SKIN DISEASE ANALYSIS

Submitted by

Md.Arafat Rahman
Reg. No: 200103020013

Supervised by

Ayon Dey
Lecturer
Department of Computer Science and Engineering

BACHELOR OF SCIENCE (Engg.)
IN
COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering
North East University Bangladesh (NEUB)
Sylhet, Bangladesh

February, 2025

SKIN DISEASE ANALYSIS

A work submitted to the Department of Computer Science and Engineering, North East University Bangladesh, for partial fulfillment of the requirements for the degree of BACHELOR OF SCIENCE (Engg.)in Computer Science and Engineering

Submitted by

Md.Arafat Rahman
Reg. No: 200103020013

Supervised by

Ayon Dey
Lecturer
Department of Computer Science and Engineering

February, 2025

Recommendation Letter from Thesis/Project Supervisor

Md. Arafat Rahman, whose thesis/project entitled **"Skin Disease Analysis"**, is under my supervision and agrees to submit for examination.

Signature of the Supervisor:

Ayon Dey
Lecturer
Department of Computer Science and Engineering

Qualification Form of BACHELOR OF SCIENCE (Engg.) degree

This thesis titled, **"Skin Disease Analysis"**, submitted by Md.Arafat Rahman (ID: 200103020013), Session: April 2020, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of BACHELOR OF SCIENCE (Engg.) in Computer Science and Engineering on 13^{th}February, 2025.

BOARD OF EXAMINERS

Arif Ahmed
Professor **Chairman**
Department of Computer Science and Engineering

Ayon Dey
Lecturer **Supervisor**
Department of Computer Science and Engineering

Candidate's Declaration

This is to certify that the work presented in this thesis entitled, "Skin Disease Analysis", is the outcome of the research carried out by -under the supervision of Ayon Dey, Lecturer, Department of Computer Science and Engineering, North East University Bangladesh (NEUB), Sylhet, Bangladesh.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma, or other qualifications.

Signature of the Candidate

Md.Arafat Rahman
ID: 200103020013

Contents

List of Figures

List of Tables

List of Abbreviations

Adam	Adaptive Moment Estimation
AI	Artificial Intelligence
CB	Callback
CNN	Convolutional Neural Network
DL	Deep Learning
EffNet-B2	EfficientNet-B2
ES	Early Stopping
GAP	Global Average Pooling
L2	L2 Regularization
LR	Learning Rate
ML	Machine Learning
RLR	ReduceLROnPlateau
SGD	Stochastic Gradient Descent

Abstract

Skin diseases affect millions of people worldwide, with early detection being crucial for effective treatment. This project aims to develop a reliable and efficient system for skin disease analysis using Convolutional Neural Networks (CNNs). Leveraging the power of deep learning, the system is designed to classify various skin conditions based on image data, providing an automated tool that can assist dermatologists in diagnosing diseases. The project involved collecting and preprocessing a dataset of skin images, representing a wide range of skin conditions. A CNN model was then trained and fine-tuned to classify these images into corresponding disease categories. Various architectures and hyperparameters were tested to optimize the model's accuracy and performance. The results demonstrated the effectiveness of CNNs in skin disease classification, with the model achieving a high accuracy rate. This project underscores the potential of AI-driven solutions in the medical field, particularly in augmenting the capabilities of healthcare professionals. Future work could focus on expanding the dataset, improving model robustness, and exploring real-time implementation in clinical settings.

Keywords: Skin disease analysis, Convolutional Neural Networks (CNNs), Deep learning,Image classification,Automated diagnosis, Medical AI, Model accuracy, Dataset preprocessing.

Chapter 1

Introduction

Skin diseases are among the most common health issues globally, affecting individuals of all ages and demographics. These conditions can range from mild irritations to life-threatening diseases such as melanoma. Early and accurate diagnosis is critical for effective treatment and management of these diseases. However, the traditional method of diagnosing skin conditions relies heavily on visual inspection by dermatologists, which can be subjective and time-consuming. The increasing prevalence of skin diseases, coupled with a shortage of dermatologists in many regions, underscores the need for automated and reliable diagnostic tools. Recent advancements in artificial intelligence, particularly in the field of deep learning, have opened new avenues for medical diagnostics. Convolutional Neural Networks (CNNs), a type of deep learning architecture, have shown remarkable success in image recognition tasks. By mimicking the way the human brain processes visual data, CNNs can learn to identify patterns and features in images that may be difficult for the human eye to discern. This makes them an ideal tool for analyzing medical images, including those related to skin diseases. This project focuses on developing a CNN-based system for the classification and analysis of various skin diseases. The goal is to create an automated tool that can assist dermatologists in diagnosing skin conditions more efficiently and accurately. The system is trained on a dataset of labeled skin images, enabling it to learn the distinguishing features of different skin diseases. Through this approach, the project aims to enhance the diagnostic process, reduce the burden on healthcare professionals, and improve patient outcomes. In this report, we will discuss the methodology used to develop the CNN model, including data collection and preprocessing, model architecture, training process, and evaluation metrics. We will also present the results of our experiments, highlighting the model's accuracy and potential areas for improvement. Finally, we will explore the implications of this technology in clinical settings and suggest direc-

tions for future research. In summary, The project tackles the need for automated skin disease diagnosis, addressing the global prevalence of these conditions and the limitations of traditional diagnostic methods. By employing ResNet, a Convolutional Neural Network (CNN) architecture known for its depth and effectiveness in image recognition tasks, the project aims to develop an accurate and efficient system for classifying skin diseases. This report details the methodology, model development, and results, highlighting the potential of AI-driven solutions like ResNet to enhance dermatological diagnostics and improve patient care.

1.1 Disease Recognition

A CNN model recognizes skin diseases by analyzing images of the skin. It learns to identify patterns, textures, and features associated with different diseases through multiple layers of filters. Each layer extracts increasingly complex details from the image, allowing the model to differentiate between various conditions and accurately classify the disease based on the learned features.

1.1.1 Skin Disease Recognition

A model like ResNet recognizes real and fake skin diseases by analyzing image features through its deep layers. It processes the image in stages, with each layer identifying specific patterns and details. ResNet's skip connections help it learn complex features without losing important information. By comparing these features to what it has learned from a large dataset, the model can distinguish between genuine skin disease patterns and anomalies that might indicate a fake or misclassified condition.

1.1.2 Skin diseases (fungal,bacterial and other types) Recognition

For recognizing different types of skin diseases, such as fungal, bacterial, and other conditions, a CNN model is trained on a large dataset of labeled skin images. During training, the CNN learns to identify specific features unique to each type of disease—like the texture, color, and pattern associated with fungal infections, bacterial infections, and other skin conditions. The model's layers progressively extract these features, with deeper layers capturing more complex patterns. Once trained, the CNN can analyze new skin images and classify them based on the features it has learned, accurately distinguishing between different types of skin diseases.

1.1.3 Skin Cancer Recognition

For skin cancer recognition, a model like ResNet analyzes skin lesion images by extracting detailed features across its deep layers. It detects subtle patterns, such as irregular shapes, colors, and textures, that are characteristic of skin cancer. ResNet's architecture, with its skip connections, allows it to efficiently capture these complex features, enabling it to differentiate between benign and malignant lesions. By comparing the extracted features with those learned from a vast dataset of skin images, the model can accurately identify and classify potential skin cancer.

Chapter 2

CHARACTERISTICS OF SKIN DISEASES

The characteristics of skin diseases can vary widely depending on the type of disease. Here are some common features for different categories:

2.1 Fungal Infections

Appearance: Often show up as red, scaly patches or ring-shaped lesions.
Texture: Can be itchy and sometimes flaky or crusty.
Borders: Usually well-defined, often with a clearer center and more inflamed edges.
Location: Common in moist areas like armpits, groin, and between toes.

2.2 Bacterial Infections

Appearance: Can manifest as pustules, boils, or sores with a yellow or greenish discharge.
Texture: Often accompanied by swelling, tenderness, and redness.
Borders: May have irregular borders and can spread rapidly if not treated.
Location: Can occur anywhere on the body but often affects areas prone to cuts or abrasions.

2.3 Viral Infections

Appearance: Includes warts, herpes lesions, and shingles; can appear as blisters or clusters of small bumps.
Texture: Blisters may be fluid-filled and can burst, forming sores.
Borders: Typically well-defined, with a distinct area of infection.
Location: Often localized to specific areas such as the face, genitals, or trunk.

2.4 Other Types

Psoriasis: Characterized by red, scaly patches, often with a silvery sheen, primarily on elbows, knees, and scalp.
Eczema: Typically appears as itchy, inflamed, and red patches, often with a dry and cracked surface.
Melanoma: A type of skin cancer that can appear as an irregularly shaped mole or spot with varied colors.

These characteristics help in differentiating between various skin diseases and are crucial for accurate diagnosis and treatment. Below are some images to show cancerous and non-cancerous disease types.

Figure 2.1: Fungal Disease

Figure 2.2: Cancerous Fungal Disease

Figure 2.3: Viral skin Disease

Figure 2.4: Cancerous Viral Skin Disease

Chapter 3

BACKGROUND STUDY

This section consists of some related works that have been done for skin disease analysis. These works inspired and helped me in my related study.

3.1 An effective multiclass skin cancer classification approach based on deep convolutional neural network.

3.1.1 Introduction

Skin cancer is the most common cancer in humans and can occasionally be fatal. It results from uncontrolled cell growth, with risk factors including smoking, alcohol consumption, allergies, infections, and environmental changes, particularly UV radiation from the sun. Skin cancer is broadly categorized into non-melanoma and melanoma types. Melanoma, known for its malignancy and high mortality rate, involves pigment-producing cells and can be challenging to distinguish from benign moles in early stages. Early detection is crucial for successful treatment.

Artificial intelligence, particularly deep learning (DL) and machine learning (ML), plays a vital role in early skin cancer detection. Convolutional Neural Networks (CNNs), a type of artificial neural network (ANN), have shown significant promise in medical imaging and skin cancer diagnosis. CNNs excel in image processing and classification tasks, demonstrating high accuracy in detecting skin lesions and cancers.

This paper introduces a novel Deep Convolutional Neural Network (DCNN) model designed for precise skin cancer classification. The model outperforms existing transfer

learning models in accuracy, recall, precision, F1-score, specificity, and AUC. It also addresses class imbalance effectively and shows potential for application in detecting various diseases.

3.1.2 Proposed Methodology

The study develops a novel Deep Convolutional Neural Network (DCNN) model for skin cancer classification. Figure 6 shows the model's workflow. We used two public datasets, HAM10000 and ISIC-2019, resizing and normalizing images to improve training efficiency and robustness. To address dataset imbalance, we applied random oversampling, which significantly enhanced model performance. The datasets were split into training and testing sets for model evaluation. The DCNN model architecture, hyperparameters, and training details are optimized to achieve high accuracy in skin cancer classification.

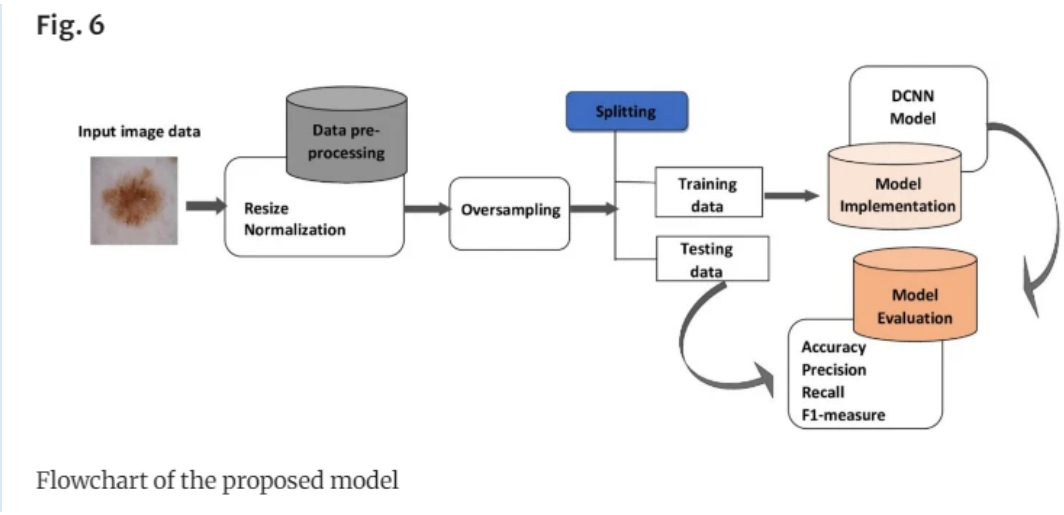

Figure 3.1: Flowchart of proposed model

3.1.3 Discussion and Conclusion

Convolutional Neural Network (DCNN) model in classifying skin cancer from dermoscopy images. The model's performance was rigorously evaluated using key metrics: accuracy, recall, precision, F1-score, specificity, and area under the ROC curve (AUC). These metrics provided a comprehensive assessment of the model's effectiveness in detecting skin cancer. Performance Metrics: The proposed DCNN model demonstrated robust performance across various evaluation metrics. Accuracy, recall, precision, and

F1-score are essential for understanding the model's capability to correctly identify and classify skin cancer instances. The model's ability to balance precision and recall through the F1-score highlights its effectiveness in managing both false positives and false negatives, which is crucial for reliable cancer diagnosis. Specificity and AUC further validated the model's performance in distinguishing between positive and negative cases, with AUC indicating the model's overall ability to rank instances of different classes correctly. Analysis on HAM10000 Dataset: In tackling class imbalance, the random oversampling technique significantly enhanced the DCNN model's performance compared to other resampling methods. The DCNN model achieved impressive accuracy and maintained low loss values during both training and testing phases, showcasing its robustness and reliability. Comparative analysis with other deep learning models and recent studies revealed that the proposed DCNN model outperformed several established models, including VGG16, VGG19, DenseNet, and MobileNetV2, in terms of recall, precision, and F1-score. This performance underscores the effectiveness of our model in addressing the challenges of skin cancer classification. Computational Performance: While the proposed DCNN model exhibited higher computational demands in terms of parameters, inference time, and floating-point operations (FLOPS) compared to other models, its focus on achieving superior accuracy in skin cancer diagnosis justifies these requirements. The balance between computational intensity and diagnostic accuracy is crucial, and our model's higher accuracy emphasizes its value in clinical settings despite the increased computational load. Conclusion: The study successfully introduces a highly effective DCNN model for skin cancer classification, demonstrating significant improvements in performance metrics compared to existing methods. The model's ability to handle class imbalance, combined with its superior accuracy, makes it a promising tool for early skin cancer detection. Future work should explore further optimization of computational efficiency without compromising accuracy and extend the model's applicability to other disease classifications. Overall, this research highlights the potential of advanced neural network architectures in enhancing diagnostic processes in dermatology and other medical fields.

3.2 Skin disease detection using deep learning

3.2.1 Introduction

The skin is the largest organ in the human body, consisting of various layers and structures that can be affected by diseases such as fungal infections, bacterial growth, allergies, and pigmentation issues. Skin diseases, which can be chronic or malignant, require timely treatment to prevent progression. Early and accurate diagnosis is crucial, but current dermatological classification often suffers from limited data and a lack of generalization, especially with methods like dermoscopy. This study focuses on using deep learning models for skin disease classification. It proposes a novel approach combining MobileNet V2 with Long Short-Term Memory (LSTM) networks to classify images of skin conditions. The dataset includes over 10,000 dermatoscopic images representing seven skin diseases. Data augmentation techniques were employed to address dataset imbalance. The MobileNet V2 model was chosen for its efficiency with low-resolution images and lightweight computational requirements. The LSTM component helps manage training challenges, such as gradient disappearance. This approach aims to provide a cost-effective, non-invasive diagnostic tool that can be integrated into mobile applications, assisting both medical practitioners and patients in early and accurate skin disease detection.

3.2.2 Proposed Methodology

MobileNet is a CNN-based model designed for efficient image classification with lower computational requirements, making it suitable for mobile devices and systems with limited resources. Unlike conventional CNNs, MobileNet uses depth-wise separable convolutions to reduce the network size and computational complexity. The MobileNet architecture includes depth-wise convolutions that process each channel separately and point-wise convolutions that combine these channels, optimizing both accuracy and latency. The model's resolution and width multipliers adjust the complexity and size of the network, enabling a balance between performance and computational cost. For skin disease classification, the MobileNet model is used with a $224 \times 224 \times 3$ input image size, where 224×224 represents the height and width, and 3 indicates the RGB channels. The architecture features 32 filters with a 3×3 filter size. By replacing traditional convolutional layers with depth-wise and point-wise convolutions, MobileNet simplifies the model and improves processing speed, making it efficient for mobile and low-resource applications.

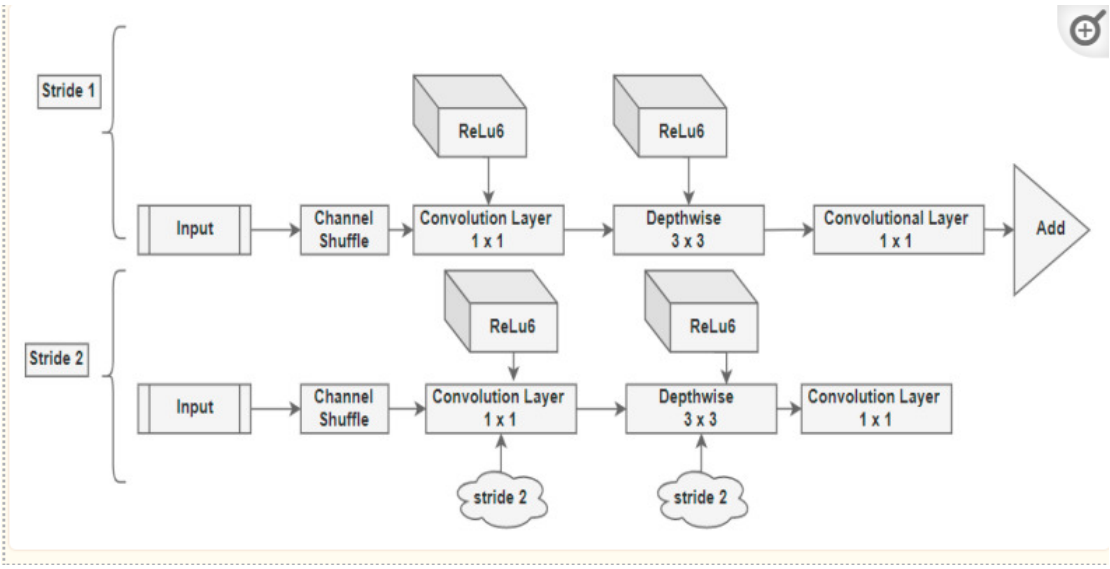

Figure 3.2: Mobilenet Architecture part 1

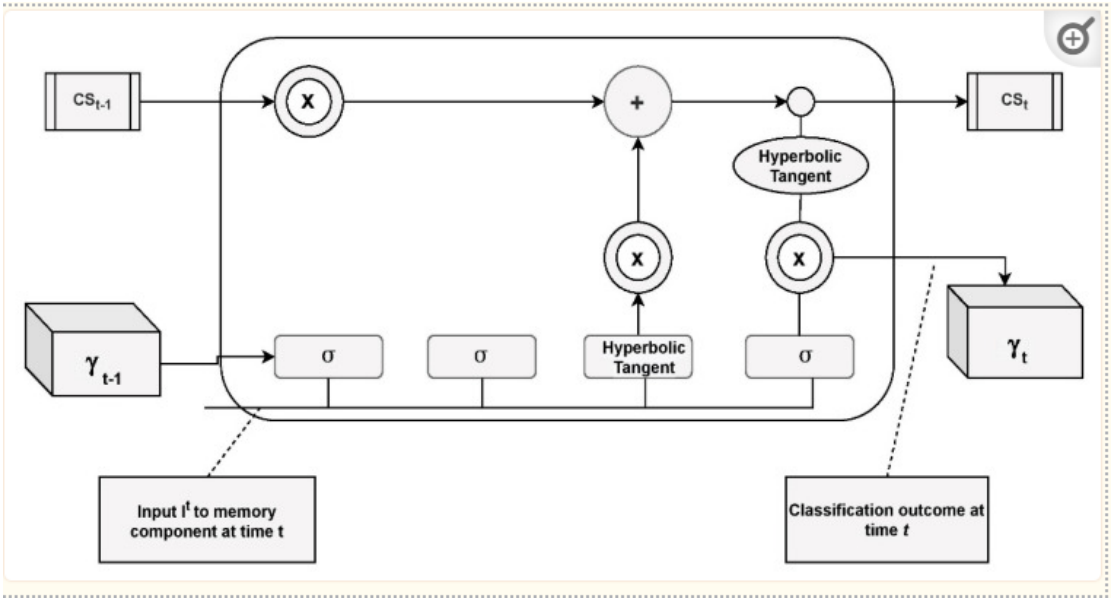

Figure 3.3: Mobilenet Architecture part 2

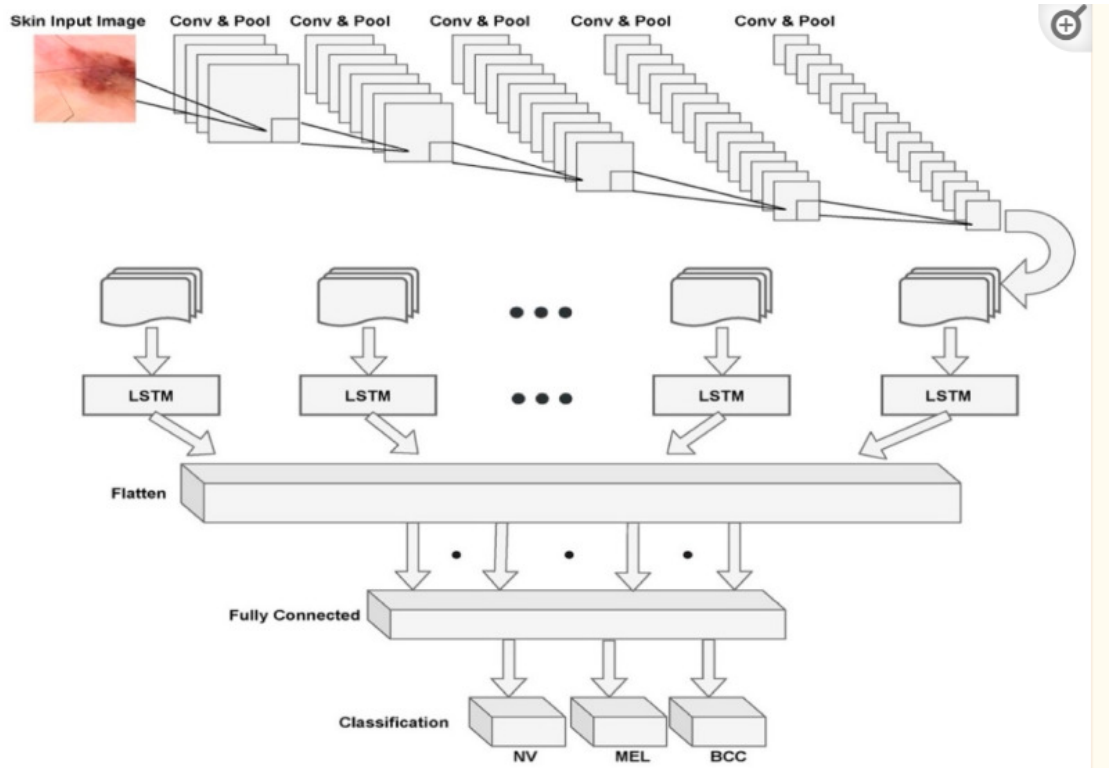

Figure 3.4: Mobilenet Architecture part 3

3.2.3 Discussion & Conclusion

The MobileNet V2 and LSTM-based model demonstrated high efficiency in skin disease classification with an accuracy of 85.34% on real-time images from Kaggle. MobileNet V2's design is well-suited for portable devices, and the integration of LSTM enhances prediction accuracy by incorporating previous timestamp data. Compared to traditional models like CNN, FTNN, and HARIS, the proposed model excelled in classification and tumor growth analysis.

However, the model's precision drops below 80% under poor lighting conditions, highlighting a limitation. The proposed approach aims to complement existing diagnostic methods rather than replace them, as laboratory tests remain more reliable than visual inspection alone for early diagnosis. Future work should address these limitations to improve performance under diverse conditions.

3.3 A machine learning approach for skin disease detection and classification using image segmentation

3.3.1 Introduction

The skin, the largest organ in the human body, is susceptible to various diseases caused by viruses, bacteria, allergies, or fungi. Skin diseases, including Actinic keratosis, Basal cell carcinoma, and Melanoma, can cause psychological and physical issues. Melanoma is particularly dangerous, but early detection can lead to recovery in about 95% of cases. In 2018, the WHO reported approximately 2794 deaths from skin cancer in Bangladesh and over 14 million cases globally. Early and accurate diagnosis is crucial but can be challenging and costly. Automated computer-aided systems using machine learning can enhance the accuracy and speed of skin disease detection, bridging the gap between dermatologists and patients.

3.3.2 Proposed Methodology

The methodology of the proposed end-to-end OCR system is as follows:

- The input image is preprocessed to remove noise and improve the contrast.
- The preprocessed image is then fed into a convolutional neural network (CNN) to extract features.
- The features extracted by the CNN are then fed into a recurrent neural network (RNN) to recognize the handwritten word.
- The RNN predicts a sequence of characters, which are then combined to form the recognized word.

The authors experiment with different CNN architectures, including DenseNet, Xception, NASNet, and MobileNet. They also experiment with two different RNN methods, LSTM and GRU. They found that the DenseNet121 model with GRU recurrent layer gave the best results.

3.3.3 Discussion & Conclusions

The proposed system was evaluated using the BanglaWritting dataset, which is a peer-reviewed Bengali handwritten image dataset. The system achieved a character error rate

(CER) of 0.091 and a word error rate (WER) of 0.273. This is a significant improvement over previous methods for Bengali handwritten OCR.

Chapter 4

Proposed Methodology

4.1 An Overview on Our Proposed Methodology

Our proposed methodology is focused on the detection and classification of skin diseases using advanced convolutional neural networks (CNNs), specifically ResNet and EfficientNet architectures.

The method involves using a series of CNN models to accurately classify skin images into various disease categories. The inclusion of deep learning techniques aims to improve the accuracy and reliability of the diagnosis.

For a comprehensive understanding of our proposed methodology, it is essential to grasp the basics of Convolutional Neural Networks (CNN), Residual Networks (ResNet), and EfficientNet. These advanced models have been shown to significantly enhance image classification tasks, particularly in medical imaging, as indicated in several studies.

4.2 Skin Disease Detection and Classification Process

Our methodology involves several key steps, including:

1. **Data Preprocessing:** The initial step involves preprocessing the images to ensure consistency in input data. This includes resizing all images to a standard size (e.g., 224x224 pixels) and normalizing pixel values to a common scale. This step

ensures that the images are in a uniform format suitable for input into the deep learning models.

2. **Data Segmentation:** While some skin images might include multiple regions of interest, our approach will focus on ensuring that only the affected area of the skin is considered. Advanced image segmentation techniques will be applied to extract these regions, aiding in more accurate classification.

3. **Data Normalization:** Normalizing the images is crucial to ensure that the variations in lighting and skin tone do not affect the model's performance. This process adjusts the image contrast and brightness, creating a uniform appearance across the dataset.

4. **Data Augmentation:** To improve the robustness of our models and prevent overfitting, various data augmentation techniques will be employed. These techniques include rotations, flips, zooms, and color adjustments. By generating multiple versions of each image, the model becomes more resilient to variations in real-world data.

   Our proposed method involves using advanced CNN architectures:

   - **ResNet:** Utilizing the residual learning framework, ResNet allows the network to maintain performance even as the depth of the network increases. This is particularly beneficial in handling complex image data like skin lesions, where subtle differences in texture and color can be critical.
   - **EfficientNet:** EfficientNet introduces a novel scaling method that uniformly scales the network's dimensions based on a compound coefficient. This allows the model to achieve higher accuracy with fewer parameters, making it more efficient for deployment in real-world applications.

5. **Classification:** The output from the CNN models will be used to classify the images into various skin disease categories. The model will predict the likelihood of each disease, and the final classification will be determined based on the highest probability score.

4.3 Model Architecture

Advanced CNN models like ResNet and EfficientNet have been chosen for their superior performance in image classification tasks, particularly in the medical domain. These models offer several advantages.

The core of our proposed methodology lies in the use of advanced CNN architectures tailored to the specific challenges of skin disease classification. We explore and compare the performance of several state-of-the-art models, including:

- **ResNet (Residual Networks):** ResNet introduces the concept of residual learning through the use of skip connections, which allow the model to bypass certain layers. This innovation addresses the problem of vanishing gradients in deep networks, enabling the training of very deep models without loss of performance. ResNet is particularly advantageous in our application, where distinguishing subtle differences between skin conditions requires deep feature extraction. We experiment with different versions of ResNet, including ResNet-50, ResNet-101, and ResNet-152, to determine the optimal depth for our task.
- **EfficientNet:** EfficientNet represents a paradigm shift in CNN design by introducing a compound scaling method that uniformly scales all dimensions of the network—depth, width, and resolution—based on a single compound coefficient. This approach allows EfficientNet to achieve superior accuracy with fewer parameters and computational resources. Given the computational constraints often present in clinical settings, EfficientNet is a prime candidate for deployment in real-world applications. We explore various versions of EfficientNet, from B0 to B7, to find the best trade-off between accuracy and efficiency.
- **Transfer Learning:** To accelerate training and improve performance, we leverage transfer learning by initializing our models with weights pre-trained on large datasets like ImageNet. Transfer learning allows the models to start with a strong foundation of general features, which are then fine-tuned on our specific dataset of skin diseases. This approach significantly reduces the amount of data and time required to train the models to a high level of accuracy.

4.4 Convolutional Neural Networks (CNNs) for Skin Disease Analysis

CNNs are a class of deep learning models particularly well-suited for image analysis tasks. They consist of multiple layers that automatically learn to extract features from input images, such as edges, textures, and more complex patterns.

- **Convolutional Layers:** These layers apply filters to the input image, detecting essential features that are crucial for classification. Multiple convolutional layers

are stacked to build a hierarchy of features.

- **Pooling Layers:** Pooling layers reduce the spatial dimensions of the feature maps, thereby decreasing the number of parameters and computational load, while retaining the most important information.
- **Activation Functions:** Non-linear activation functions like ReLU are used to introduce non-linearity into the model, enabling it to learn more complex patterns.
- **Fully Connected Layers:** These layers at the end of the network consolidate the learned features to make the final prediction. They connect every neuron in one layer to every neuron in the next, facilitating the learning of intricate relationships within the data.

CNNs have proven to be effective for tasks such as skin disease classification due to their ability to automatically learn and extract relevant features from images. They are especially useful in medical applications where the subtlety of features can be critical for accurate diagnosis.

4.4.1 Initial Model

EfficientNet is a family of convolutional neural networks (CNNs) designed for high efficiency and accuracy. The series was introduced by Google AI and is based on a compound scaling method that balances network depth, width, and resolution.

4.4.1.1 EfficientNet-B0

- **Baseline model of the EfficientNet family.**
- Uses Neural Architecture Search (NAS) to find an optimal balance between performance and computational cost.
- **Parameters:** ~ 5.3M
- **FLOPS:** ~ 0.39B
- **Input Image Size:** 224×224
- **Performance:** Achieves 77.1% top-1 accuracy on ImageNet.
- **Good for:** Lightweight applications, mobile devices, and tasks requiring low computational cost.

4.4.1.2 EfficientNet-B1

- **An extended version of B0 with slightly increased depth, width, and resolution.**
- Scales up the input resolution and number of layers slightly to improve accuracy.
- **Parameters:** ~ 7.8M
- **FLOPS:** ~ 0.70B
- **Input Image Size:** 240×240
- **Performance:** Achieves 79.1% top-1 accuracy on ImageNet.
- **Good for:** Applications where slightly more computation can be afforded for improved accuracy.

Comparison

Model	Parameters	FLOPS	Input Size	Accuracy (Top-1)
EfficientNet-B0	5.3M	0.39B	224×224	77.1%
EfficientNet-B1	7.8M	0.70B	240×240	79.1%

Table 4.1: Comparison of EfficientNet Models

EfficientNet-B1 is a good choice if you want a slight performance boost over B0 and can handle the extra computational cost. However, if efficiency is the priority, B0 is still a strong option.

4.4.2 Final Model

4.4.2.1 EfficientNet-B2

EfficientNet-B2 is a scaled-up version of EfficientNet-B0, using the compound scaling method to increase the network's depth, width, and input resolution while maintaining efficiency.

Key Features

- More Layers & Channels: Compared to B0 and B1, B3 has a deeper and wider architecture, leading to better feature extraction.

- Higher Input Resolution: Uses a 300×300 input image size, capturing more fine-grained details.
- Improved Accuracy: Achieves 81.6% top-1 accuracy on ImageNet.
- Moderate Computational Cost: Though heavier than B0/B1, it remains efficient compared to traditional models like ResNet.

Specifications

Model	Parameters	FLOPS	Input Size	Top-1 Accuracy
EfficientNet-B0	5.3M	0.39B	224×224	77.1%
EfficientNet-B1	7.8M	0.70B	240×240	79.1%
EfficientNet-B2	9.1M	1.0B	260×260	80.1%

Table 4.2: Specifications of EfficientNet Models

Chapter 5

Results and Discussion

5.1 MY Current Work

5.1.1 Background Study

Background study is important for research because it provides context, guides research design, identifies gaps, validates research relevance, and informs data analysis. I have added all the research I have studied for my thesis in the Reference section. I have added some summary for some of the paper. Due to time limitation I could not mention each of the paper, but almost all of them helped me for selecting my current methodology.

5.1.2 Data Analysis

Accurate diagnosis of skin diseases is vital for effective treatment. With the rise of deep learning, particularly ResNet-based Convolutional Neural Networks (CNNs), we can enhance skin disease classification by analyzing dermoscopic images. These high-resolution images capture essential details, but variability in lighting, skin tone, and lesion types presents challenges.

Our approach tackles these challenges through systematic data preprocessing, segmentation, and augmentation. By refining and enhancing the image data, we extract key features that improve classification accuracy, ultimately aiding in more precise and reliable diagnoses.

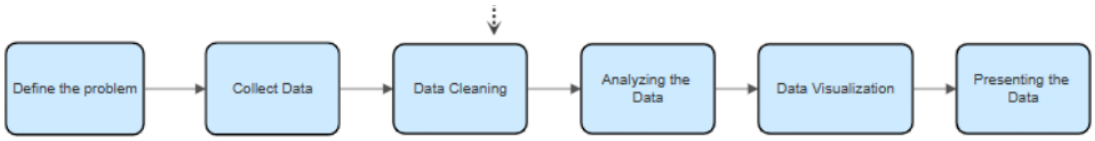

Figure 5.1: Data Analysis steps

5.1.3 Data Collection

Our proposed methodology requires a large number of data to be trained. I have collected my data from various resources. For our skin disease analysis, we gathered a diverse dataset from various dermatological sources, including medical databases and publicly available image repositories. The dataset includes images of common skin conditions such as fungal infections, psoriasis, eczema, and other dermatological disorders. This diverse collection ensures a comprehensive representation of different skin diseases, enabling more accurate and robust model training for better diagnosis and classification.

Figure 5.2: Collected data few images

5.1.4 Data Splitting and File Organization

Code:

```
import os
import shutil
from sklearn.model_selection import train_test_split

filepaths = []
labels = []
for class_dir in os.listdir(dataset_dir):
class_path = os.path.join(dataset_dir, class_dir)
for fname in os.listdir(class_path):
filepaths.append(os.path.join(class_path, fname))
labels.append(class_dir)
```

```
12
13 train_files, temp_files, train_labels, temp_labels = train_test_split
       (
14 filepaths, labels, test_size=0.3, stratify=labels)
15
16
17 val_files, test_files, val_labels, test_labels = train_test_split(
18 temp_files, temp_labels, test_size=0.5, stratify=temp_labels)
19
20 def copy_files(filepaths, labels, dest_dir):
21 for filepath, label in zip(filepaths, labels):
22 dest_path = os.path.join(dest_dir, label)
23 os.makedirs(dest_path, exist_ok=True)
24 shutil.copy(filepath, dest_path)
25
26 copy_files(train_files, train_labels, train_dir)
27 copy_files(val_files, val_labels, val_dir)
28 copy_files(test_files, test_labels, test_dir)
```

In this process, I have organized the dataset for my skin disease analysis project by first obtaining all file paths and their corresponding labels. Each file is categorized based on the skin condition it represents.

To ensure a robust evaluation of the model, I implemented a systematic split of the data:

1. **Initial Split (70/30):** I divided the data into training and temporary sets, with 70% allocated for training and 30% set aside for further splitting. This stratified split maintains the class distribution across both sets, ensuring balanced representation of each skin condition.

2. **Secondary Split (50/50):** The temporary set was further split into validation and test sets, each comprising 50% of the temporary set. This ensures that both validation and test datasets are equally representative and distinct from the training data.

3. **File Organization:** I then organized the files into their respective directories for training, validation, and testing. Each file was copied into directories named after its label, ensuring that each set is correctly categorized for model training and evaluation.

This approach ensures that the model is trained on a diverse and representative sample, while the validation and test sets provide an unbiased evaluation of the model's

performance.

5.2 Data Cleaning and Preprocessing

Code:

```
IMG_SIZE = (224, 224)
BATCH_SIZE = 32

train_datagen = ImageDataGenerator(
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest',
    preprocessing_function=preprocess_input
)

val_datagen = ImageDataGenerator(preprocessing_function=
   preprocess_input)
test_datagen = ImageDataGenerator(preprocessing_function=
   preprocess_input)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical'
)

val_generator = val_datagen.flow_from_directory(
    val_dir,
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical'
)

test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical'
```

```
)

print(f"Classes: {train_generator.class_indices}")
print(f"Number of classes: {len(train_generator.class_indices)}")

def plot_class_distribution(generator):
    class_counts = {class_name: 0 for class_name in generator.
        class_indices.keys()}
    for subdir, _, files in os.walk(generator.directory):
        if files:
            class_name = os.path.basename(subdir)
            if class_name in class_counts:
                class_counts[class_name] += len(files)

    plt.figure(figsize=(10, 6))
    plt.bar(class_counts.keys(), class_counts.values(), color='
        skyblue')
    plt.xlabel('Classes')
    plt.ylabel('Number of Images')
    plt.title('Distribution of Images per Class')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

plot_class_distribution(train_generator)
```

I have implemented these steps to effectively prepare and manage the dataset for my skin disease analysis project. Here's why:

1. **Image and Batch Size Configuration:** By setting the image size to (224, 224) and the batch size to 32, I align with EfficientNetB0's default input requirements and optimize the processing efficiency for training.

2. **Data Augmentation and Preprocessing:** For the training set, I used ImageDataGenerator to apply various augmentations such as rotations, shifts, and flips. This approach enhances model robustness by providing a more varied set of images and preventing overfitting. For validation and test data, I applied EfficientNetB0-specific preprocessing without augmentation to ensure a fair evaluation of the model's performance.

3. **Data Generators:** I created data generators for training, validation, and testing with the specified configurations. This setup facilitates the efficient loading and batching of data, ensuring that the model trains on diverse examples and is evaluated properly.

4. **Verification and Visualization:** I verified the class indices and the number of classes to confirm correct data loading. Additionally, I plotted the class distribution to visualize the balance of images across different classes. This helps in identifying any class imbalances that could affect model performance and ensures a fair evaluation.

These steps collectively ensure that the data is prepared and managed effectively, leading to a more reliable and robust model for skin disease analysis.

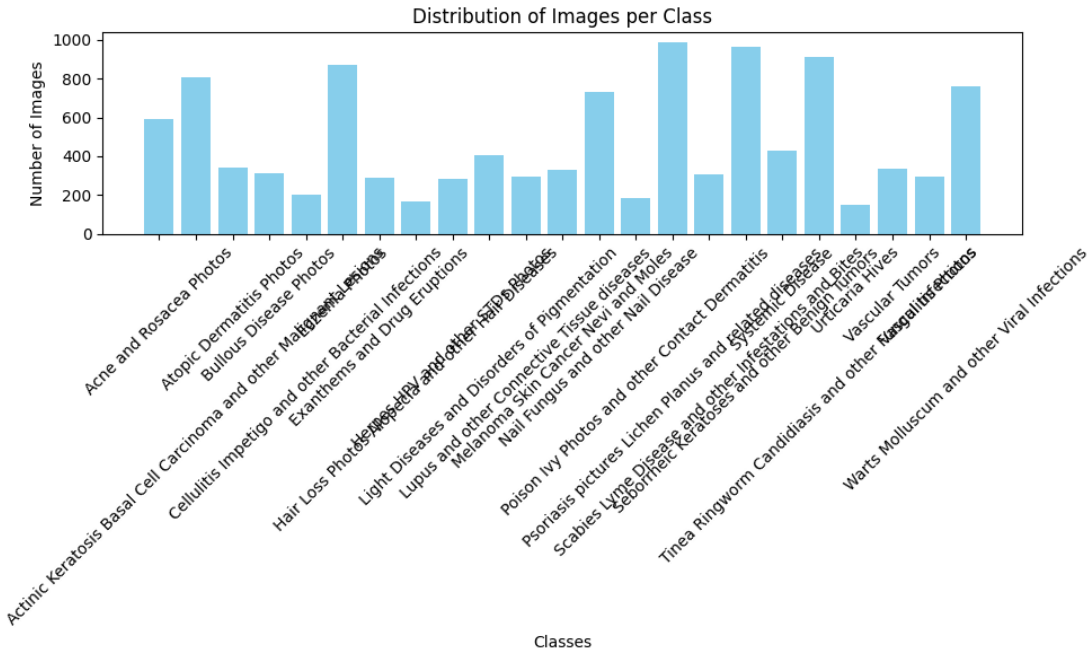

Figure 5.3: Distribution of images per class

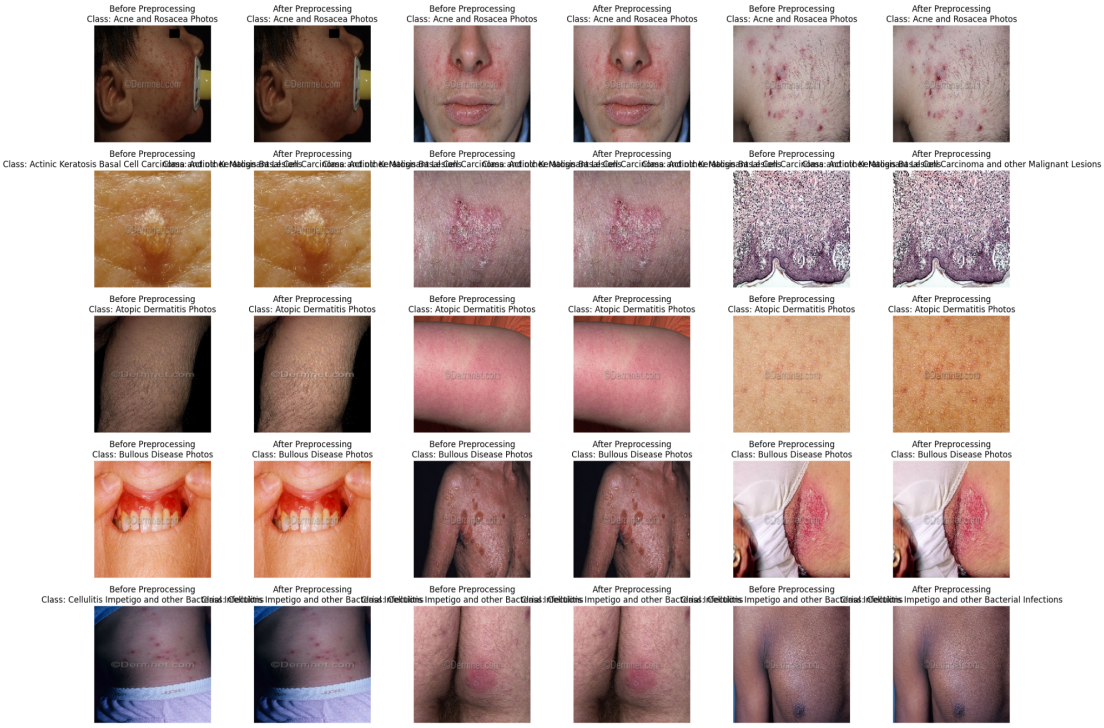

Figure 5.4: Image Before and After preprocessing

5.3 Training the model

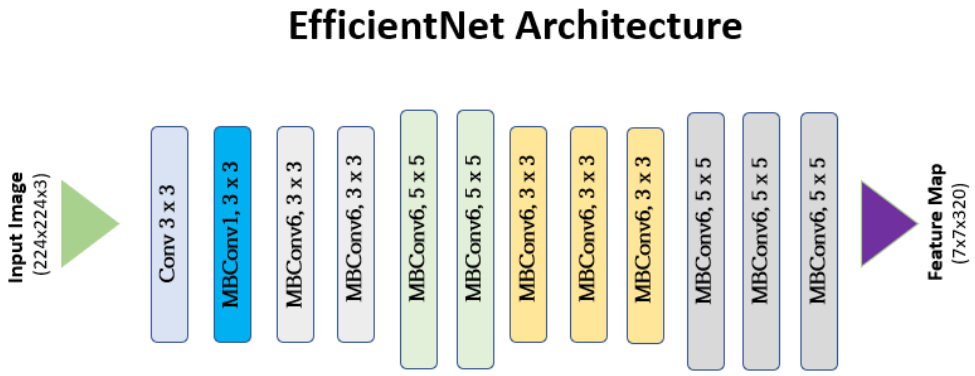

Figure 5.5: EfficientNet Architecture

The EfficientNet architecture depicted in the image will work for skin disease analysis by processing input images of skin lesions through a series of convolutional layers and mobile inverted bottleneck convolution (MBConv) blocks. Starting with an input im-

age of size 224x224x3, the model will extract hierarchical features, from simple edges and textures in the initial layers to more complex patterns in the deeper layers. Each MBConv block refines these features, allowing the model to recognize various characteristics of skin diseases like fungal infections, psoriasis, or eczema. The final feature map generated by the network captures the essential information needed for classification. This feature map can then be passed to fully connected layers (not shown in the image) for classification, predicting the type of skin disease present in the image. EfficientNet's balance of depth, width, and resolution makes it particularly effective for this task, offering high accuracy with fewer computational resources.

Each layer in the EfficientNet architecture will work as follows for skin disease analysis:

1. **Input Layer (224x224x3):**
    - Each layer will work by accepting standardized skin lesion images of size 224x224 pixels with three color channels (RGB). This input layer is crucial for ensuring consistency in feature extraction across all images.
2. **Conv 3x3:**
    - Each layer will work by applying a 3x3 convolutional filter to the input image. This step captures basic visual features, such as edges and textures, which are important for recognizing initial patterns in skin lesions.
3. **MBConv1, 3x3:**
    - Each layer will work by utilizing a Mobile Inverted Bottleneck Convolution (MBConv) block with a 3x3 kernel and an expansion ratio of 1. This helps process simple structures in the lesion images with minimal computational cost.
4. **MBConv6, 3x3:**
    - Each layer will work by expanding the input features by a factor of 6 before applying a 3x3 convolution. This allows the model to capture more complex patterns, like the unique textures of different skin diseases.
5. **MBConv6, 5x5:**
    - Each layer will work by using a 5x5 convolution kernel after expanding the input features. This larger kernel size captures more contextual information,

essential for distinguishing between similar skin conditions such as eczema and psoriasis.

6. **MBConv6, 3x3 and 5x5 (Repeated):**

   - Each layer will work by continuously refining the feature maps through additional 3x3 and 5x5 convolutions, enabling the model to learn more abstract and intricate representations of the skin diseases.

7. **Final MBConv6, 5x5:**

   - Each layer will work by further refining the deep features using a final 5x5 convolution, ensuring that the model captures the most detailed and significant characteristics necessary for accurate skin disease classification.

8. **Feature Map (7x7x320):**

   - Each layer will work by outputting a 7x7x320 feature map, which encapsulates the distilled information from all previous layers. This feature map is critical for making the final prediction about the type of skin disease present in the image.

Overall, each layer in the EfficientNet architecture contributes progressively to the model's ability to analyze and classify various skin diseases, such as fungal infections, psoriasis, and eczema, with high accuracy.

5.4 Final Trained model

Class-wise Accuracy (By EfficientNet B2 on Test Set of Data)

Class	**Accuracy**
Acne and Rosacea Photos	69.05%
Actinic Keratosis Basal Cell Carcinoma and other Malignant Lesions	47.40%
Atopic Dermatitis Photos	42.47%
Bullous Disease Photos	34.33%
Cellulitis Impetigo and other Bacterial Infections	23.26%
Eczema Photos	64.86%
Exanthems and Drug Eruptions	38.33%

Hair Loss Photos Alopecia and other Hair Diseases	50.00%
Herpes HPV and other STDs Photos	46.67%
Light Diseases and Disorders of Pigmentation	52.94%
Lupus and other Connective Tissue diseases	41.27%
Melanoma Skin Cancer Nevi and Moles	62.86%
Nail Fungus and other Nail Disease	83.33%
Poison Ivy Photos and other Contact Dermatitis	58.97%
Psoriasis pictures Lichen Planus and related diseases	50.24%
Scabies Lyme Disease and other Infestations and Bites	55.38%
Seborrheic Keratoses and other Benign Tumors	64.08%
Systemic Disease	38.46%
Tinea Ringworm Candidiasis and other Fungal Infections	67.18%
Urticaria Hives	68.75%
Vascular Tumors	56.16%
Vasculitis Photos	61.29%
Warts Molluscum and other Viral Infections	68.10%

Skin disease Model Building code:

```
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from tensorflow.keras.callbacks import TensorBoard, ModelCheckpoint,
    EarlyStopping, ReduceLROnPlateau, Callback
from tensorflow.keras.applications import EfficientNetB2
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D,
    Dropout
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications.efficientnet import
    preprocess_input
from keras.models import load_model
from tensorflow.keras.regularizers import l2

# Data Preparation

dataset_dir = '/content/drive/MyDrive/Final_year_project/dataset'
train_dir = '/content/drive/MyDrive/Final_year_project/train_data'
val_dir = '/content/drive/MyDrive/Final_year_project/validation_data'
test_dir = '/content/drive/MyDrive/Final_year_project/test_data'

for dir_path in [train_dir, val_dir, test_dir]:
    os.makedirs(dir_path, exist_ok=True)

IMG_SIZE = (260, 260)
BATCH_SIZE = 32

# Data Augmentation and Generators

train_datagen = ImageDataGenerator(
    rotation_range=45,
    width_shift_range=0.15,
    height_shift_range=0.15,
    shear_range=0.1,
    zoom_range=0.25,
    horizontal_flip=True,
    vertical_flip=True,
    brightness_range=[0.7, 1.3],
    fill_mode='nearest',
    preprocessing_function=preprocess_input
```

```
)

val_datagen = ImageDataGenerator(preprocessing_function=
    preprocess_input)
test_datagen = ImageDataGenerator(preprocessing_function=
    preprocess_input)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical'
)

val_generator = val_datagen.flow_from_directory(
    val_dir,
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical'
)

test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical'
)

print(f"Classes: {train_generator.class_indices}")
print(f"Number of classes: {len(train_generator.class_indices)}")

# Model Definition

def create_model(num_classes):
    base_model = EfficientNetB2(weights='imagenet', include_top=False
        , input_shape=(260, 260, 3))
    x = base_model.output
    x = GlobalAveragePooling2D()(x)
    x = Dense(1024, activation='relu', kernel_regularizer=l2(0.001))(
        x)
    x = Dropout(0.5)(x)
    outputs = Dense(num_classes, activation='softmax')(x)
    model = Model(inputs=base_model.input, outputs=outputs)
    model.compile(optimizer=Adam(learning_rate=1e-5), loss='
        categorical_crossentropy', metrics=['accuracy'])
```

```
    return model

# Model Loading and Initialization

model_path = r'/content/drive/MyDrive/Final_year_project/Imagenet2_best_model.keras'

if os.path.exists(model_path):
    print("Loading model from checkpoint...")
    model = load_model(model_path)
    initial_epoch = model.optimizer.iterations // (train_generator.samples // BATCH_SIZE)
else:
    print("No checkpoint found, initializing new model...")
    num_classes = len(train_generator.class_indices)
    model = create_model(num_classes)
    initial_epoch = 0

# Callbacks

class PrintMetrics(Callback):
    def on_epoch_end(self, epoch, logs=None):
        print(f"Epoch {epoch+1} - Loss: {logs['loss']:.4f}, Accuracy: {logs['accuracy']:.4f}, "
              f"Val Loss: {logs['val_loss']:.4f}, Val Accuracy: {logs['val_accuracy']:.4f}")

early_stopping = EarlyStopping(monitor='val_loss', patience=20, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=10, min_lr=1e-7)
model_checkpoint = ModelCheckpoint(model_path, save_best_only=True, monitor='val_loss', mode='min')
print_metrics = PrintMetrics()

# Model Training

history = model.fit(
    train_generator,
    epochs=130,
    validation_data=val_generator,
    callbacks=[early_stopping, reduce_lr, model_checkpoint, print_metrics],
    initial_epoch=initial_epoch
)
```

```
118
119 # Model Evaluation
120
121 test_loss, test_accuracy = model.evaluate(test_generator)
122 print(f"Test Loss: {test_loss:.4f}, Test Accuracy: {test_accuracy:.4f
    }")
```

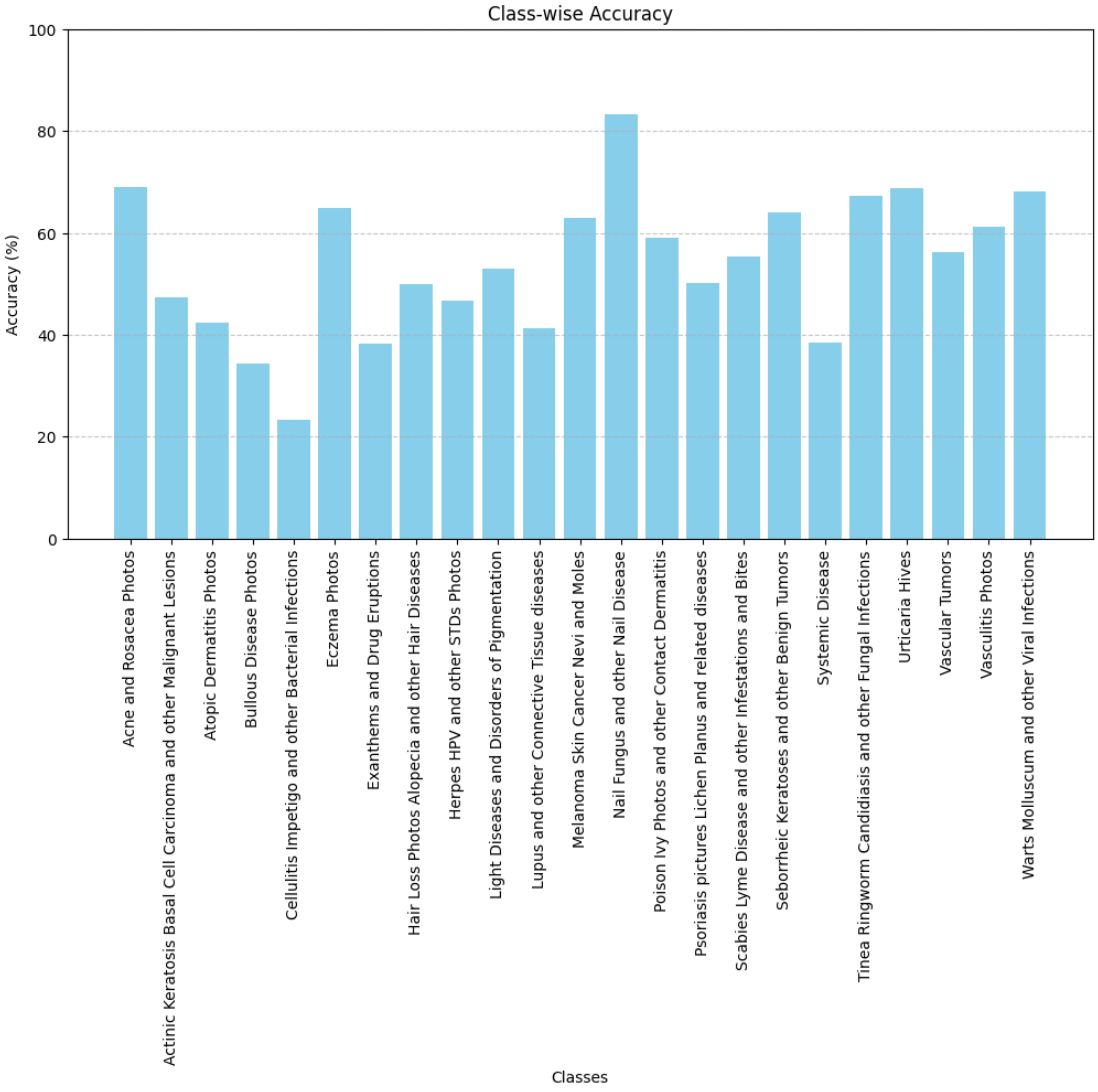

Figure 5.6: Class wise accuracy

Chapter 6

Result & Conclusion

6.1 Result

After training the dataset using the EfficientNetB0 and EfficientNetB1 model. I have received the following best result from EfficentB1. This is currently the best result I have . And the model is still in training .

Train Set	Validation Set
Accuracy: 0.81	**Accuracy:** 0.56
Train Loss: 0.60	**Validation Loss:** 2.55

Table 6.1: Initial Result on EfficientNet B1

Here is a screenshot taken after last fine tuning the model:

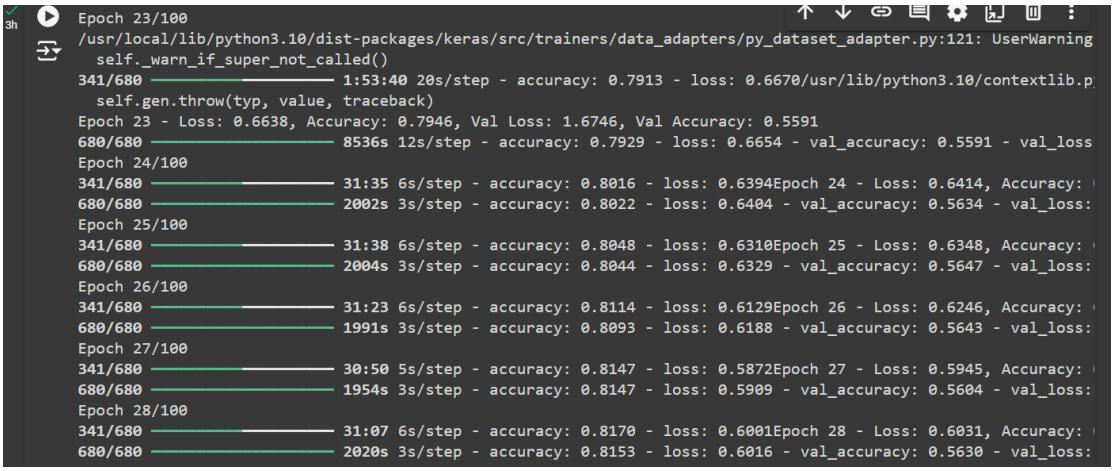

Figure 6.1: Initial model Accuracy and Error

Train Set	Validation Set
Accuracy: 0.89	**Accuracy:** 0.60
Train Loss: 0.57	**Validation Loss:** 2.08

Table 6.2: Final Result on EfficientB2

Here is a screenshot taken after last fine tuning the model:

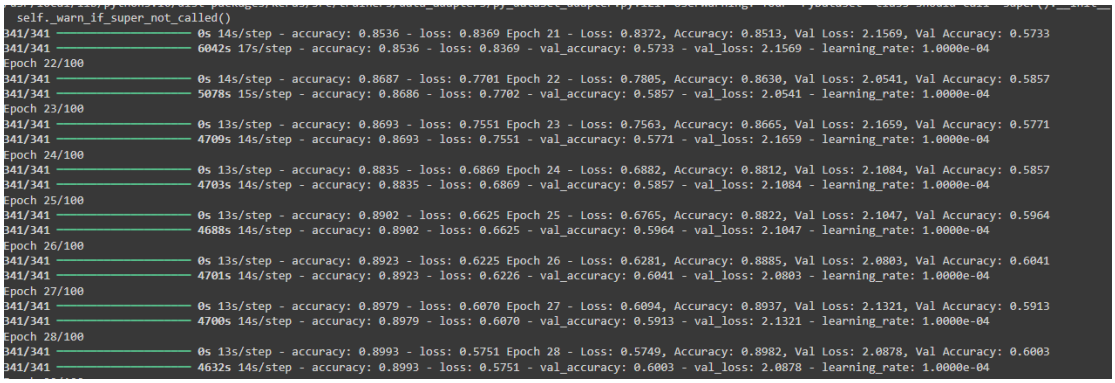

Figure 6.2: Final model Accuracy and Error

6.2 CONFUSION MATRIX

Overall Test Set Accuracy is 57.50.
Summary of Confusion Matrix: Matrix Structure: Rows represent true classes, and columns represent predicted classes. Each cell shows the number of samples classified into the respective true-predicted class combination.

Diagonal Cells (Correct Predictions): Diagonal values indicate correctly classified samples for each class. Higher diagonal values signify better performance for that class.

Off-Diagonal Cells (Misclassifications): Non-diagonal cells represent misclassified samples. These highlight confusion between similar classes or insufficient training data.

Class-Specific Accuracy: Accuracy = Correct predictions for a class ÷ Total samples in that class. This identifies how well the model performs on each individual class.

Overall Trends: Strong classes show high diagonal values, while weak classes have many off-diagonal misclassifications. Analyze confused pairs for improvement.

Overall Accuracy: Total Accuracy = Sum of diagonal values ÷ Total test samples. This gives the model's overall performance on the test set.

Insights for Improvement: Strengthen weak classes with more data or targeted augmentation. Refine confusing pairs with better feature extraction or preprocessing.

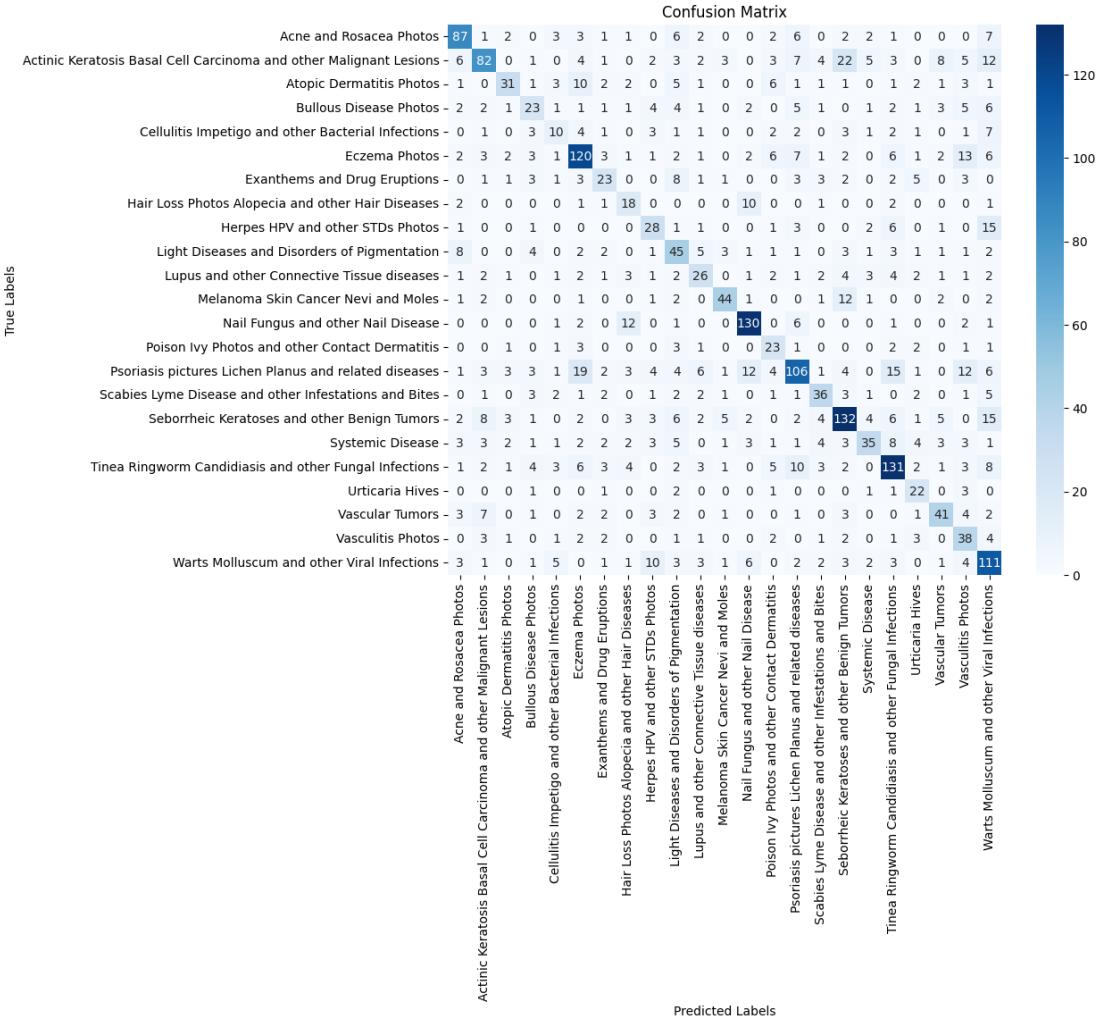

Figure 6.3: Confusion Matrix

6.3 SCREENSHOT OF WEBSITE

Figure 6.4: Flask website

6.4 Conclusion

In this project, we developed a deep learning model for skin disease analysis using the EfficientNetB2 architecture. The model has demonstrated promising results, with training accuracy surpassing 89% by Epoch 28. The validation accuracy has also shown consistent improvement, indicating that the model is effectively learning to generalize from the training data.

Despite these positive trends, it is important to note that the model is still in the fine-tuning phase and is not yet finalized. During this phase, we observed that while the training loss continues to decrease, the validation loss has shown some fluctuations. This suggests that the model may benefit from further adjustments to prevent overfitting and improve its generalization capabilities.

As we continue to refine the model, our focus will be on closing the gap between training and validation performance. This may involve experimenting with different hyper-

parameters, learning rates, and data augmentation techniques.

In summary, the current model shows strong potential for accurate skin disease classification, but further fine-tuning is required to ensure optimal performance and reliability. The next steps in this project will involve continuing this iterative process of refinement to achieve a final model that generalizes well to new, unseen data.

6.5 Future Scope

Future work will focus on fine-tuning the model to further improve its performance. Fine-tuning is crucial as it allows for adjustments to the model's parameters to better capture the nuances in the data, leading to better generalization and accuracy. This process will help in reducing the gap between training and validation accuracy, ensuring that the model performs well not only on the training set but also on new, unseen data.

References

[1] E. H. Houssein, D. A. Abdelkareem, and G. Hu, "An effective multiclass skin cancer classification approach based on deep convolutional neural network," *Cluster Comput*, vol. 27, pp. 12799–12819, 2024. [Online]. Available: https://doi.org/10.1007/s10586-024-04540-1.

[2] R. Yadav and A. Bhat, "A systematic literature survey on skin disease detection and classification using machine learning and deep learning," *Multimed Tools Appl*, vol. 83, pp. 78093–78124, 2024. [Online]. Available: https://doi.org/10.1007/s11042-024-18119-w.

[3] M. Ahammed, M. A. Mamun, and M. S. Uddin, "A machine learning approach for skin disease detection and classification using image segmentation," *Healthcare Analytics*, vol. 2, art. no. 100122, Nov. 2022. [Online]. Available: https://doi.org/10.1016/j.health.2022.100122.

[4] T. G. Debelee, W. A. Mustafa, and H. Alquran, "Skin Lesion Classification and Detection Using Machine Learning Techniques: A Systematic Review," *Diagnostics (Basel)*, vol. 13, no. 19, p. 3147, Oct. 2023. [Online]. Available: https://doi.org/10.3390/diagnostics13193147.