Abstract:

Skin cancer, including melanoma, is a serious global health concern. It is essential to recognize skin diseases early to improve treatment results. In this project, our goal is to develop a system that enhances the early detection of skin melanoma and other types of skin lesions using deep learning and computer vision techniques.. The project aims to provide an effective solution for dermatologists, healthcare professionals, and individuals to identify potential skin melanoma through an intelligent system. Our system employs convolutional neural networks (CNN) models (EfficientNet-B0, Resnet50, Ensemble Model) designed for skin melanoma detection. Users can upload skin images to the web application, which is then fed to the backend CNN models for analysis. The model's predictions are presented to users through the web application interface, enabling early detection and informed decision-making. The significance of this project lies in its potential to aid medical professionals and individuals in identifying potential skin melanoma at an early stage, thus improving the chances of successful treatment. Early detection can make a substantial difference in patient outcomes. The developed system enhances skin melanoma detection by harnessing the power of deep learning and making it accessible to users through a user-friendly web application. This project serves as a valuable tool in the ongoing efforts to combat skin melanoma effectively.

Chapter 1

Introduction

1.1 Introduction

Skin melanoma is deadliest forms of cancer in the world. If it is not detected at an early stage, it has the potential to infect other parts of the skin [1]. The American Cancer Society reports that although melanoma skin cancer accounts for only 1% of all cancer cases, it has a higher mortality rate [2]. About 75% of skin cancer-related deaths are caused by skin melanoma [3]. Melanoma is a dangerous, rare, and fatal form of skin cancer. An early and accurate diagnosis of this condition is essential for successful treatment. However, diagnosing skin melanoma can be challenging, even for experienced dermatologists. To improve diagnosis and treatment outcomes, we built a deep learning model that can identify skin melanoma from images [4]. This deep learning-based project has the potential to increase accessibility and cost-effectiveness of skin melanoma diagnosis, particularly in underdeveloped nations where access to advanced medical technology may be constrained.

1.2 Medical Imaging and its Applications:

Medical image processing plays a pivotal role in the timely diagnosis, treatment, and detection of disorders [5]. Utilizing a range of technologies, medical imaging provides valuable insights into the inner workings of organs and structures, facilitating diagnosis and continuous monitoring. However, as the field of medical image analysis advances, challenges in organ segmentation and identifying abnormalities become more intricate. The classification of medical images is instrumental in determining medication dosage and minimizing radiation exposure while curbing the progression of conditions like tumors [6].

This methodology involves capturing images of the body's interior, which is essential for clinical research, medical treatment, and gaining insights into the physiology of specific organs and tissues. It has become an indispensable tool for both medical diagnosis and treatment, with doctors relying on these images to study internal anatomy and make informed decisions [7]. Medical image processing encompasses various domains, including computer vision, pattern recognition, image mining, and machine learning [8].

Neural networks have emerged as highly effective tools in addressing various challenges in image identification, and they are increasingly integrated into medical practices [9]. Recent advancements in computer vision have sparked a growing interest in the application of Transformers within the field of medical imaging. Transformers, with their ability to capture global context, offer a distinct advantage over Convolutional Neural Networks (CNNs) with their local receptive fields [10].

The advancements in modern medicine are heavily reliant on the field of medical imaging, with X-rays, MRIs, ultrasounds, endoscopies, tactile imaging, computerized tomography (CT scan), and various other imaging techniques serving as prime examples of these invaluable methods.

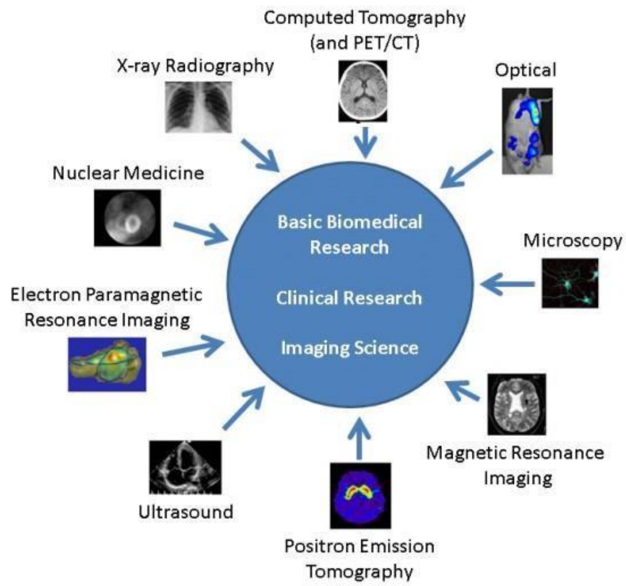

Fig 1.1. Different types of Medical Imaging.

Medical imaging involves the utilization of diverse imaging techniques in clinical research and medical interventions to generate visual representations of the interior of the human body. Various imaging modalities have been developed, including cross-sectional and X-ray-based methods like SPECT, PET, and ultrasound, as well as tomographic techniques. In the field of medical image processing, one critical and challenging aspect is segmentation, which has gained prominence in image interpretation [11]. Currently, methods for image segmentation are advancing rapidly, offering increased precision and efficiency.

By combining new ideas and technology, comprehensive segmentation algorithms are beginning to develop that can successfully segment different kinds of images [12]. Many tasks, including mass detection in mammograms, automated blood cell classification, border detection in coronary angiograms, surgical planning, surgical simulation, tumor detection and segmentation, brain development research, functional mapping, angiogram border detection, image registration, heart segmentation, and cardiac image analysis, are among the medical scenarios where these image segmentation techniques find use.

Segmentation is a vital technique in the medical field, employed to separate distinct tissues by extracting and categorizing relevant features. This process involves grouping visual pixels into anatomical regions, aiding in the identification of structures such as bones, muscles, and blood vessels. Furthermore, this approach can be applied to the characterization of breast tumors using MRI images and the extraction of information related to skin cancer from visual data.

1.3. Skin Cancer and its Types:

One of the biggest global health burdens is cancer, accounting for over 10.0 million deaths from the disease in 2020 (or 9.9 million if non-melanoma skin cancer is taken out) [13]. While individuals with darker skin tones typically have lower incidences of skin cancer compared to light-skinned
Caucasians, it is crucial for medical professionals to enhance their knowledge of skin cancer in people with diverse complexions to facilitate early diagnosis and improve outcomes [14]. Skin cancer has now become the fifth most commonly reported disease globally, exerting a substantial impact on both the economy and public health [15].

Melanoma skin cancer (MSC) and non-melanoma skin cancer (NMSC) are the two main kinds of skin cancer, and they each have unique characteristics [16]. The primary preventable cause of skin cancer is determined to be excessive exposure to UV radiation from both artificial and natural sources, including tanning beds and the sun. According to predictions from the World Health Organization (WHO)-affiliated International Agency for Research on Cancer (IARC), 8.2 million deaths worldwide were attributed to cancer in 2012; by 2030, an additional 27 million new cases are expected [17].

The inner dermis and outer epidermis are the two main layers of skin. There are three main cell types in the epidermis, where skin cancer first appears. Squamous cells are the thin, flat cells that make up the top layer of the epidermis. Melanocytes and spherical cells known as basal cells are located beneath the squamous cells. Cells that produce melanin are found in the epidermis' basal layer. Skin color is determined by a pigment called melanin, which is produced in excess by melanocytes in response to sun exposure. This causes the skin to become darker. There are numerous subtypes of skin cancer, and each has distinct traits. There are the following subtypes of skin cancer.

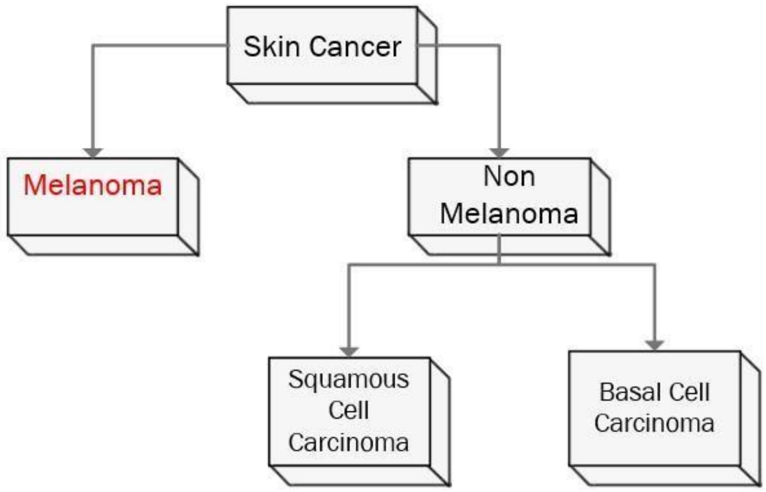

Fig 1.2. Types of Skin Cancer

1.3.1 Non-Melanoma Skin Cancer:

NMSC is a type of skin cancer found mostly in individuals having lighter skin tones. It includes cancers that develop in skin cells other than melanocytes, which are responsible for skin color. The two primary types of NMSC are BCC and SCC.

1.3.1.1 Basal cell carcinoma:

BCC is another type of skin cancer. This often appears on sun-exposed areas like the face, neck, and scalp. BCC typically grows slowly and rarely spreads to other parts of the body. It originates from basal cells in the top layer of the skin, called the epidermis.

1.3.1.2 Squamous cell carcinoma:

SCC can grow more rapidly than BCC and tends to develop in sun-exposed areas. If left untreated, SCC has a higher chance of spreading to nearby lymph nodes or other organs. It originates from squamous cells in the upper layers of the skin, which is the outermost layer.

1.3.2 Melanoma Skin Cancer:

Melanoma is a very deadly type of skin cancer. It can either be malignant (spreading) or benign (not spreading). It arises from melanocytes, the skin's pigment-producing cells, and can develop anywhere on the body, even in areas not exposed to the sun. If not detected and treated early, melanoma has a high likelihood of spreading to other organs. Risk factors include exposure to UV radiation, numerous moles, fair skin prone to sunburn, and a family history of the condition. Physicians may face challenges when manually performing segmentation, potentially introducing bias in medical interpretations. After reviewing complex images, which is time-consuming, doctors often arrive at a collective diagnosis. Once melanoma has penetrated deeper into the skin or other body parts, it becomes more difficult to treat and can even be fatal. Although it is less common than non-melanoma skin cancers, its incidence has been increasing among fair-skinned populations for several years.

1.4 Models Used

Our skin melanoma detection application utilizes two CNN models for disease prediction:

- ResNet-50
- EfficientNet-B0

Deep learning architectures for image categorization include ResNet-50 and EfficientNet-B0. ResNet50 is renowned for its richness and for using skip connections to reduce problems during training. In order to balance resource consumption and performance, EfficientNet-B0 concentrates on efficient model scaling.

1.4.1 ResNet-50:

ResNet, which is an acronym for "Residual Network," is a kind of CNN architecture that was first presented in 2015. The idea of residual learning was first presented by ResNet, and it aids in solving the vanishing gradient issue in very deep networks. A particular version of the ResNet architecture with 50 layers is called ResNet-50. With respect to a number of image classification benchmarks, including the ImageNet dataset, it has demonstrated cutting-edge performance. The network includes 50 layers, including fully connected and convolutional layers, as shown by the number 50, which also denotes the network's depth.

1.4.1 EfficientNet-B0:

In 2019, the CNN architecture family EfficientNet was unveiled. The main concept underlying EfficientNet is to balance and efficiently scale the network's depth, width, and resolution, producing incredibly accurate and efficient models in the process. EfficientNet-B0 is the base model in the EfficientNet family. It is designed to provide a good balance between model size and accuracy. The "B0" in its name denotes the compound scaling coefficient for this specific variant. EfficientNet models have been known for their efficiency in terms of both computational resources and model size while achieving competitive performance on image classification tasks.

1.4.3 Ensemble Model:

We employ a weighted combination of EfficientNet-B0 and ResNet-50 in our Ensemble Model. We have used the disease prediction capabilities of two deep learning models, ResNet-50 and EfficientNetB0, in our skin melanoma detection application. By combining the predictions of these two models with a weighted average method, we have developed an ensemble model that will further improve the precision and robustness of our skin cancer detection system. Using the results of several models, ensemble learning is a potent machine learning technique that generates a single prediction or classification. In our ensemble model, we have assigned specific weights to each model's predictions, with a 60% weight given to EfficientNet-B0 and a 40% weight given to ResNet-50. This approach allowed us to emphasize the strengths of each model effectively, making our ensemble model a robust and adaptable tool for skin cancer diagnosis.

1.4 Problem Statement

Skin melanoma can be a serious medical problem and people of all ages and backgrounds can be affected by this illness. An immediate and accurate diagnosis of this disease is crutial for effective treatment and management. However, identifying skin disorders can be a difficult and challenging task, even for experienced dermatologists due to the great range of skin diseases and the similarities in their symptoms.

Modern healthcare sectors not only rely on doctor diagnosis but also on computer-aided diagnosis [18]. The biopsy method is typically used by dermatologists for diagnosing skin cancer. Through this treatment, a dermatologist removes a sample of a potentially cancerous skin lesion for testing. This procedure is painful, challenging, and time-consuming [19]. Additionally, the expertise of the dermatologist might affect how accurately these diagnoses are made, which can result in incorrect diagnoses and a delay in receiving treatment.

Our project aims to assist dermatologists in their diagnosis of skin melanoma by providing a more efficient and accurate tool. Given the hectic nature of a dermatologist's routine, our model can be used as a second opinion for the dermatologists.

Therefore, there is a need for a fast, accurate, and cost-effective system that can aid dermatologists in diagnosing skin melanoma. The use of deep learning and computer vision techniques can provide a promising solution to this problem. By analyzing skin images, a deep learning model can learn to classify skin melanoma with high accuracy and provide valuable insights to aid in the diagnosis and treatment of skin melanoma.

1.5 Relevance to Course Modules

Our final year project is related to our course modules because we studied courses like “Artificial intelligence” and “Introduction to Data Science” in our 6^{th} semester which helped un in developing our deep learning model. We also studied a course called “Web Engineering” in our 7^{th} semester which greatly helped us to understand the basics of web development and how the web works. This helped us in developing our web application.

1.6 Methodology

a. **Data Collection:** The first step in the methodology was to collect a comprehensive dataset of skin melanoma images [20]. The dataset we used was ISIC 2019. This dataset served as the basis for the deep learning model. The dataset is large enough and includes enough images to allow for accurate classification.

b. **Data Pre-processing:** After the dataset had been collected, the images were preprocessed. This involved resizing, normalizing, and augmenting the images. Resizing ensured that all images were of the same dimensions, making them easier to process. Normalization improved the quality of the images by adjusting the photos' brightness and contrast.

c. **Model Training:** Using the preprocessed dataset, a deep model was then trained. Convolutional neural networks served as the model's foundation (CNN). A portion of the data was utilized to train the model, while the remaining data was used for testing and validation. Building a model that could correctly identify skin conditions from the input photos was the aim of the training process.

d. **Model Evaluation:** Once the model was trained, its performance was evaluated using different standard metrics like confusion matrix, accuracy, precision, recall and F1Score. These metrics indicate how well the model can classify skin diseases. When the performance was not satisfactory, the model was adjusted and retrained.

e. **Model Improvement:** In this step of the methodology, the model was refined and improved. The goal was to create an accurate, efficient, and robust model.

f. **Testing:** We are conducting extensive testing on the deployed application to ensure that it performs accurately and efficiently. We are also gathering user feedback and making improvements to the application based on the feedback.

g. **Deployment:** Once we had a trained and evaluated model, we deployed it in a web application. For the web application, we used web development frameworks like Flask and Django to build a user-friendly interface that allows users to upload an image and receive a prediction about their disease.

1.7 Objectives

a. Develop a deep learning model that accurately diagnoses skin conditions from images, Web based application that allows users to easily upload the skin condition image and receive accurate diagnoses.

b. Collect and pre-process a large dataset of skin condition images to train and test EffecientNet-B0 and ResNet-50 to ensure that these models provides accurate and reliable diagnoses.

c. Utilize a deep learning approach to train the model and optimize its performance.

d. Perform comparative study.

e. Deploy the trained deep learning model on a web application, allowing users to easily input images and receive quick and accurate diagnoses.

f. Continuously collect and analyze user feedback to increase the robustness and relevance of the deep learning models and improve the user experience of the application.

1.8 Project Scope

a. To make a consolidated web-based application that should provide results when user upload their skin conditions image.

b. Predict skin melanoma disease using a deep learning model.

c. Collect and compile a large dataset of skin melanoma images from multiple sources.

d. Pre-process the images to ensure consistency and uniformity in terms of size, color, and orientation.

e. Train and test CNN models on the pre-processed dataset to classify different skin diseases accurately.

f. Use Evaluation metrics to assess the deep learning model's performance.

g. Deploy the skin disease classification system as a web application that takes in user inputs and provides a diagnosis.

1.9 Solution Application Areas

a. **Clinics and hospitals:** The system can assist healthcare providers in the accurate and timely diagnosis of skin melanoma, potentially improving patient outcomes and reducing misdiagnoses.

b. **Telemedicine platforms:** The skin melanoma classification system can be integrated into telemedicine platforms to allow remote dermatologists to accurately diagnose melanoma disease without requiring an in-person visit.

c. **Hospital Management System:** This project can be used as an inbuilt component in a Hospital Management System.

Chapter 2

Literature Review

2.1 Journal 1 [21].

2.1.1 Introduction:

The authors of this journal trained 11 CNN models to categorize lesion images as either noncancerous or cancerous. Among those 11 CNN models, a model called DenseNet169 gave the best results. So, the authors decided to use DenseNet169 for their mobile application implementation.

2.1.2 Meta Data:

Lesion Name	Images
Actinic Keratosis and Intraepithelial Carcinoma (AKIEC)	5,243
Basal Cell Carcinoma (BCC)	5,888
Melanoma (MEL)	5,959
Benign Keratosis Lesions(BKL)	5,995
Melanocytic nevus (NV)	6,705
Vascular lesion (VASC)	5,301
Dermatofibroma (DF)	4,416

Table 2.1. Meta Data of HAM10000 dataset showing total number of images belonging to each class.

2.1.3 **Methodology:** Following methodologies were used in this journal.

2.1.3.1 Data Preprocessing:

The dataset used by the authors was imbalanced, so they used data augmentation and transfer learning techniques.

2.1.3.2 Data Augmentation:

The authors used different techniques for Data augmentation.

2.1.3.3 Transfer Learning:

- The authors implemented transfer learning through fine tuning.
- They employed transfer learning from a dataset called “imageNet”.
- They also tried to train the model from scratch, but the results were poorer compared to transfer learning.

2.1.3.4 Model Development:

- The authors presented 11 CNN Models in this journal. From those 11 CNN models, a model called “DenseNet169” produced the best results.
- DenseNet169 showed a 93.27% F1 score, a 93.59% recall (sensitivity), and an accuracy of
  92.25%.
- A lite variant of DenseNet169 was employed in the creation of mobile applications.

2.1.3.5 Application Development:

- Mobile Application was developed on Android Studio 3.1.3 and Android 8.
- To make it optimized for smartphones, they converted it to a TensorFlowLite model.

2.1.4 Application Working:

- Using the application, users can take a picture of the skin lesion they wish to investigate, and then add it to the model.

2.1.5 Results:

DenseNet169 attained an F1 score of 93.27%, an accuracy of 92.25%, and a recall (sensitivity) of
93.59%.

2.2 Journal 2 (Melanoma Detection Using Deep Learning-Based Classifications [22].)

2.2.1 Introduction:

- The authors of this journal took the HAM10000 dataset, applied various preprocessing techniques on it.
- They applied segmentation to the images.
- They then trained their CNN model based on modified version of “Res-net50”.

2.2.2 Meta data:

The authors used the HAM10000 dataset for model training.

Class	Number of Training Images
Akiec	5684
Bcc	5668
Mel	5886
Vasc	5570
Nv	5979
Df	4747
Bkl	5896
Total	39430

Table 2.2 A balanced dataset as a result of using strategies for oversampling or augmentation. Segmented images were added as part of the data augmentation.

2.2.3 Methodology: Following methodologies were used in this journal.

2.2.3.1 ESRGAN Preprocessing:

The authors enhanced the picture quality of skin lesions using a preprocessing method known as ESRGAN.

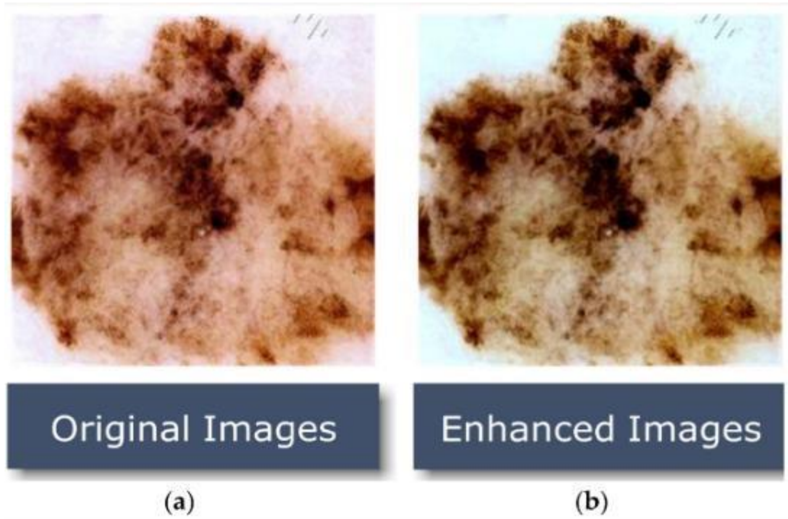

Fig 2.1. Results of the suggested image-enhancement algorithm: (a) the original image; (b) an improved version of the original image.

2.2.3.2 Image Segmentation:

The authors then applied segmentation to the lesion images. For this purpose, a ground truth maskavailable for general use from the HAM10000 dataset was applied.

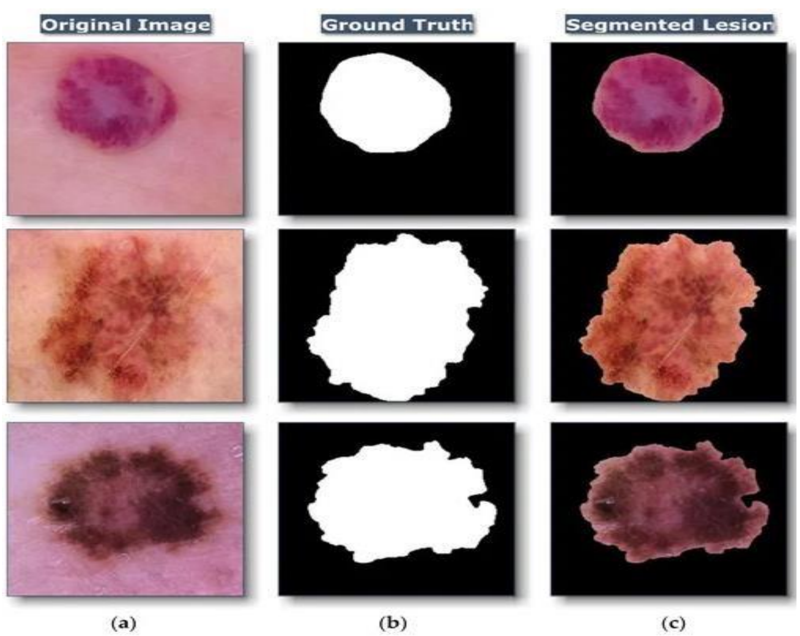

Fig 2.2. Samples of (a) original images, (b) ground truth masks, and (c) segmented lesions.

2.2.3.3 Data Augmentation:

The authors used different techniques for Data augmentation.

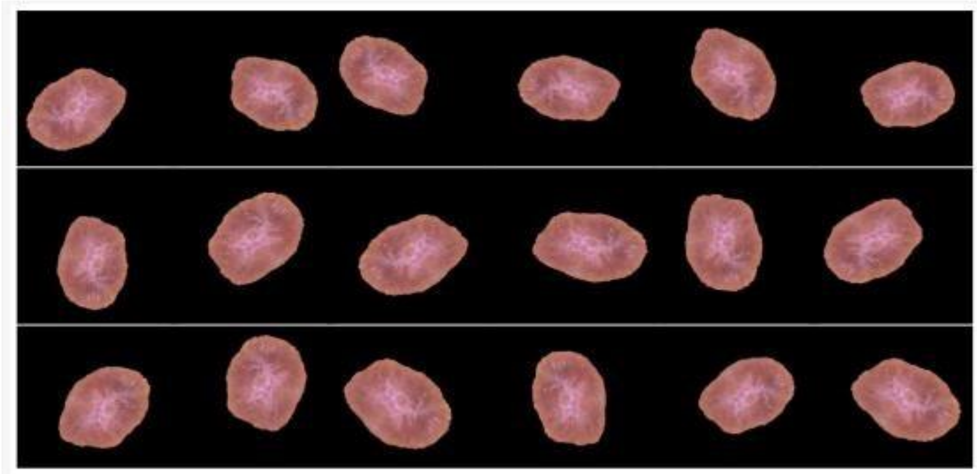

Fig 2.3. Augmentation of the same image.

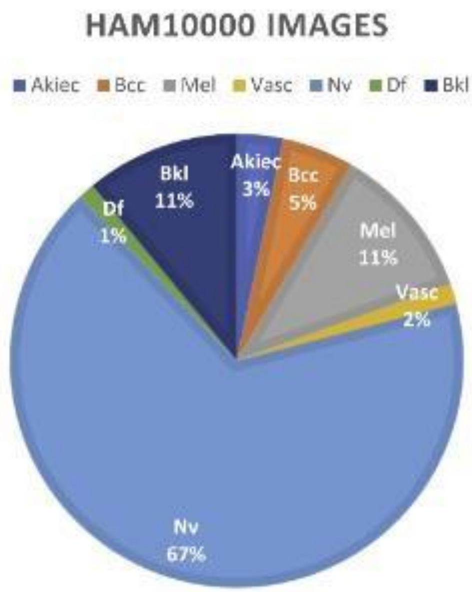

Fig 2.4. Dataset that was unbalanced prior to the use of data augmentation techniques.

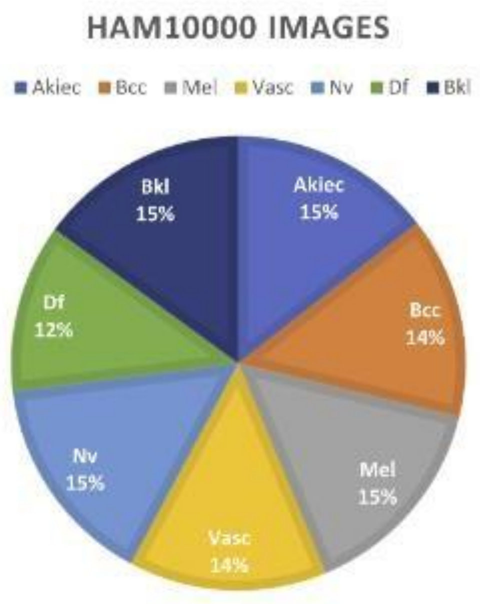

Fig 2.5. Balanced dataset following the use of tools for data augmentation.

2.2.4 Model Training:

The model trained by authors is based on some modified version of Resnet-50.

2.2.5. Results:

The proposed CNN-based model got an accuracy of 0.86, a precision of 0.84, a recall of 0.86, and an Fscore of 0.86.

2.3 Journal 3 [23].

2.3.1 Introduction:

- The authors of this journal took the ISIC dataset, applied ESRGAN technique on it to retouch and improve photos.
- They then used augmentation to distribute the data evenly.
- They used CNN to identify two main categories of cancers, benign and malignant.
- They took three CNN models and tested their accuracies.

2.3.2 Meta data:

The authors of this dataset use ISIC 2018 dataset for their model.

2.3.3 Methodology: Following methodologies were used in this journal.

2.3.3.1 ESRGAN Preprocessing:

The authors used the ERGAN technique to improve resolution of lesion images by upscaling them.

2.3.3.2 Augmentation:

Augmentation was applied to images by techniques like rotation, reflection, shifting, brightness, and resizing.

2.3.3.3 Model Development:

- A number of hyperparameter values were examined to determine how they will impact the recommended systems' effectiveness. As seen in Figure, the suggested CNN model has three layers.

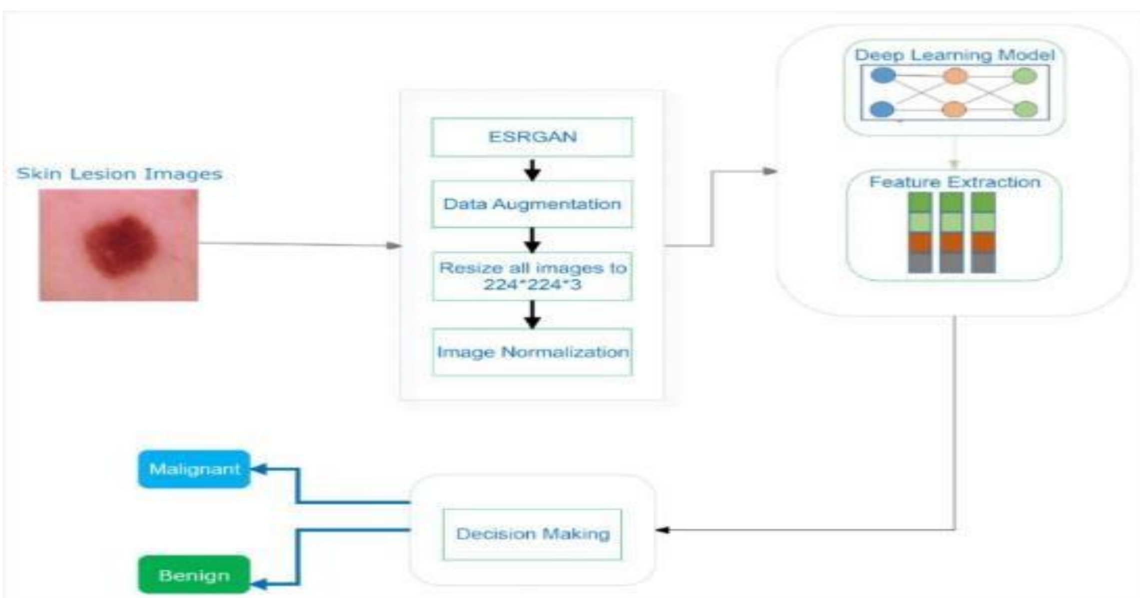

Fig 2.6. An illustration of the working of proposed system.

- The skin cancer system used a transfer learning approach, as shown in the figure..
- The three models Resnet50, Inception, and Inception Resnet50 served as the foundation for the system's architecture .

2.3.4 Results:

Reference	Dataset	Model	Accuracy
[47]	ISIC2018	VGG19_2	76.6%
[48]	ISIC2016	VGGNet	78.6%
[49]	ISBI2017	AlexNet + VGGNet	79.9%
[50]	ISIC2017	U-Net	80.0%
[51]	2-ary, 3-ary, 9-ary	DenseNet	82%
[52]	HAM10000	AlexNet	84%
[53]	HAM10000	MobileNet	83.9%
Proposed	ISIC2018	CNN	83.1%
Proposed	ISIC2018	Resnet50	83.6%
Proposed	ISIC2018	Resnet50-Inception	84.1%
Proposed	ISIC2018	Inception V3	85.7%

Table 2.3. Comparison of proposed models with existing methods.

Chapter 3

Requirement Specification

Skin detection is an essential task in various applications, such as image processing, face recognition, and content filtering. This document provides a detailed requirement specification for developing a deep learning-based skin detection system.

3.1 Functional Requirement

3.1.1 Input Management:

- The system should accept RGB images of varied resolutions.
- The system should handle multiple image formats, including JPEG, PNG, and BMP. • An option for batch processing should be available.

3.1.2 Model Training:

- The system should support model training with annotated image datasets.
- The system should support different deep learning architectures like CNN etc.
- Model checkpointing and resuming capabilities should be provided. • Real-time tracking of accuracy and loss during training should be visible.

3.1.3 Model Evaluation

- Ability to assess the model's performance using metrics such as training accuracy, Validation accuracy, training loss, validation loss etc.

3.1.4 Detection

- Model should classify given image into one of the classes.

3.1.5 Post-processing: • Option to fine-tune detection results manually.

3.1.6 Integration & Scalability:

- APIs for integrating with other applications.

- The system should support scaling to handle large datasets and high-resolution images.

3.2 Non-Functional Requirements

To consider all the targeted users, our application focuses on providing each one of the stakeholders an excellent experience with the system. For that purpose, the non-functional requirements that were considered are the following.

3.2.1 Usability

- 95% of the intended user population should express positive comments about a specific functionality while using the tools.
- Equal to or less than 5% of the intended user population should misinterpret the information provided by a display. • The average usability rating from the users should not be less than 4.

3.2.2 Performance

- When a picture of the disease is uploaded, it should not take more than 5 seconds from the time when the user uploads the picture to the time when the user gets the results over a stable machine.

3.3.3 Modifiability • Modifying any module should not take more than 2 days for a 5-member

maintainability team.

3.3.4 Consistency

- The system shall use the same color scheme for all the user interfaces in the application
- The system shall use the same font type in all the user interfaces in the application.

3.3.5 Availability

- The users should get the desired service from the system 99/100 times. It's near to impossible
  to ensure 100% availability.

3.3.6 Simplicity

- A user should be able to completely comprehend the logical flow and the design of the system
  in its first 3 uses at maximum.

3.3 Use Cases

An explanation of how users will carry out tasks on your website or application is called a use case. It describes how a system behaves in response to a request from the perspective of a user. Every use case is shown as a series of easy actions that start with the user's objective and finish when it is achieved. Use cases are valuable because they aid in clarifying the expected behavior of the system and, in the process, aid in identifying potential problems. They offer a list of objectives, and we can use this list to calculate the system's cost and complexity. Then, project teams can agree on which features should be developed as requirements.

Chapter 4

Design

4.1 CNN Architecture

Among the deep learning models, CNN are made especially for processing images. They operate by pooling layers and convolutional kernels to extract features from pictures. One pixel at a time, convolutional kernels are tiny filters that are applied to the image. The dimensionality of the feature maps generated by the convolutional layers is decreased by pooling layers. Many image recognition tasks, including object detection, segmentation, and classification, can be handled by CNNs.

4.1.1 Convolutional layer:

- CNN's fundamental building components are convolutional layers.
- They use convolutional filters or kernels to perform feature extraction from the input data.
- These filters slide across the input data (usually an image) and compute dot products to identify specific patterns or features.

4.1.2 Activation Function:

- An activation function is applied to the feature map element-wise to add non-linearity to the model after each convolutional layer..
- Rectified Linear Unit (ReLU) and its variations, which set all negative values to zero, is one of the common activation functions.

4.1.3 Pooling Layer:

- Spatial downsampling using pooling layers reduces the feature maps' spatial dimensions.
- The most popular pooling procedure, known as max pooling, extracts the maximum value from a feature map's tiny neighborhood, or pooling window.

4.1.4 Fully Connected Layer:

- For classification or regression tasks, fully connected layers are employed after the convolutional and pooling layers.
- Every neuron in these layers is linked to every other neuron in the layer above it.

- Predictions are derived from the fully connected layers' output by using the features that the preceding layers have extracted.

4.1.5 Output Layer:

- The output layer, which is the last layer in the CNN, is in charge of producing the predictions.In this layer, we apply sigmoid or softmax activation function based on the type of classification we are performing.
- If we are performing binary classification, we will use sigmoid activation function.
- In the case of multiclass classification, we will use softmax activation function.

4.1.6 Diagram:

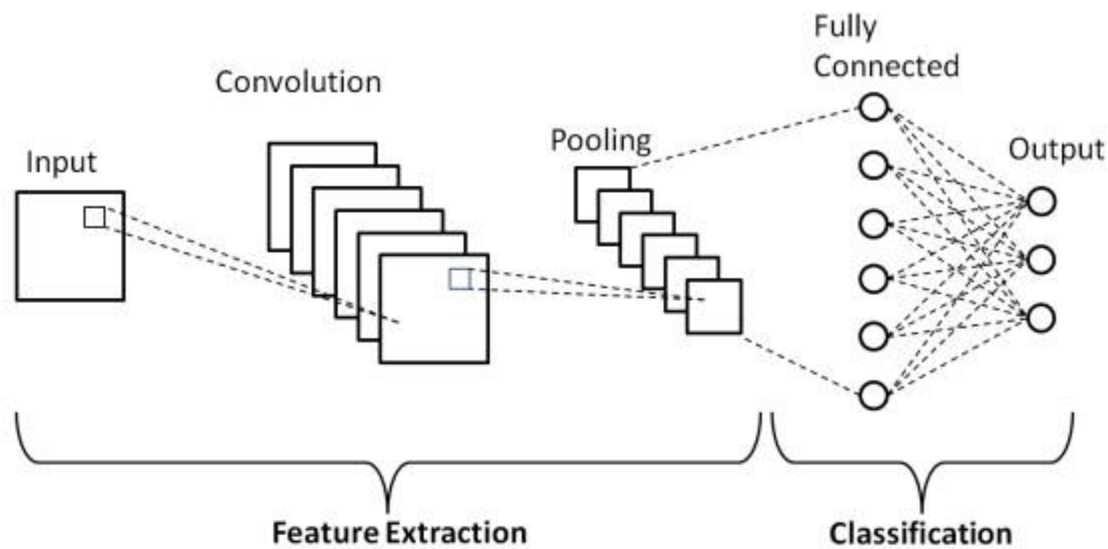

Fig 4.1. Architecture Diagram of a Convolutional Neural Network (CNN)

4.2 EfficientNet-B0 Architecture

EfficientNet-B0 is a member of the EfficientNet family, celebrated for its efficiency and effectiveness in image classification tasks.

EfficientNet-B0 is composed of:

- Convolution Layers
- Compound Scaling
- Depth-wise Separable Convolutions:
- Activation Functions
- Efficient Block Design:
- Dropout and Batch Normalization:
- Pooling Layers

- Fully Connected Layer
- Output Layer

4.2.1 Convolutional Layers:
EfficientNet-B0 utilizes a series of convolutional layers, similar to other convolutional neural networks (CNNs), to extract features from input data.

4.2.2 Compound Scaling:
EfficientNet stands out due to its compound scaling strategy, which equalizes the depth, width, and resolution of the network. By scaling the model effectively, this method strikes a fair balance between processing resources and accuracy..

4.2.3 Depth-wise Separable Convolutions:
EfficientNet-B0 uses depth-wise separable convolutions to minimize the amount of parameters and calculations. These layers produce a more efficient network architecture by combining pointwise and depth-wise convolutions..

4.2.4 Activation Functions:
Convolutional layers are followed by activation functions such as Rectified Linear Unit (ReLU) to add non-linearity to the model.

4.2.5 Efficient Block Design:
EfficientNet-B0 employs efficient block designs that balance depth and width, making it effective in handling various tasks with minimal resources.

4.2.6 Dropout and Batch Normalization:
To enhance generalization and training stability, dropout and batch normalization techniques are often integrated into the architecture.

4.2.7 Pooling Layers:
EfficientNet-B0 may include pooling layers for down-sampling, reducing the spatial dimensions of feature maps while retaining important information.

4.2.8 Fully Connected Layer:
Like other CNN architectures, EfficientNet-B0 incorporates fully connected layers towards the end of the network for classification or regression tasks.

4.2.9 Output Layer:

The final output layer of EfficientNet-B0 is responsible for generating predictions. It typically utilizes a softmax activation function for multiclass classification tasks or sigmoid activation for binary classification.

4.2.9 Diagram:

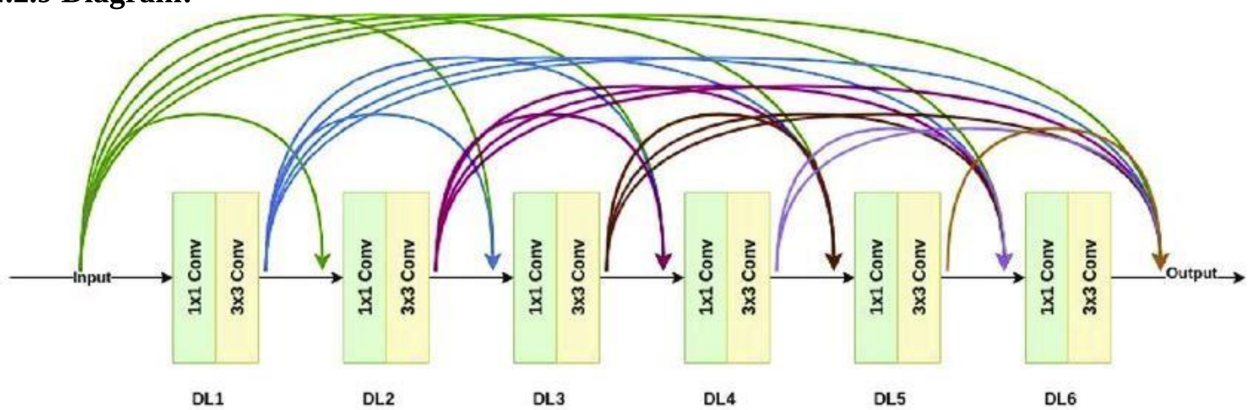

Fig 4.2. Architecture Diagram of EfficientNet-B0

4.3 ResNet-50 Architecture

ResNet-50 is a deep convolutional neural network architecture known for its innovative design. It introduces the concept of residual learning, which addresses challenges in training very deep networks. The architecture is composed of several key elements:

ResNet-50 is composed of:

- Convolution Layers
- Residual Blocks
- Bottleneck Architectures
- Activation Functions
- Pooling Layers
- Fully Connected Layer
- Output Layer

4.3.1 Convolutional Layers:

ResNet-50 includes multiple convolutional layers that extract features from the input data. These layers use convolutional filters to perform feature extraction, similar to traditional CNNs.

4.3.2 Residual Blocks:

A distinctive feature of ResNet-50 is the use of residual blocks. These blocks contain shortcut connections that allow the network to learn residual functions, making it easier to train deep networks.

4.3.3 Bottleneck Architectures:

ResNet-50 employs bottleneck architectures in some of its layers to reduce computational complexity. These bottleneck layers consist of 1x1, 3x3, and 1x1 convolutions, enabling efficient feature extraction.

4.3.4 Activation Functions:

Activation functions like ReLU (Rectified Linear Unit) are applied within the residual blocks and after each convolutional layer to add non-linearity to the model.

4.3.5 Pooling Layers:

Max pooling and average pooling layers are used for spatial down-sampling, reducing the spatial dimensions of feature maps in ResNet-50.

4.3.6 Fully Connected Layer:

The fully connected layers of ResNet-50 are in charge of classification after the convolutional and pooling layers. By creating connections between each neuron in a layer and every other layer's neuron, these layers allow the network to use the features it has learnt to create predictions.

4.3.7 Output Layer:

The output layer of ResNet-50 is responsible for generating predictions. It typically uses a softmax activation function, especially in the case of multiclass classification, to produce class probabilities.

4.3.8 Diagram:

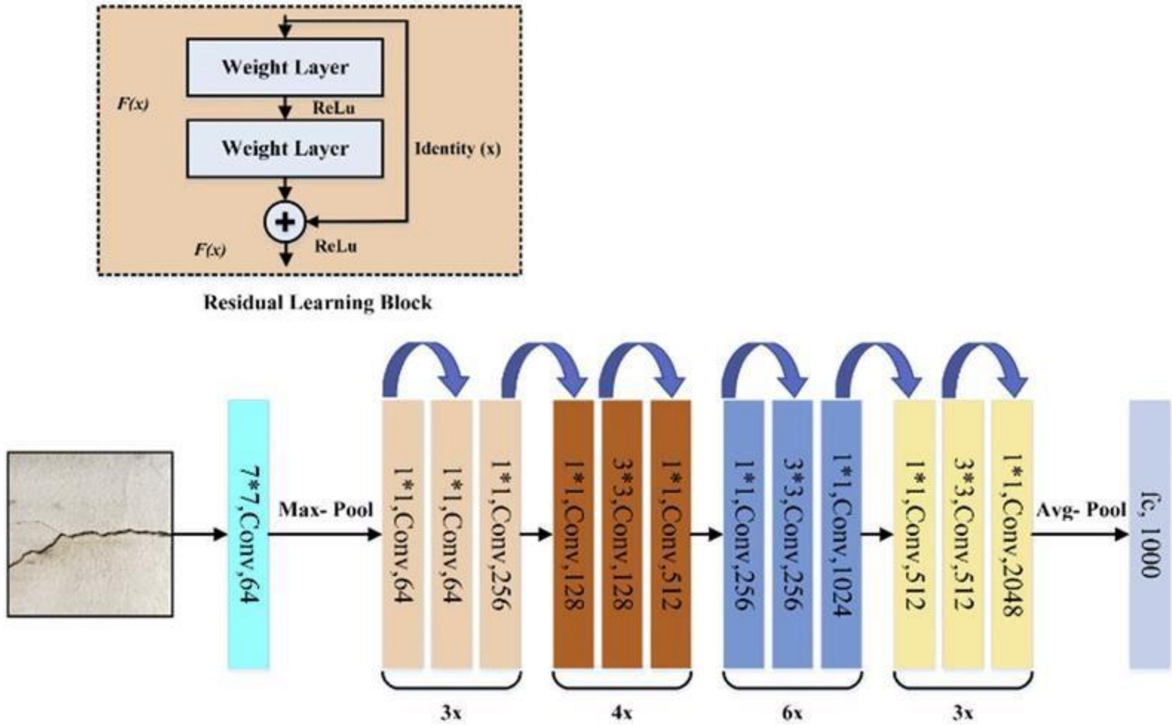

Fig 4.3. Architecture Diagram of ResNet-50

4.4 Application User Interface

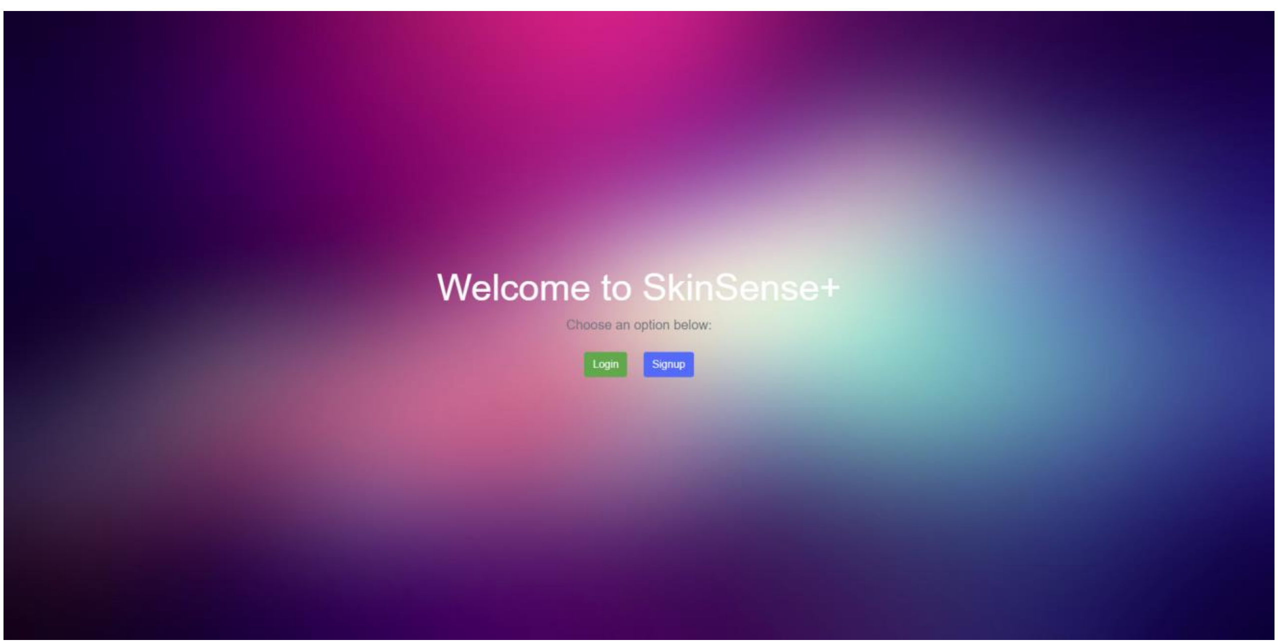

Fig 4.4. Application Dashboard Before Login

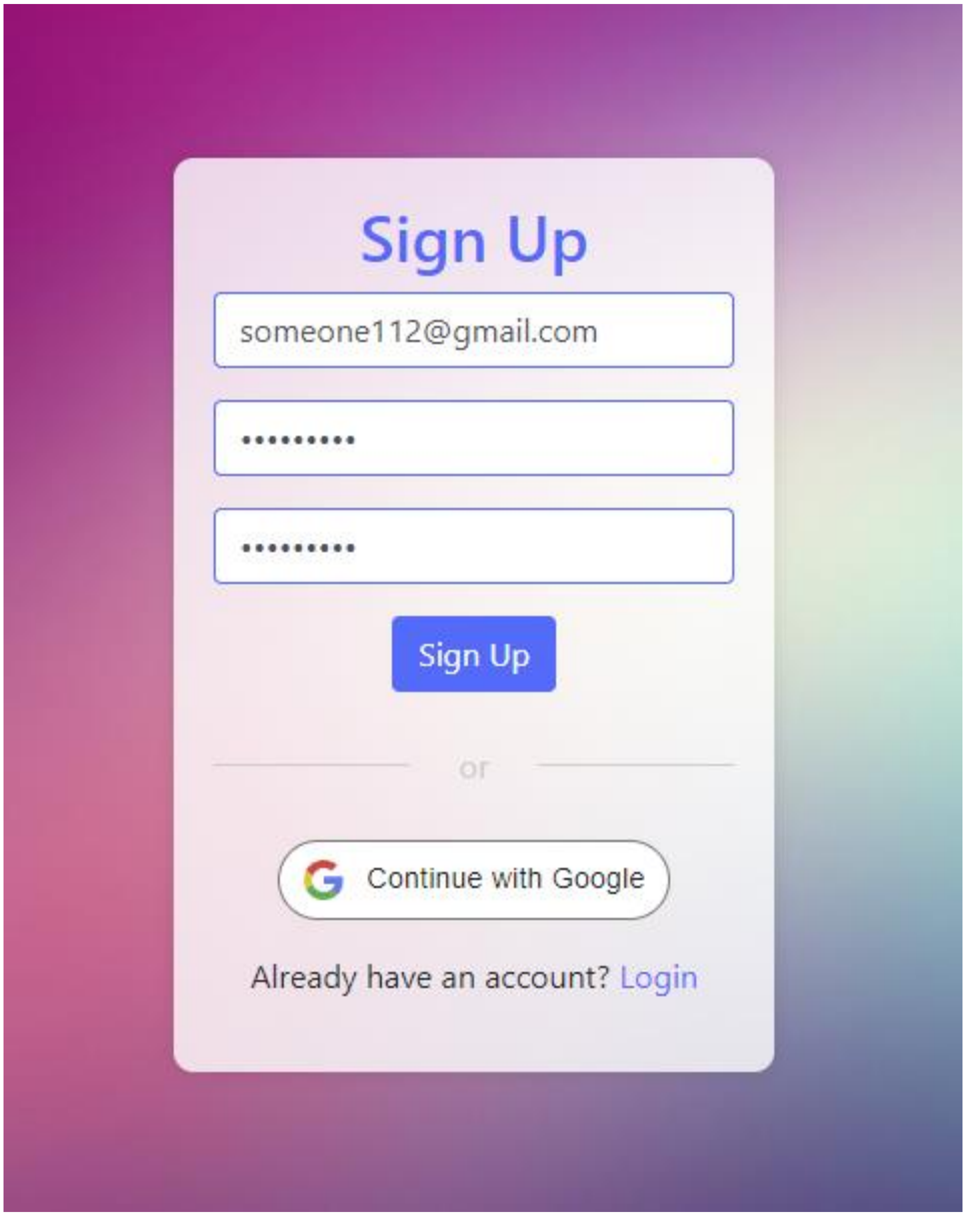

Fig 4.5. Application Signup Page

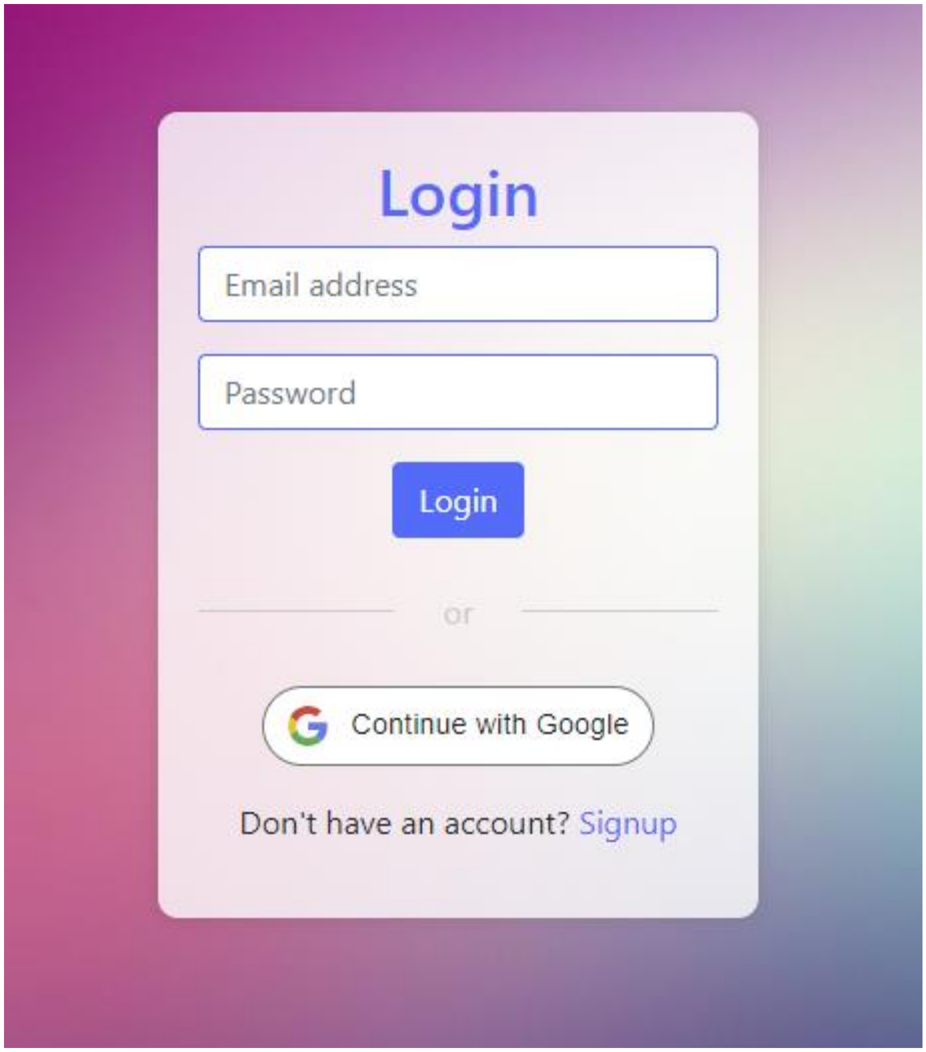

Fig 4.6. Application Login Page

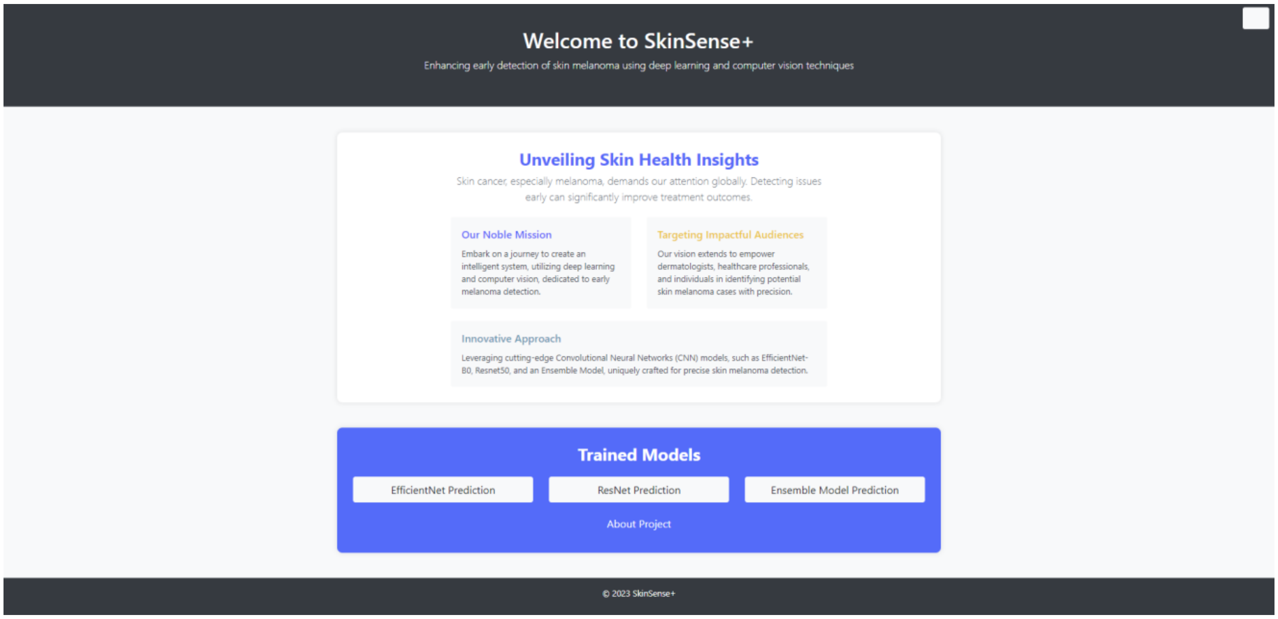

Fig 4.7. Application Dashboard After Login

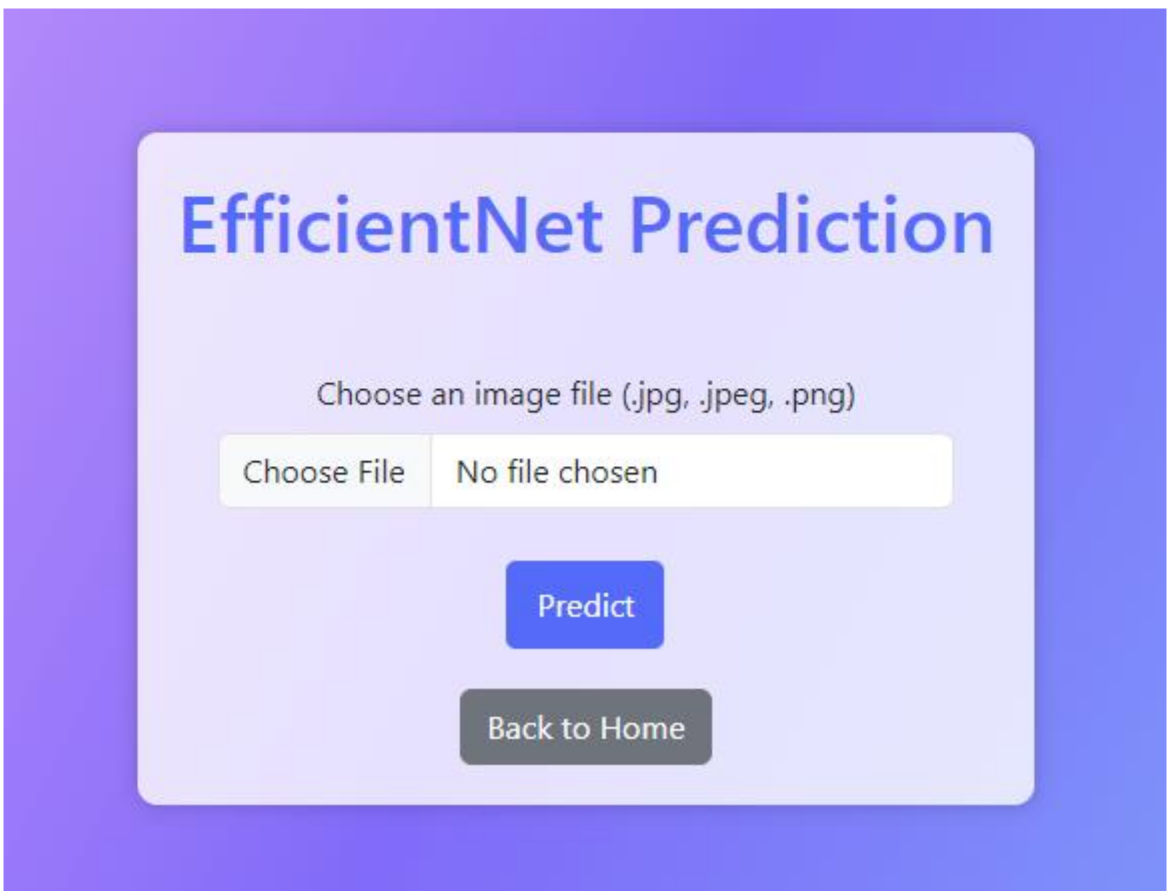

Fig 4.8. EfficientNet Prediction Page

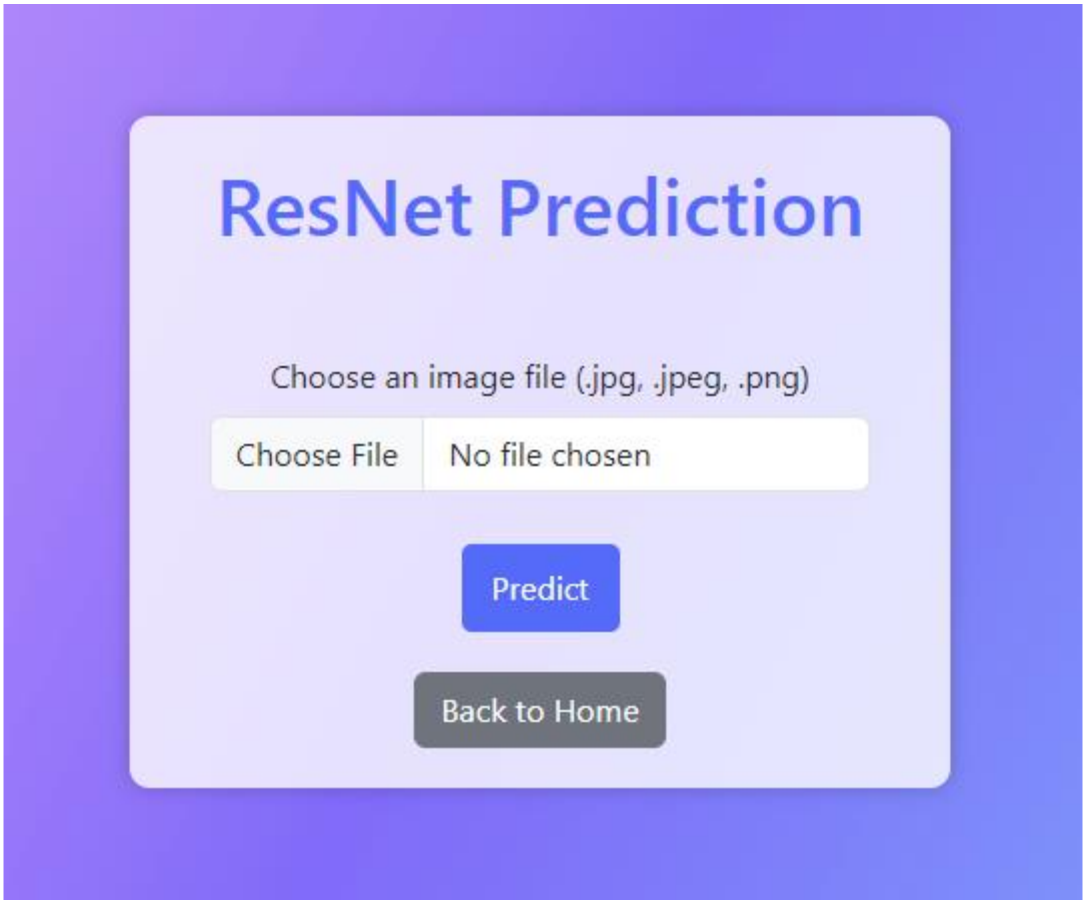

Fig 4.9. ResNet Prediction Page

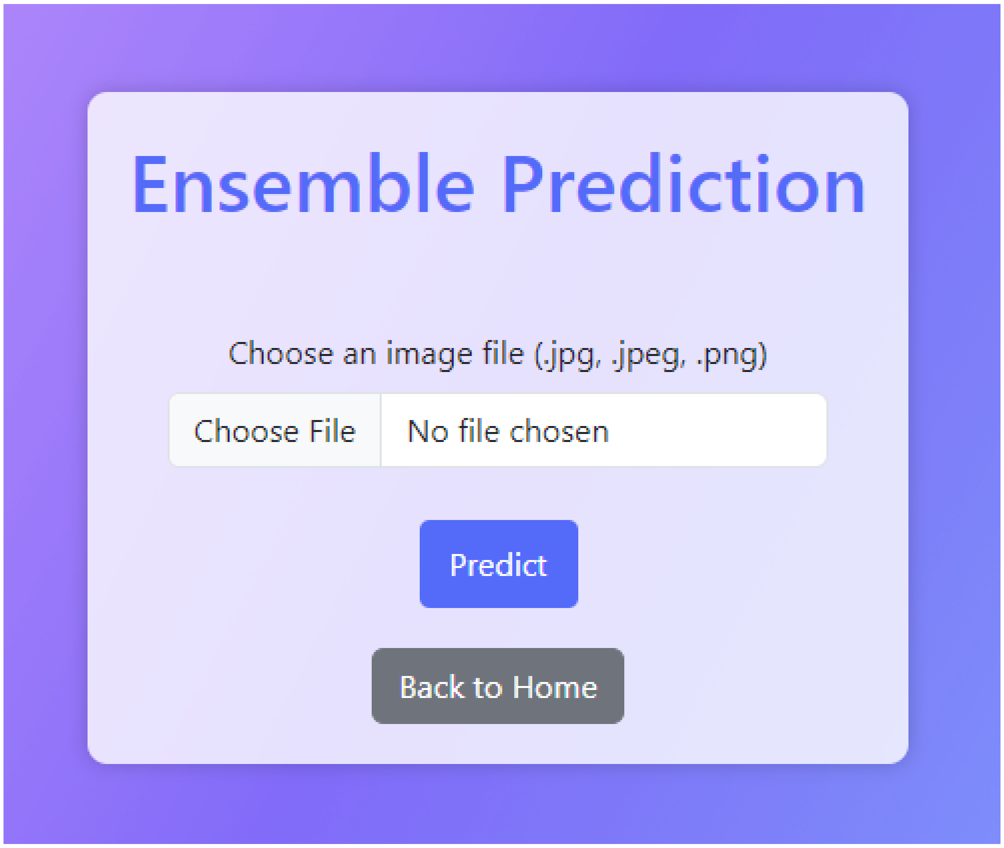

Fig 4.10. Ensemble Model Prediction Page

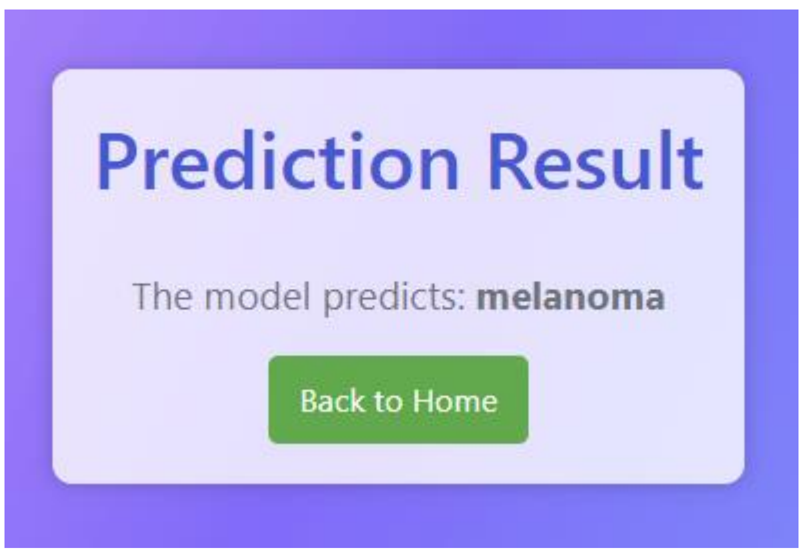

Fig 4.11. Prediction Result Page

Chapter 5

Implementation

5.1 Dataset Augmentation

Data augmentation is a fundamental step in enhancing the performance of machine learning models, including those used for skin cancer classification in this project. By creating variations of the existing training data, we aimed to improve the model's ability to recognize and classify different skin conditions.

5.1.1. Data Augmentation Techniques:

The data augmentation process leveraged a range of augmentation techniques, including rotation, shifting, zooming, flipping, and more. These techniques introduced diversity into the training dataset, which is crucial for training models capable of handling diverse input scenarios and generalizing effectively to unseen data.

5.1.2. Advantages of Data Augmentation:

Data augmentation provides multiple advantages in the context of skin cancer classification:

5.1.2.1 Mitigating Data Imbalance:

Data augmentation helps address data imbalance issues commonly encountered in medical datasets, where some classes may have limited samples. By generating additional training examples, we contribute to balanced and reliable model training.

5.1.2.2 Enhancing Model Robustness:

The augmented data exposes the model to a broader range of image variations, improving its resilience to differences in angles, scales, orientations, and other factors.

```
Training Images:
Basal cell carcinoma 2820
Melanoma 3812
Squamous cell carcinoma 541
```

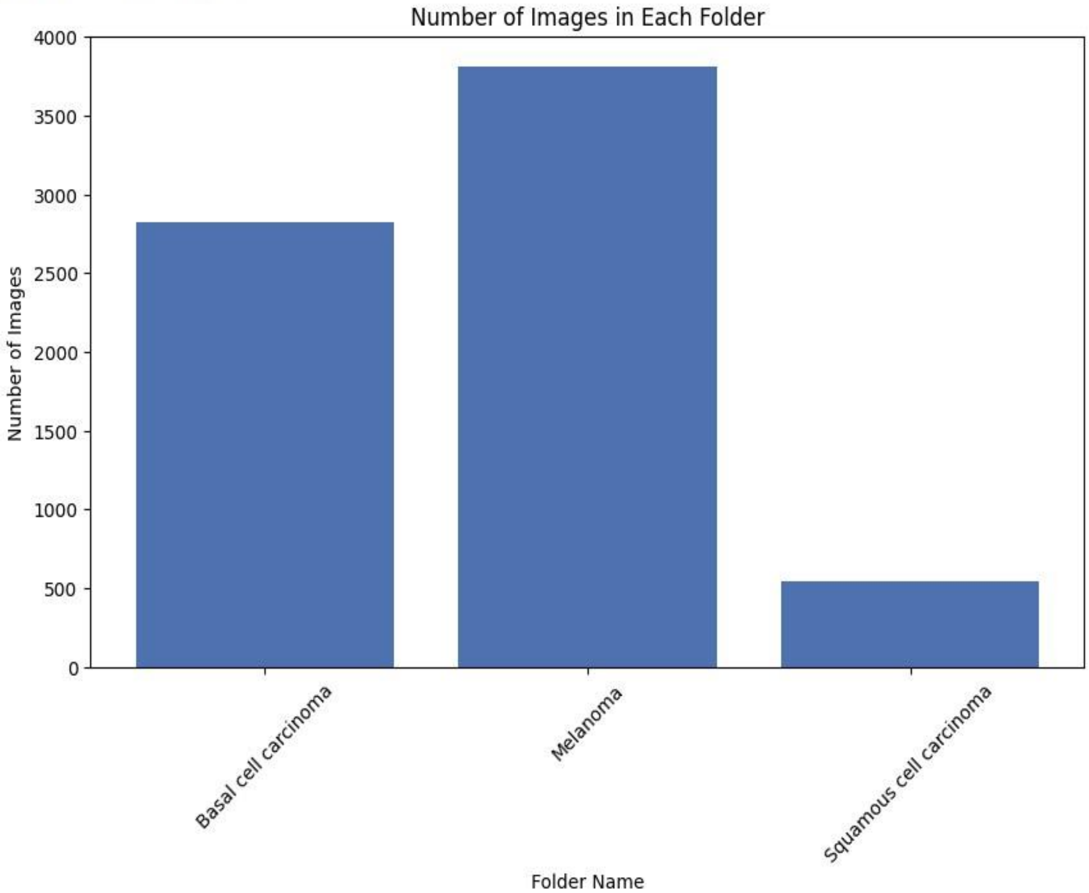

Fig 5.1. Image Distribution of classes before Data Augmentation

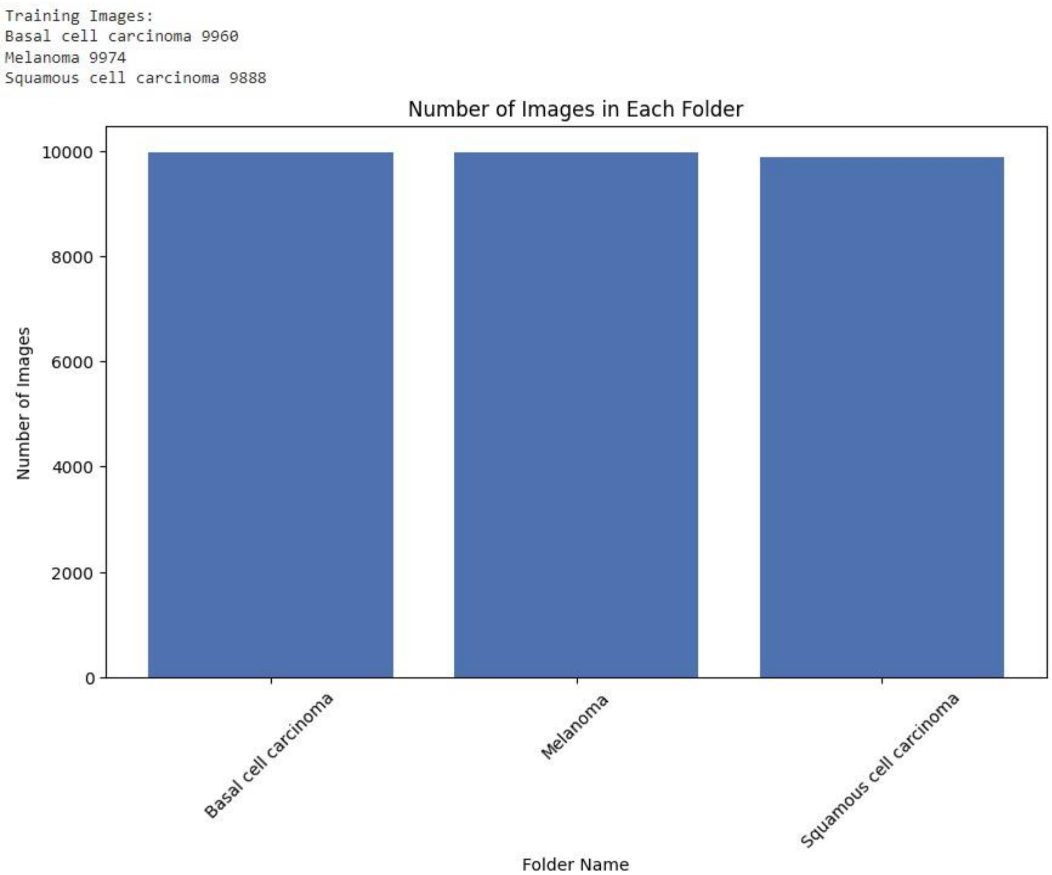

Fig 5.2. Image Distribution of classes after Data Augmentation

5.2 Algorithm

In this chapter, we delve into the implementation details of the skin melanoma detection system using our CNN models. We outline the major components and functions that drive this application.

5.1.1 Disease Prediction using CNNs:

CNNs are the backbone of our skin melanoma detection system. CNNs are a class of deep learning models specifically designed for tasks like image classification. They are inspired by the human visual system, where neurons in the visual cortex respond to visual stimuli in a restricted region of the visual field, known as the receptive field.

5.1.2 CNN Models:

We have developed CNN models (EfficientNet-B0, ResNet-50, Ensemble Model) for skin melanoma detection. These models are tailored for the task of skin melanoma detection. They efficiently process input images and provide accurate results.

5.2 Authentication

5.2.1 Login/Registration:

In this implementation, we have chosen to incorporate an authentication system into the skin melanoma detection web application. This means that users will need to log in or register to access the features of the application.

5.3 End User

1. The user accesses the web application and login or sign up.
2. The user uploads an image for skin melanoma detection.
3. The user chooses the model which he/she needs to use for prediction.
4. The application processes the image using trained models, our dedicated CNN models. 5. The detection results are displayed to the user.

5.4 Technologies Used

5.4.1 Google Colab:

Google Colab was chosen for its robust infrastructure, providing an efficient environment for training deep learning models. This cloud-based platform grants easy access to powerful GPUs and TPUs, significantly accelerating model training.

5.4.2 Streamlit:

The project utilized the Streamlit web framework to build a user-friendly web application for skin melanoma detection. Streamlit enabled the creation of web pages, making the model's capabilities accessible to users through a web interface.

5.4.3 Anaconda Navigator (JupyterLab):

Anaconda Navigator, especially in conjunction with JupyterLab, was instrumental in running the Flask web application. This combination provided a well-structured, interactive development environment, streamlining web application development and ensuring smooth operation.

5.4.4 TensorFlow:

TensorFlow, a cornerstone of deep learning, played a pivotal role in this project. It was the core library for designing, training, and deploying the deep learning models essential for skin melanoma detection. TensorFlow's versatility and performance optimization contributed to the project's success.

5.4.5 Matplotlib:

Matplotlib served as a valuable tool for generating data visualizations, charts, and plots. These visual aids were instrumental in assessing model performance, understanding data distribution, and enhancing the overall project analysis.

5.4.6 PIL (Python Imaging Library):

The Python Imaging Library was employed for image processing and manipulation tasks. It facilitated image data preparation, ensuring that the input data met the model's requirements and effectively processed the images to ensure accurate detection.

5.4.7 Keras:

Keras, integrated within the TensorFlow ecosystem, streamlined the process of building and configuring neural network architectures. Its high-level, user-friendly interface made designing complex models straightforward, contributing to the efficient development of the melanoma detection model.

5.5 Languages Used

5.5.1 Python:

Python served as the project's primary backend language. It was used for implementing the deep learning models, handling data, and training the models. Python's simplicity and a rich ecosystem of machine learning libraries make it an ideal choice for developing the core functionality of the melanoma detection system.

5.5.2 HTML, CSS, and JavaScript:

These web technologies were employed for building the user interface of the web application. HTML structured the content, CSS provided styling, and JavaScript added interactivity. This combination created an engaging and user-friendly web interface for users to interact with the melanoma detection system.

5.5.2 Streamlit (Python):

Streamlit, a Python web framework, played a pivotal role in building the web application's back end. It facilitated the integration of the deep learning model into the web application, ensuring that users could easily access and interact with the melanoma detection system.

5.6 Evaluation Metrics Used

By comparing predictions to actual class labels, measurement metrics are used to assess the performance and efficacy of classification models. The particulars of the categorization problem and the intended trade-offs between various performance factors determine which assessment metrics should be used. A fundamental idea in machine learning, the confusion matrix allows us to display widely used assessment metrics for classification while also providing comprehensive information about the actual and expected classifications made by a classification system. There are two dimensions to this matrix: one shows the item's true class, and the other shows its predicted class. To show the expected and actual classifications, we use a 2x2 confusion matrix. True positives (TP) represent successfully anticipated events, false positives (FP) represent wrongly predicted events, true negatives (TN) represent correctly predicted events, and false negatives (FN) represent incorrectly predicted events in this table.

Actual Values

		Positive	Negative	Total
Predicted Values	Positive	TP	FP	TP + FP
	Negative	FN	TN	FN + TN
	Total	TP + FN	FP + TN	

5.6.1 Accuracy

Accuracy is used to evaluate the performance of a machine-learning model. The success rate of the model is defined as the ratio of its accurate predictions to all of its previous forecasts. By dividing the total number of predictions by the percentage of true positive and true negative predictions, one can calculate the accuracy of binary classification problems.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

5.6.2 Sensitivity

Sensitivity is a measure of a model’s accuracy in identifying positive cases. It is calculated by dividing the total number of true positive data cases by the number of cases in which the model accurately predicted the positive class. A high sensitivity score indicates that the model is good at spotting positive circumstances and has a low rate of false negatives. In some situations when the cost of missing a positive instance is substantial, sensitivity is a valuable statistic. For instance, it is important for medical diagnoses to have high sensitivity to guarantee that all positive instances are found.

$$Sensitivity = \frac{TP}{TP + FP}$$

5.6.3 Specificity

Specificity describes the extent to which a model or system can accurately identify a certain class or category based on a set of inputs. In terms of machine learning, Specificity is a model's capacity to detect negative occurrences, High specificity means that there are few false positive errors, meaning that the model is good at avoiding false alarms.

$$Specificity = \frac{TN}{TN + FP}$$

5.6.4 F1 Score

The F1 score, which takes into account both precision and recall, is used to evaluate the accuracy of a model. The harmonic mean of recall and accuracy, it offers a harmony between the two measures. The F1 score is frequently used as a statistic to calculate the effectiveness of model. Its range is 0 to 1, with 1 reflecting perfect accuracy.

$$F1 - Score = \frac{(2 * P) * (SNS)}{P + SNS}$$

5.6.5 Precision

Precision is a statistic used in information retrieval and machine learning that is used to evaluate the accuracy of a classifier or retrieval system. The percentage of relevant occurrences among the recovered instances is computed by dividing the total number of true positive instances by the total number of false positive instances.

$$Precision = \frac{TP}{TP + FP}$$

5.6.6 Error Rate

The ratio of inaccurate predictions to all of the predictions produced is known as the error rate, which serves as a gauge of a model's or system's accuracy. A frequently employed statistic to evaluate the performance of a classifier is the number of errors (false positives plus false negatives) divided by the total number of occurrences.

$$ER = \frac{FP + FN}{TP + TN + FP + FN}$$

Chapter 6

System Testing and Evaluation

6.1 Testing Techniques

Several testing techniques were employed to evaluate our system thoroughly. These techniques help
us ascertain the functionality, performance, usability, and security of the application:

6.1.1 Functional Testing:
This testing focuses on the core functionality of the system, ensuring that it correctly identifies and classifies melanoma in skin images.

6.1.2 Performance Testing:
The performance of the system was evaluated to determine its speed, responsiveness, and stability. This helped us understand how the system behaves under different conditions.

6.1.3 Usability Testing:
Usability testing was conducted to identify design and user interface issues. It aimed to verify if users could perform desired tasks successfully and whether they were satisfied with the application.

6.1.5 Integration Testing:
We assessed how well different components of the system work together to ensure they interact seamlessly.

6.1.6 End-to-End Testing:
This testing verified the complete user journey, from image upload to prediction display, to ensure all components work harmoniously.

6.2 Functional Testing

Functional testing focused on ensuring the core features of our melanoma detection system were functioning correctly. Here's an example test case:

6.2.1 Test Case:

The following test case illustrates our approach to assessing the functionality of the application:

Test Case ID	Test Case 01
Description	Testing the system's core functionality
Initial Conditions	The system is ready for testing
Steps	Tasks and expected results
1	Open the web application. PASS
2	Verify all tabs are working. PASS
3	Check if the user has an upload image option. PASS
4	Check if the image is properly uploaded. PASS
5	Check if the prediction is displayed to the user. PASS

Table 6.1. Functional Test Case

6.3 Performance Testing

Performance testing is a crucial aspect of evaluating our system. It encompasses the processes of assessing system speed, responsiveness, and stability.

6.3.1 Test Case:

The following test case illustrates our approach to assessing the performance of the application:

Test Case ID	Test Case 02
Description	Evaluating the performance of the application
Initial Conditions	The system is prepared and equipped for testing
Steps	Tasks and expected results

1	Open the application. PASS
2	Verify that loading time of application is minimum.PASS
3	Verify that response time of application to user input takes minimum time. PASS

Table 6.2. Performance Test Case

6.4 Usability Testing

Usability testing aimed to enhance the user experience by identifying design and interface issues. An example test case is as follows:

6.4.1 Test Case:

The following test case illustrates our approach to assessing the performance of the application:

Test Case ID	Test Case 03
Description	Evaluating the usability of the application
Initial Conditions	The system is set up for testing
Steps	Tasks and expected results
1	Users find the buttons and icons relatable. PASS
2	All users can smoothly move between screens. PASS
3	The system accepts image data without errors. PASS
4	Users can successfully perform their intended actions. PASS

Table 6.3. Usability Test Case

Chapter 7

Conclusion

7.1 Results:

7.1.1 EfficientNet-B0:

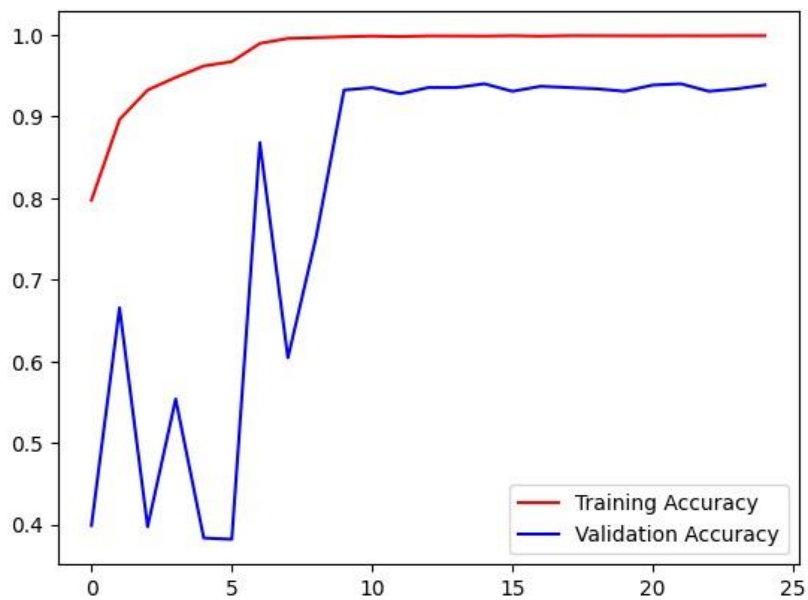

Fig 7.1. EfficientNet-B0: Graph between Training Accuracy and Validation Accuracy

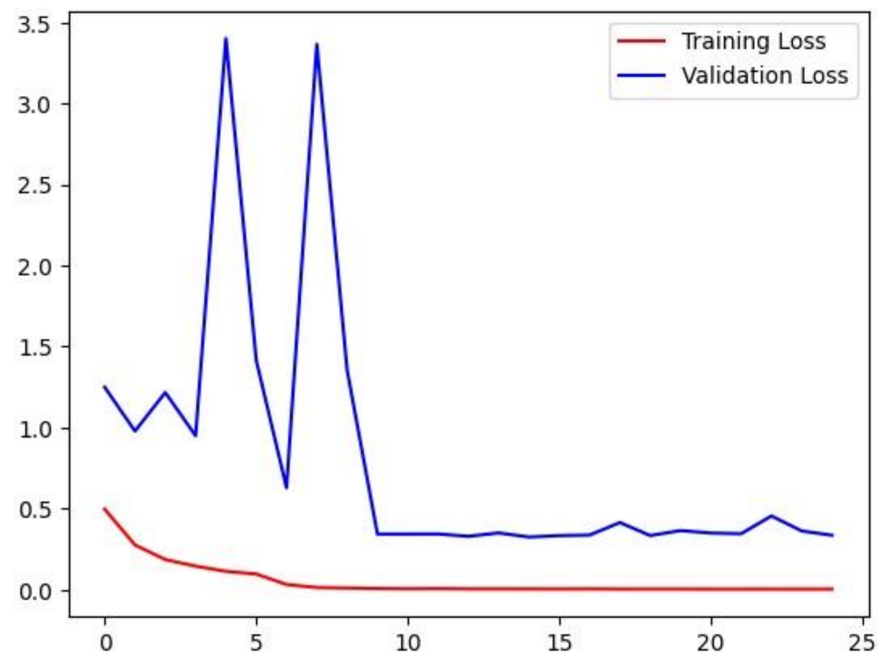

Fig 7.2. EfficientNet-B0: Graph between Training Loss and Validation Loss

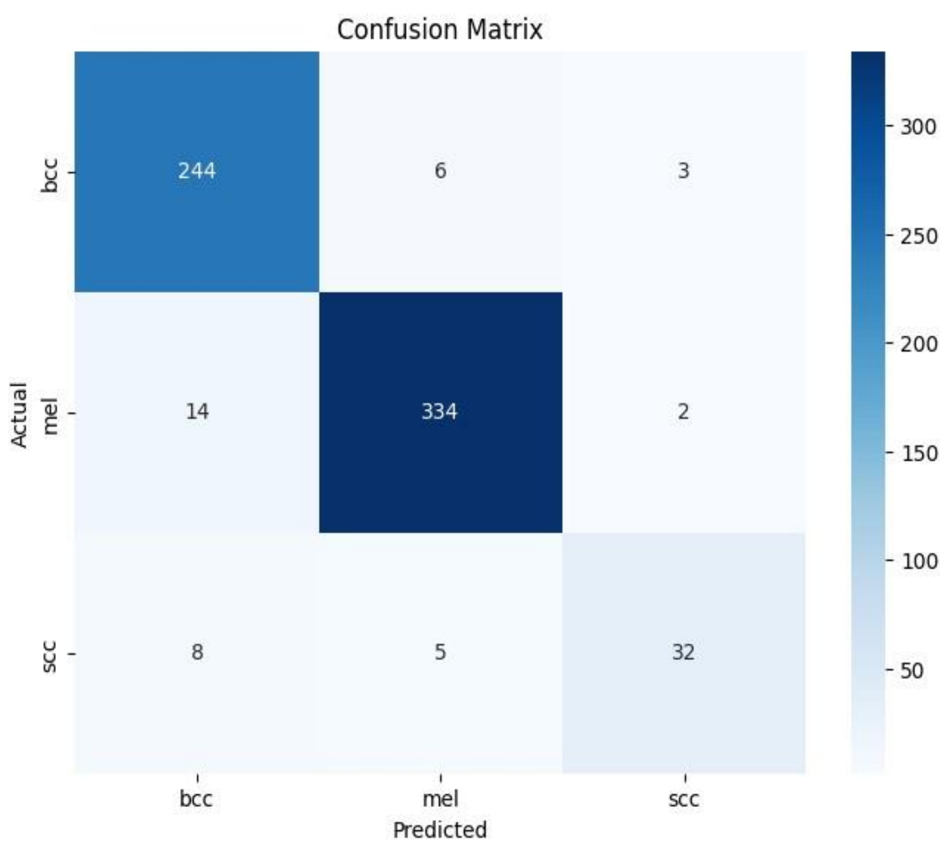

Fig 7.3. EfficientNet-B0: Confusion Matrix

7.1.1 ResNet-50:

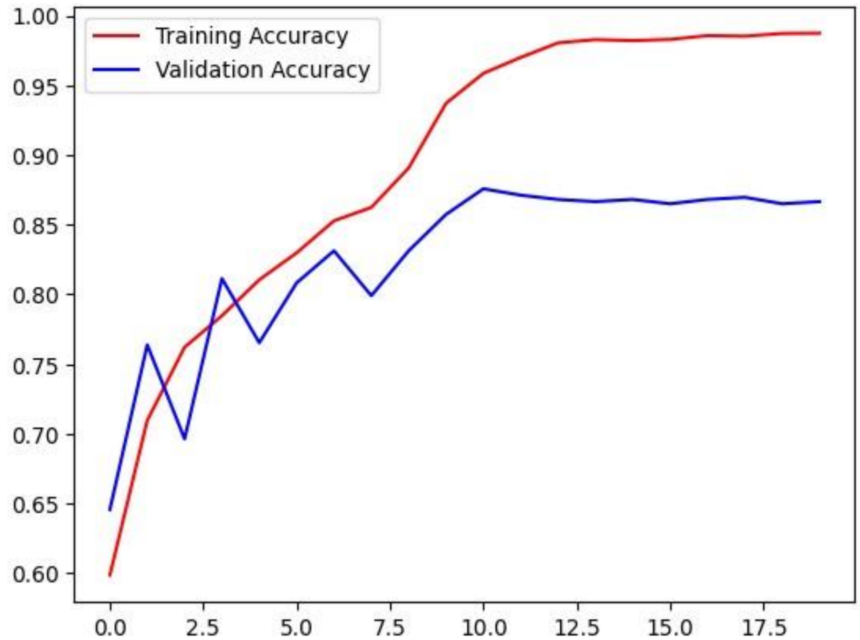

Fig 7.4. ResNet-50: Graph between Training Accuracy and Validation Accuracy

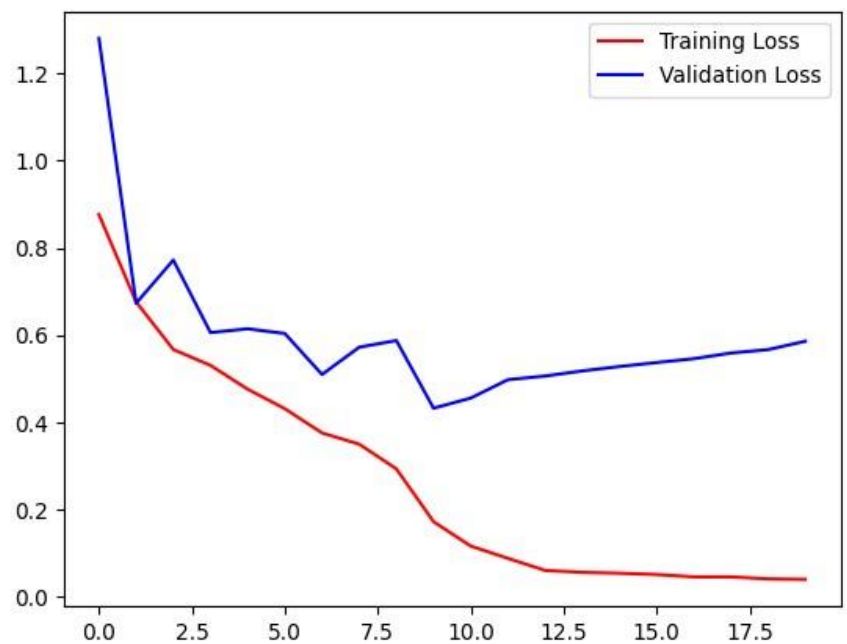

Fig 7.5. ResNet-50: Graph between Training Loss and Validation Loss

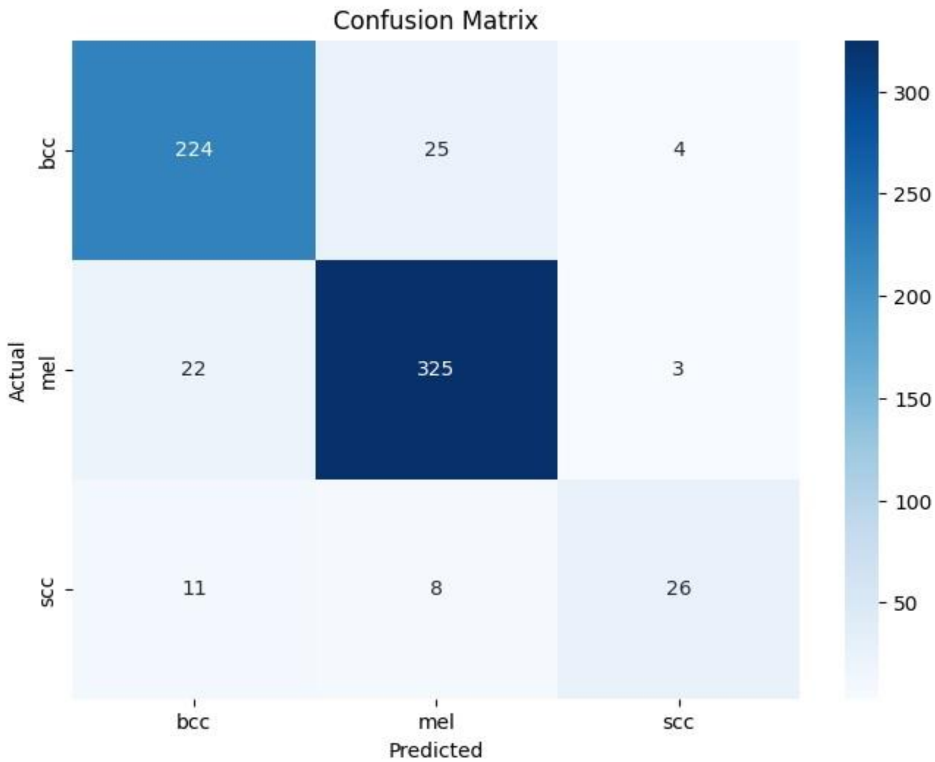

Fig 7.6. ResNet-50: Confusion Matrix **7.1.3**

Ensemble Model:

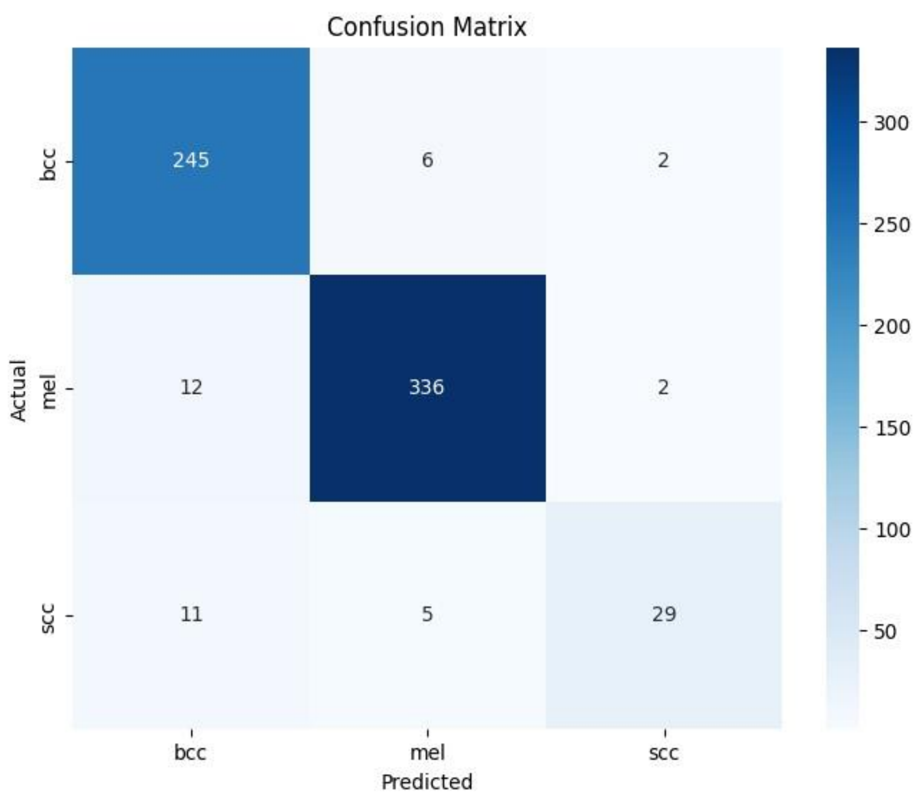

Fig 7.7. Ensemble Model: Confusion Matrix

Metrics	EfficientNet-B0	ResNet-50	Ensemble Model
Accuracy	0.97	0.90	0.97
Sensitivity	0.96	0.90	0.97
Specificity	0.98	0.91	0.98
F1-Score	0.97	0.92	0.97
Precision	0.98	0.93	0.98
Error Rate	0.03	0.09	0.03

Table 7.1. Performance of Neural Networks on Melanoma skin cancer image data set.

7.2 Conclusion

Our skin melanoma detection system is a comprehensive solution designed to harness cutting-edge technology for early skin cancer diagnosis. This system empowers users with the ability to detect potential skin issues, specifically focusing on melanoma, a dangerous form of skin cancer. It provides an efficient means of assessing skin lesions and delivers insights that can assist users in understanding the nature of the detected condition. Our application employs deep learning techniques to analyze skin images, offering users an initial assessment of their skin's health. By identifying potential skin abnormalities and melanoma symptoms, our system promotes early detection and prevention. We have streamlined the process of skin analysis, ultimately enhancing the user's awareness and facilitating timely medical intervention when necessary. Additionally, our system caters to the needs of both professionals and individuals concerned about their skin health. It bridges the gap between users and healthcare providers, enabling them to collaborate on the path to early diagnosis and treatment. Moreover, it maintains records of skin images and analysis results, ensuring that users can easily track changes in their skin over time.

7.3 Future Work

While our system is a significant step towards efficient skin melanoma detection, there are several areas that offer opportunities for future enhancements:

- **Expanding the system's capabilities** to detect a broader range of skin lesions and conditions by researching and incorporating additional datasets.
- **Exploring methods for real-time video or stream processing** to further enhance the application's capabilities. This expansion may require adjustments to the application's architecture and potentially greater computational resources.

- **Considering the development of a more user-friendly interface** to reach a broader audience. Making the application more accessible and intuitive will broaden the user base and ensure a more comprehensive impact in the future.

References

[1] Daghrir, Jinen, et al. "Melanoma skin cancer detection using deep learning and classical machine learning techniques: A hybrid approach." 2020 5th international conference on advanced technologies for signal and image processing (ATSIP). IEEE, 2020.

[2] Byrd, Allyson L., Yasmine Belkaid, and Julia A. Segre. "The human skin microbiome." Nature Reviews Microbiology 16.3 (2018): 143-155.

[3] Li, Yuexiang, and Linlin Shen. "Skin lesion analysis towards melanoma detection using deep learning network." Sensors 18.2 (2018): 556.

[4] Nasr-Esfahani, Ebrahim, et al. "Melanoma detection by analysis of clinical images using convolutional neural network." 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2016.

[5] R. A. Abumalloh, M. Nilashi, M. Y. Ismail, A. Alhargan, A. Alghamdi, A. O. Alzahrani, L. Saraireh, R. Osman, and S. Asadi, “Medical image processing and covid-19: a literature review and bibliometric analysis,” Journal of infection and public health, vol. 15, no. 1, pp. 75–93, 2022.

[6] P. Malhotra, S. Gupta, D. Koundal, A. Zaguia, W. Enbeyle, et al., “Deep neural networks for medical image segmentation,” Journal of Healthcare Engineering, vol. 2022, 2022.

[7] T. Saba, S. Al-Zahrani, and A. Rehman, “Expert system for offline clinical guidelines and treatment,” Life Science Journal, vol. 9, no. 4, pp. 2639–2658, 2012.

[8] S. Pang, X. Yang, et al., “Deep convolutional extreme learning machine and its application in handwritten digit classification,” Computational intelligence and neuroscience, vol. 2016, 2016.

[9] P. Dhruv and S. Naskar, “Image classification using convolutional neural network (cnn) and recurrent neural network (rnn): A review,” Machine Learning and Information Processing: Proceedings of ICMLIP 2019, pp. 367–381, 2020.

[10] F. Shamshad, S. Khan, S. W. Zamir, M. H. Khan, M. Hayat, F. S. Khan, and H. Fu, “Transformers in medical imaging: A survey,” Medical Image Analysis, p. 102802, 2023.

[11] X. Liu, L. Song, S. Liu, and Y. Zhang, “A review of deep-learning-based medical image segmentation methods,” Sustainability, vol. 13, no. 3, p. 1224, 2021.

[12] F. Lateef and Y. Ruichek, “Survey on semantic segmentation using deep learning techniques,” Neurocomputing, vol. 338, pp. 321–348, 2019.

[13] K. Das, C. J. Cockerell, A. Patil, P. Pietkiewicz, M. Giulini, S. Grabbe, and M. Goldust, “Machine learning and its application in skin cancer,” International Journal of Environmental Research and Public Health, vol. 18, no. 24, p. 13409, 2021.

[14] H. M. Gloster Jr and K. Neal, “Skin cancer in skin of color,” Journal of the American Academy of Dermatology, vol. 55, no. 5, pp. 741–760, 2006.

[15] L. Zeng, B. J. Gowda, M. G. Ahmed, M. A. Abourehab, Z.-S. Chen, C. Zhang, J. Li, and P. Kesharwani, “Advancements in nanoparticle-based treatment approaches for skin cancer therapy,” Molecular Cancer, vol. 22, no. 1, p. 10, 2023.

[16] C. Fitzmaurice, D. Abate, N. Abbasi, G. B. of Disease Cancer, et al., “Global, regional, and national cancer incidence, mortality, years of life lost, years lived with disability, and disability-adjusted life-years for 29 cancer groups, 1990 to 2017: a systematic analysis for the global burden of disease study,” JAMA Oncology, vol. 5, pp. 1749–1768, 2019.

[17] P. Boyle, B. Levin, et al., World cancer report 2008. IARC Press, International Agency for Research on Cancer, 2008.

[18] Daghrir, Jinen, et al. "Melanoma skin cancer detection using deep learning and classical machine learning techniques: A hybrid approach." 2020 5th international conference on advanced technologies for signal and image processing (ATSIP). IEEE, 2020.

[19] Dildar, Mehwish, et al. "Skin cancer detection: a review using deep learning techniques." International journal of environmental research and public health

18.10 (2021): 5479.

[20] Vijayalakshmi, M. M. "Melanoma skin cancer detection using image processing and machine learning." International Journal of Trend in Scientific Research and Development (IJTSRD) 3.4 (2019): 780-784.

[21] Kousis, Ioannis, et al. "Deep learning methods for accurate skin cancer recognition and mobile application." Electronics 11.9 (2022): 1294.

[22] Alwakid, Ghadah, et al. "Melanoma detection using deep learning-based classifications." Healthcare. Vol. 10. No. 12. MDPI, 2022.

[23] Gouda, Walaa, et al. "Detection of skin cancer based on skin lesion images using deep learning." Healthcare. Vol. 10. No. 7. MDPI, 2022.