can bias training and lead to misleading performance results. To address this issue, duplicate image detection and removal were performed as an initial preprocessing step.

● Perceptual hashing (pHash) was applied to each image using the imagehash library.

● Images producing identical hash values were considered duplicates.

● Only one unique instance was retained, and duplicate files were permanently removed from the dataset.

This process reduced redundancy and prevented the model from learning repeated patterns.

```python
# ======================================================
# REMOVE DUPLICATES
# ======================================================
hashes = {}
removed = 0

print("\nRemoving duplicate images...")
for img_path in all_images:
    try:
        with Image.open(img_path) as img:
            img = img.convert("RGB")
            h = imagehash.phash(img)
            if h in hashes:
                os.remove(img_path)
                removed += 1
            else:
                hashes[h] = img_path
    except:
        pass

print(f" Removed duplicate images: {removed}")
```

3.5 Removing Duplicates

Image Resizing

To ensure compatibility with the DenseNet architecture, all images were resized to a fixed input resolution.

● Each image was resized to 224 × 224 pixels, matching the input size required by DenseNet121.

● The Lanczos resampling method was used to preserve important visual details during resizing.

● Images were converted to RGB format to maintain consistency across the dataset.

● Resizing was performed after duplicate removal to avoid unnecessary computation.

Uniform image dimensions improved computational efficiency and ensured stable batch processing during training.

```
# =====================================================
# RESIZE IMAGES
# =====================================================
def resize_image(img_path):
    try:
        with Image.open(img_path) as img:
            img = img.convert("RGB")
            img = img.resize(IMG_SIZE, Image.LANCZOS)
            img.save(img_path)
    except:
        pass

print("\nResizing images...")
with ThreadPoolExecutor(max_workers=4):
    list(map(resize_image, all_images))

print(" Image resizing completed")
```

3.6 Resizing Images

Handling Class Imbalance

The dataset exhibited noticeable class imbalance, with certain skin disease categories having significantly more images than others. To mitigate this issue, class weighting was applied during training.

● Class frequencies were computed from the training dataset.

● Balanced class weights were calculated using the compute_class_weight function from Scikit-learn.

● Higher weights were assigned to minority classes, while lower weights were assigned to majority classes.

● These weights were later used during model training to penalize misclassification of underrepresented classes more heavily.

Handling class imbalance improved fairness and enhanced the model’s ability to correctly classify all skin disease categories.

```
# =====================================================
# CLASS WEIGHTS
# =====================================================
labels = []
for _, y in train_ds:
    labels.extend(np.argmax(y.numpy(), axis=1))

labels = np.array(labels)

class_weights = compute_class_weight(
    class_weight="balanced",
    classes=np.unique(labels),
    y=labels
)

class_weight_dict = dict(enumerate(class_weights))

print("\n===== CLASS WEIGHTS =====")
for cls, w in zip(class_names, class_weights):
    print(f"{cls:<40} : {w:.3f}")
```

3.7 Handling Imbalance

Data Distribution:

To ensure effective training and unbiased evaluation of the deep learning model, the skin disease image dataset is divided into three standard subsets:

- **Training Set:** Approximately 70% of the images are used for model learning and feature extraction.
- **Validation Set:** Around 15% of the images are used to tune hyperparameters and monitor model performance during training.
- **Testing Set:** The remaining 15% of the images are reserved for final evaluation of themodel’s classification accuracy and generalization ability.

3.2.3 Analysis of Class Variables

The Skin Disease Image Dataset is structured around ten categorical class variables, each representing a distinct dermatological condition. These class variables form the foundation of the multi-class classification task and are designed to capture a wide range of skin diseases with varying visual characteristics, severity levels, and prevalence rates.

The dataset exhibits a non-uniform class distribution, reflecting real-world medical data where certain skin conditions are more common than others. This diversity ensures that the model learns both frequently occurring diseases and comparatively rare but clinically significant conditions.

- Melanocytic Nevi (NV) – 7,970 images, representing the most commonly observed benign skin condition.
- Basal Cell Carcinoma (BCC) – 3,323 images, providing substantial data for learning malignant patterns.
- Melanoma – 3,140 images, emphasizing early detection of a life-threatening condition.
- Eczema – 1,677 images
- Atopic Dermatitis – 1,257 images
- Psoriasis, Lichen Planus and related diseases – 2,055 images
- Benign Keratosis-like Lesions (BKL) – 2,079 images
- Seborrheic Keratoses and other Benign Tumors – 1,847 images
- Warts, Molluscum and other Viral Infections – 2,103 images
- Tinea, Ringworm, Candidiasis and other Fungal Infections – 1,702 images

Due to this imbalance, class weighting techniques were applied during model training to reduce bias toward majority classes and improve performance on underrepresented categories.

3.3 Data Visualization

Data visualization plays a vital role in understanding the characteristics, distribution, and quality of dermatological image data before training a deep learning model. In this project, visualization techniques were employed to analyze the skin disease image dataset, verify data consistency, examine class distribution, and assess the effectiveness of preprocessing steps such as duplicate removal, image resizing, and class imbalance handling. Visual exploration helped identify dataset bias, image quality variations, and potential issues that could affect model performance.

Image dimension visualizations were used to examine variations in width and height across different skin disease classes. These plots revealed inconsistencies in image resolution across the dataset, which justified resizing all images to a fixed dimension of 224 × 224 pixels to ensure uniform input size for the DenseNet-based classification model. In addition, sample image grids were visualized for each class to observe distinctive visual patterns, including texture, color variations, lesion boundaries, and shape characteristics, aiding in better understanding of inter-class similarities and differences.

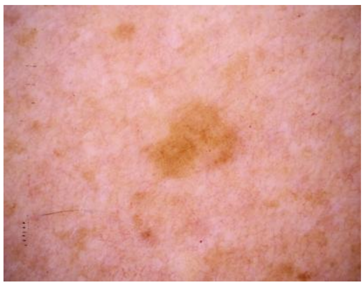
```
                                                    height
                                                       min   max        mean
class
1. Eczema 1677                                         222  1080  530.151461
10. Warts Molluscum and other Viral Infections ...     222  1128  415.441274
2. Melanoma 15.75k                                     512   512  512.000000
3. Atopic Dermatitis - 1.25k                           222   720  380.542562
4. Basal Cell Carcinoma (BCC) 3323                     450  1024  935.213963
5. Melanocytic Nevi (NV) - 7970                        450   995  513.852572
6. Benign Keratosis-like Lesions (BKL) 2624            450  1024  764.195286
7. Psoriasis pictures Lichen Planus and related...     222  1080  505.290024
8. Seborrheic Keratoses and other Benign Tumors...     222   720  494.620466
9. Tinea Ringworm Candidiasis and other Fungal ...     222   720  518.446533
```

3.8 Height Measurements

```
IMAGE SIZE STATISTICS (per class)
                                                    width                  \
                                                      min   max        mean
class
1. Eczema 1677                                        294   720  615.471079
10. Warts Molluscum and other Viral Infections ...    294   720  549.552544
2. Melanoma 15.75k                                    512   512  512.000000
3. Atopic Dermatitis - 1.25k                          294   720  468.842482
4. Basal Cell Carcinoma (BCC) 3323                    600  1024  958.415889
5. Melanocytic Nevi (NV) - 7970                       576  1024  701.004517
6. Benign Keratosis-like Lesions (BKL) 2624           600  1024  832.088504
7. Psoriasis pictures Lichen Planus and related...    294   720  598.035036
8. Seborrheic Keratoses and other Benign Tumors...    294   720  655.793720
9. Tinea Ringworm Candidiasis and other Fungal ...    294   720  649.458872
```

3.9 Width Measurements





3.10 Images from 10 different classes

3.4 Analysis of Architecture

3.4.1 DenseNet-169 Architecture

In this project, a DenseNet-169–based Convolutional Neural Network (CNN) architecture is employed for multi-class skin disease classification. DenseNet-169 is an advanced deep learning architecture that extends the DenseNet family by increasing network depth while maintaining efficient parameter usage through dense connectivity. The architecture is particularly well suited for medical image analysis, where capturing subtle visual differences is critical.

Unlike traditional CNN architectures where each layer receives input only from the immediately preceding layer, DenseNet-169 introduces dense connections, allowing each layer to receive feature maps from all previous layers within the same dense block. This design enhances feature reuse, strengthens gradient flow, and mitigates the vanishing gradient problem, enabling effective training of very deep networks.

Key Points about DenseNet-169

a) **CNN-Based Feature Learning:**

● DenseNet-169 is built upon convolutional layers that automatically learn hierarchical features such as edges, textures, color distributions, and lesion patterns from dermatological images.

● Lower layers capture fine-grained visual details, while deeper layers learn complex semantic representations essential for distinguishing between similar skin diseases.

● The architecture eliminates the need for manual feature engineering by learning discriminative features directly from raw image data.

b) **Dense Connectivity Mechanism:**

● In DenseNet-169, each layer is connected to all preceding layers within a dense block, forming direct information pathways throughout the network.

● These dense connections encourage extensive feature reuse and reduce redundancy across layers.

● Improved gradient propagation enables stable and efficient training even with increased network depth.

c) **Backbone Feature Extraction:**

● DenseNet-169 serves as the backbone feature extractor in the proposed system.

● It consists of multiple dense blocks and transition layers that progressively extract high-level feature representations while reducing spatial resolution.

● Transfer learning is applied by initializing the model with pretrained ImageNet weights, allowing the network to leverage previously learned visual knowledge and adapt effectively to skin disease classification.

d) **Classification Head:**

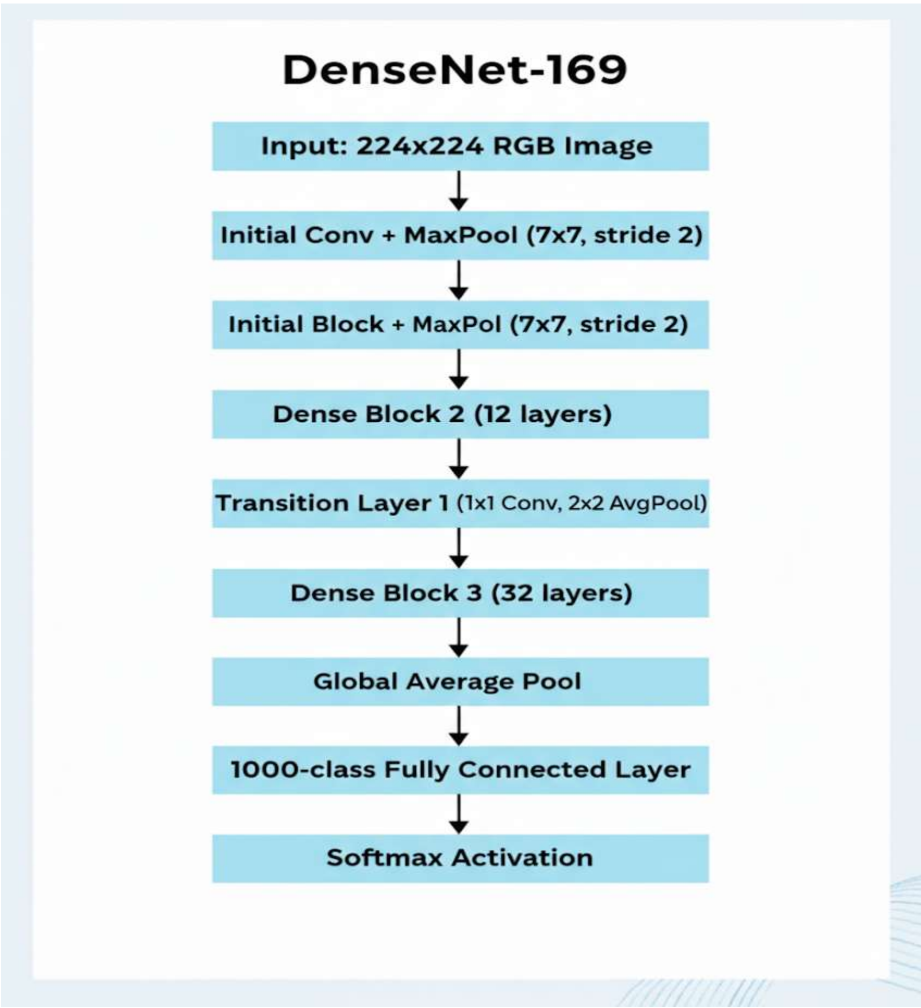
After feature extraction, a custom classification head is attached to the DenseNet-169 backbone, consisting of:

- Global Average Pooling to reduce feature dimensionality
- Fully connected (Dense) layers for learning class-specific features
- Softmax activation to generate probability scores for each skin disease category

This enables accurate multi-class classification across diverse dermatological conditions.

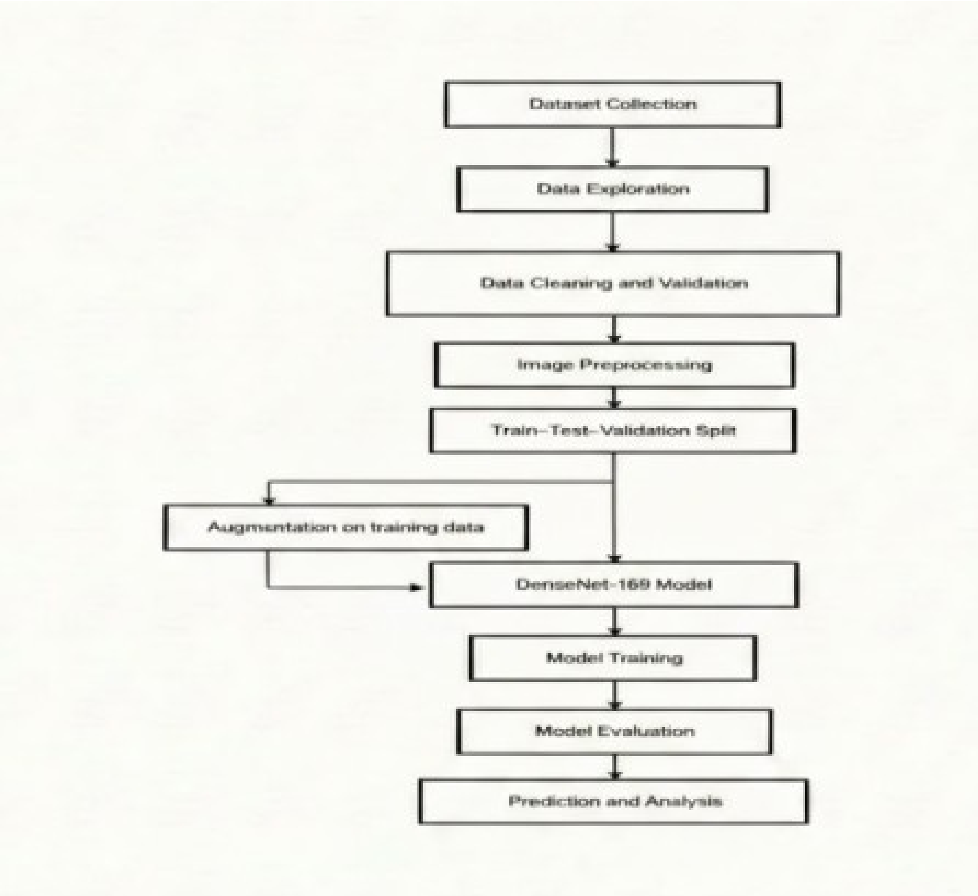
e) **End-to-End Training Strategy:**

- The DenseNet-169 model is trained end-to-end, jointly optimizing feature extraction and classification components.
- Selected layers of the backbone can be fine-tuned to improve performance while preventing overfitting.
- This training strategy ensures high accuracy, robustness, and generalization capability for real-world dermatological image analysis.



3.10 DenseNet-169 Architecture

3.5 Project Pipeline



3.11 Project Pipeline

Phase 1: Project Initiation

During this initial stage, the primary objective was to define the project scope and identify essential features required for a functional skin disease detection and clinical support system. Key stakeholders, including dermatologists, patients, and healthcare administrators, were identified to ensure the system met real-world clinical needs. The technology stack was finalized during this phase, selecting Python as the primary language, Django for the web framework, DenseNet for deep learning-based skin disease classification, SQLite for local data persistence, and cloud storage for handling uploaded dermatological images and prediction results.

Phase 2: System Design

This phase focused on the architectural layout of the application and its underlying database structure. The database design included defining essential schemas:

- tbl_patient for storing patient details, including name, contact info, and medical history.
- tbl_doctor for managing dermatologist profiles and credentials.
- tbl_booking for managing appointments between patients and doctors.

Simultaneously, the UI/UX design process involved creating wireframes for the web interface, including the patient registration portal, image upload interface, diagnostic result display, and appointment booking system, ensuring ease of use for both medical staff and patients.

Phase 3: Implementation

The implementation phase involved the development of all functional system components. Backend development included initializing the Django project, configuring settings, and creating the database models for tbl_patient, tbl_doctor, and tbl_booking. User management features were implemented for secure patient and doctor logins.

A critical part of this phase was integrating the DenseNet model through a script (model.py), connecting the web application to the deep learning inference engine. Cloud storage integration was implemented to securely store uploaded dermatological images and model-generated diagnostic results, ensuring scalable and long-term access.

Phase 4: Testing & Validation

To ensure reliability, the system undergoes rigorous testing and validation of its classification and booking functionalities. Unit testing was applied to individual Django models, views, and cloud storage operations, while integration testing confirmed the end-to-end flow from image upload to AI analysis, database updates, and appointment management. The DenseNet model was validated against a testing subset of dermatological images, confirming high accuracy, precision, recall, and F1-score across multiple skin disease classes.

Phase 5: Deployment & Maintenance

The final phase involves deploying the Django web application and DenseNet model onto a cloud platform for scalable remote access.

3.6 Feasibility Analysis

A feasibility study is conducted to evaluate the practicality of the proposed system and to determine whether the project can be successfully developed and implemented. It helps in identifying the strengths and limitations of the proposed solution, the availability of required resources, and the overall viability of the system.

The feasibility of the proposed Skin Disease Detection and Clinical Support System is evaluated under the following categories:

- Technical Feasibility
- Economic Feasibility
- Operational Feasibility

3.6.1 Technical Feasibility

The proposed system is technically feasible as all the required technologies, tools, and frameworks are readily available and well-supported. The system is developed using Python for deep learning model implementation and Django for web application development. Python provides extensive libraries such as TensorFlow, Keras, NumPy, and OpenCV, which simplify the development of deep learning–based image classification systems.

The skin disease classification model is implemented using a DenseNet-based Convolutional Neural Network, which can be efficiently trained using publicly available datasets and executed on platforms such as Google Colab for GPU acceleration. This eliminates the need for high-end local hardware during model training.

For the web component, Django supports secure authentication, role-based access (doctor and patient modules), and smooth integration with machine learning models. SQLite is used for managing patient details and appointment records, while cloud storage is utilized to store uploaded images and diagnostic prediction results. These technologies are reliable, scalable, and widely used, making the system technically feasible and easy to maintain.

3.6.2 Economic Feasibility

The proposed system is economically feasible as it does not require expensive hardware or proprietary software. Development is carried out using open-source tools and frameworks such as Python, Django, TensorFlow, and Google Colab, which significantly reduce the overall cost.

The system requires only a standard computer with internet access for development and testing. Model training can be performed using free or low-cost cloud resources, and deployment can be done on affordable cloud platforms. The use of SQLite for local data management further minimizes database costs.

Since the system automates skin disease analysis and reduces dependency on repeated manual diagnosis, it can also help reduce operational costs in the long run. Therefore, the project can be developed and maintained within a reasonable budget, making it economically viable.

3.6.3 Operational Feasibility

Operational feasibility evaluates how effectively the proposed system can be used by its intended users. The system is designed to be user-friendly and workflow-oriented, making it suitable for clinical environments.

Patients can easily book appointments through the web interface, while dermatologists can upload skin images during consultation and view prediction results generated by the deep learning model. The interface is simple and does not require advanced technical knowledge to operate.

The automation of image analysis supports dermatologists by providing consistent preliminary predictions, reducing manual effort and diagnostic variability. Since the application is web-based, it can be accessed from any authorized device, improving usability and acceptance. Overall, the system is operationally feasible and aligns well with real-world clinical workflows.

3.7 System Environment

The system environment specifies the required hardware and software configuration for the development and execution of the proposed system. These specifications ensure that the system performs efficiently and meets user requirements.

3.7.1 Software Environment

The Skin Disease Detection and Clinical Support System is developed using a combination of programming languages, frameworks, and tools that together form the complete software environment of the project.

• Programming Language:- Python is the primary programming language used for implementing the deep learning model, backend logic, and data processing. Its extensive support for machine learning libraries and ease of integration with web frameworks makes it suitable for developing AI-based medical applications.

• Web Framework :- Django is used as the backend web framework for developing the clinical web application. It manages URL routing, user authentication, role-based access (doctor and patient modules), backend logic, and interaction with the database. Django also facilitates secure handling of medical data and smooth integration of the deep learning model into the web system.

• Database :- SQLite is used as the local database to store structured data such as patient details, doctor information, appointment records, and authentication credentials. SQLite is lightweight, easy to configure, and well-suited for academic and prototype-level clinical applications.

• Cloud Storage :- Cloud storage is used to store uploaded skin images, diagnostic prediction results, and model-related outputs. Storing these components in the cloud ensures data availability, scalability, and secure access while reducing the load on the local database. This separation allows SQLite to manage structured data efficiently while cloud storage handles large image and result files.

• AI / Deep Learning Environment

a) DenseNet (CNN Architecture) :-A DenseNet-based Convolutional Neural Network is used for skin disease classification. DenseNet improves feature reuse and gradient flow, making it effective for multi-class dermatological image analysis.

b) TensorFlow / Keras :-These frameworks are used to build, train, and evaluate the DenseNet model. They provide high-level APIs for deep learning model development and experimentation.

c) Google Colab :-Google Colab is a cloud-based platform used for training and testing the deep learning model. It provides free access to GPU resources, which significantly speeds up model training.

In this project, Google Colab is used for:

- Dataset preprocessing and organization
- Training the DenseNet model
- Model evaluation and experimentation
- Visualization of accuracy and loss graphs

Google Colab also integrates seamlessly with Google Drive, making it convenient to store datasets and trained models.

• Frontend Technologies

a) HTML (HyperText Markup Language) :-HTML is used to define the structure of web pages such as login pages, dashboards, appointment booking forms, and result display pages.

b) CSS (Cascading Style Sheets) :-CSS is used to style the web pages and provide a clean, consistent, and user-friendly interface suitable for clinical use.

c) Bootstrap :-Bootstrap is used to create responsive web pages that adapt to different screen sizes, ensuring accessibility on desktops, tablets, and mobile devices.

d) JavaScript :-JavaScript is used to add interactivity to the web application, such as form validation, dynamic updates, alerts, and improved user experience.

• IDE / Code Editor

Visual Studio Code (VS Code) :-

VS Code is used as the primary development environment for writing, testing, and managing the project code. It supports Python and Django development with debugging and extension support.

• Version Control :- Git is used for version control to track code changes, manage project versions, and ensure safe backup. GitHub is used to store the project repository and support collaboration and code management.

3.7.2 Python Packages in Virtual Environment

A virtual environment is used to manage project dependencies independently of the system-wide Python installation. This ensures consistency and avoids compatibility issues.

The important Python packages used in this project include:

- Django – For backend web development and application management
- TensorFlow / Keras – For building and training the DenseNet deep learning model
- NumPy – For numerical computations
- Pandas – For dataset handling and preprocessing
- OpenCV – For image processing tasks
- scikit-learn – For model evaluation metrics such as accuracy, precision, recall, and F1-score
- Matplotlib – For plotting training and evaluation graphs

3.7.3 Hardware Requirements

The hardware configuration required for developing and testing the proposed system is as follows:

- Processor : Intel Core i5 or higher
- Memory : Minimum 8 GB RAM (16 GB recommended)
- Disk Space : 40 GB or greater
- Internet Connectivity : Required for cloud storage access, model training, and deployment

The above configuration is sufficient for smooth development, training, and execution of the proposed skin disease detection system.