Ain Shams University
Faculty of Computer & Information Sciences
Computer Science Department

Skin Lesions Detection and Classification

July 2024

Ain Shams University
Faculty of Computer & Information Sciences
Computer Science Department

Skin Lesions Detection and Classification

By:

Marleen Tawadrous Yaacoub [CS]
Martina Sabry Mosaad [CS]
Mohammed Tamer Mohammed [CS]
Mina Adel Louiz [CS]
Mina Nabil Asaad[CS]

Under Supervision of:

Dr. Islam Hegazy
Lecturer,
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

TA. Zeina Rayan
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

Acknowledgement

First of all, we would like to thank God for giving us the strength to achieve this project.

We owe our deepest gratitude to our supervisors Dr. Islam Hegazy and TA. Zeina Rayan for all the efforts they made guiding us through the whole project duration, their encouragement and continuous optimism concerning this work.

Moreover, their leadership and how they managed to inspire and guide us through the past year, their experience and enthusiasm that they passed on to every one of the team.

We are deeply grateful for our college support that came from all the staff there, all the professors who lectured us and teaching assistants who gave us lots of guidance.

We would like to thank our department represented in its head Dr. Abeer Mahmoud and the whole facility represented in the dean Prof. Dr. Nagwa Badr for all of the support we got from them.

Finally, it is a pleasure to thank those who made this project possible who are represented in our colleagues and most important our families for their support through the whole project duration.

Abstract

Skin lesions, such as moles and melanomas, present significant challenges in diagnosis and early detection. This project focuses on developing an automated system using deep learning techniques to classify skin lesions from dermoscopic images. Two main datasets, HAM10000 and ISIC2018, are utilized for training and evaluating machine learning models. The system leverages TensorFlow for model development and Flask for creating an API to facilitate seamless integration with a Flutter-based mobile application.

Key components include image preprocessing techniques like resizing and data augmentation to enhance model accuracy. The InceptionResNet model, trained on resized images, achieves a notable 96.13% accuracy in lesion classification. The project also emphasizes the use of Colab for GPU-accelerated computations and Git for version control.

Future enhancements aim to further refine the model's accuracy, integrate real-time lesion detection capabilities, and ensure compliance with medical standards. By providing a user-friendly interface and reliable diagnostic capabilities, this system aims to assist healthcare professionals in timely and accurate skin lesion diagnosis, ultimately improving patient care outcomes.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

API	Application Programming Interface
APP	Application
CNN	Convolutional Neural Network
HAM	Human Against Machine
ISIC	International Skin Imaging Collaboration
SVM	Support Vector Machine
DT	Decision Trees
RF	Random Forest
ResNet	Residual Network
Adam	Adaptive Moment Estimation
GUI	Graphical User Interface

1- Introduction

1.1 Motivation

Skin diseases are becoming more common around the world, affecting millions of people. Many individuals, especially those in rural and underserved areas, struggle to get timely access to dermatologists. This can lead to delays in getting a diagnosis, causing stress and worsening their condition.

Additionally, even when people do see a healthcare provider, the accuracy of diagnoses can vary. This inconsistency underscores the need for a reliable and easy-to-use tool that can help identify different types of skin lesions. By using deep learning and advanced image analysis, our project aims to create a system that can classify skin images into seven specific categories. This tool will support healthcare professionals by providing consistent results and offer a valuable resource for patients, making dermatological care more accessible and efficient for everyone.

1.2 Problem Definition

Skin lesions have many levels of severity, ranging from benign conditions to serious diseases like skin cancer. Some factors, such as exposure to UV rays, can increase the severity of these conditions. Moreover, approximately 25% of malignant skin lesions can develop from moles, adding to the diagnostic challenges.

Given these challenges, there is a critical need for an automated system that can quickly and accurately classify different types of skin lesions. This application encourages people to check their skin lesions promptly, helping to detect potential issues before the disease becomes more serious. Such a system will assist healthcare providers by offering consistent and reliable assessments, improving diagnostic efficiency, and ensuring timely and appropriate care for patients.

1.3 Objective

The primary objective of this project is to develop a machine learning-based system capable of classifying skin images into seven distinct categories. These categories represent various types of skin lesions, including both benign and malignant conditions. By utilizing advanced image processing techniques and state-of-the-art algorithms, our system aims to achieve high accuracy and reliability in lesion classification. Importantly, our system is designed to serve as an aid in the detection process without providing any medical advice, ensuring that patients are directed towards appropriate medical consultation for further evaluation and treatment.

1.4 Time Plan

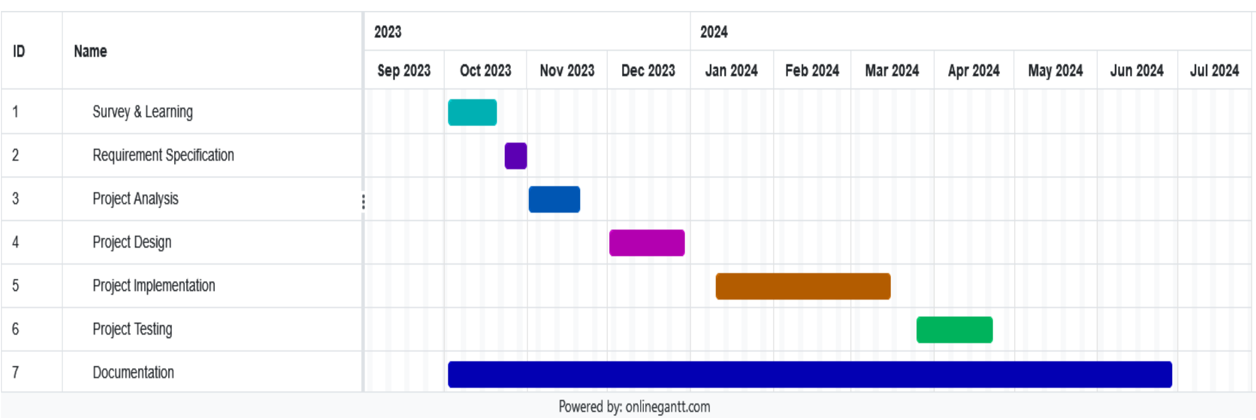

Figure 1 Time Plan

1.5 Document Organization

The thesis includes six chapters, that’s the first. The chapters’ description is presented briefly as follows :

Chapter 2 : Background

In this chapter, we will define all the information about our application.

Chapter 3 : System Analysis & Design:

In this chapter, we will present the architecture of our system, including the main components and their relationships, as well as detailed diagrams such as the system architecture, use case diagram, class diagram, and sequence diagram.

Chapter 4 : System Implementation & Testing :

In this chapter, we will describe the implementation and results of the system in details.

Chapter 5 : User Manual :

In this chapter, we will show our mobile application workflow and how it works.

Chapter 6 : Conclusion and Future Work :

In this chapter, we will summarize the project.

2- Background

the Field of the Project

The skin is the broadest organ in the body which protects the body against the heat, light, and infection. It also helps to control the body temperature and to store the fat and the water. One of the most important problems of skin in the body is its infection risk to skin cancer. Melanoma is the most malignant and most serious type of skin cancer and is the reason for most deaths from skin cancer. The underlying cause of melanoma is unknown . But several factors, including genetic factors, ultraviolet radiation, and environmental contact are involved in causing the disease.[3]
Traditional diagnostic methods rely heavily on the expertise of dermatologists, making the process time-consuming and subjective. The advent of artificial intelligence (AI) and deep learning techniques has introduced a new paradigm in dermatology, offering automated and efficient approaches to classify skin lesions, thereby potentially improving diagnostic accuracy and accessibility.

Scientific Background Related to the Project

Machine Learning and Deep Learning

Artificial intelligence is a field of science concerned with building computers and machines that can reason, learn, and act in such a way that would normally require human intelligence or that involves data whose scale exceeds what humans can analyze. AI is a broad field that encompasses many different disciplines, including computer science, data analytics and statistics, hardware and software engineering, linguistics, neuroscience, and even philosophy and psychology. On an operational level for business use, AI is a set of technologies that are based primarily on machine learning and deep learning, used for data analytics, predictions and forecasting, object categorization, natural language processing, recommendations, intelligent data retrieval, and more.[33]

Machine learning, a subset of AI, involves training algorithms to recognize patterns in data and make predictions based on those patterns.[34]

Deep learning is set of Algorithms or a machine learning techniques that uses multiple layers to progressively to learn the useful representation or higher level features of the raw input data (images, text, sound, etc.) to represent it, there are many different architectures of Deep learning used in different fields such as Deep neural Networks, Deep belief networks, Convolution Networks have been applied in different fields such as computer vision, machine vision, Speech recognition, NLP and other different applications where they have produced results comparable to and in some cases surpassing human expert performance.[35]

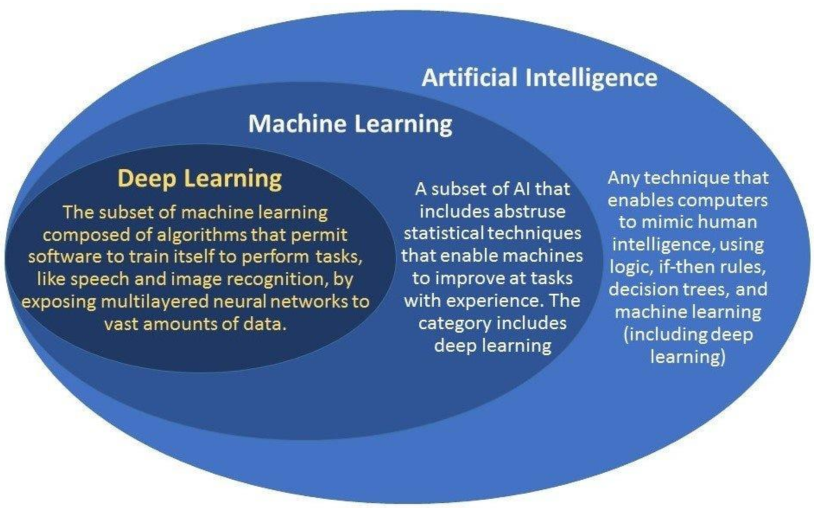

Figure 2 AI Fields

Support Vector Machine

Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification, regression, and even outlier detection tasks. SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection. SVMs are adaptable and efficient in a variety of applications because they can manage high-dimensional data and nonlinear relationships.

SVM algorithms are very effective as we try to find the maximum separating hyperplane between the different classes available in the target feature.

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression.

Though we say regression problems as well it's best suited for classification. The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three. [40]

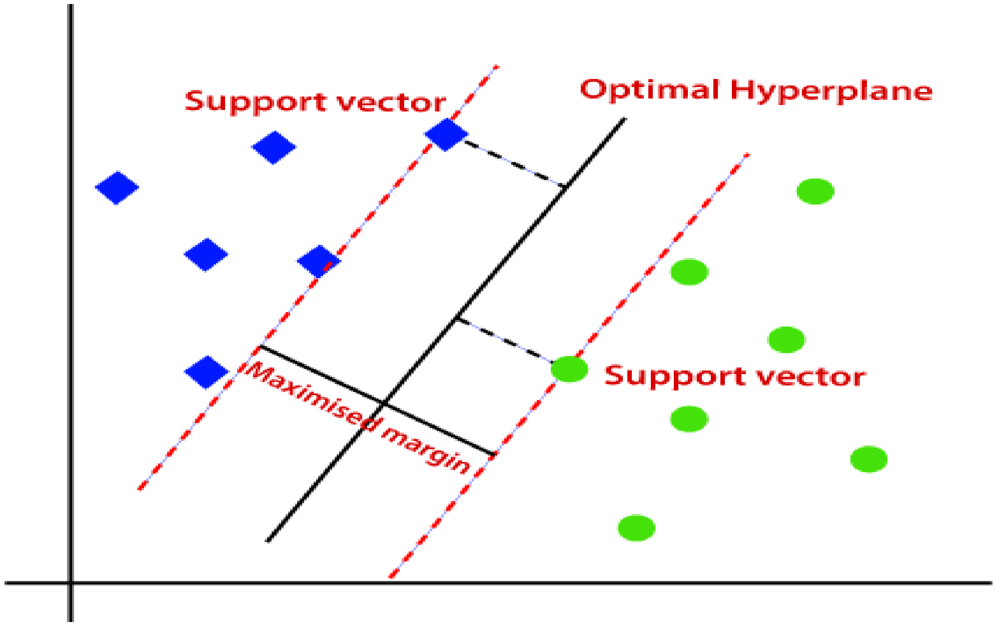
Support vector
Optimal Hyperplane
Support vector
Maximised margin

Figure 3 SVM

Decision Trees (DT)
A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.
As you can see from the diagram below, a decision tree starts with a root node, which does not have any incoming branches. The outgoing branches from the root node then feed into the internal nodes, also known as decision nodes. Based on the available features, both node types conduct evaluations to form homogenous subsets, which are denoted by leaf nodes, or terminal nodes. The leaf nodes represent all the possible outcomes within the dataset.[39]

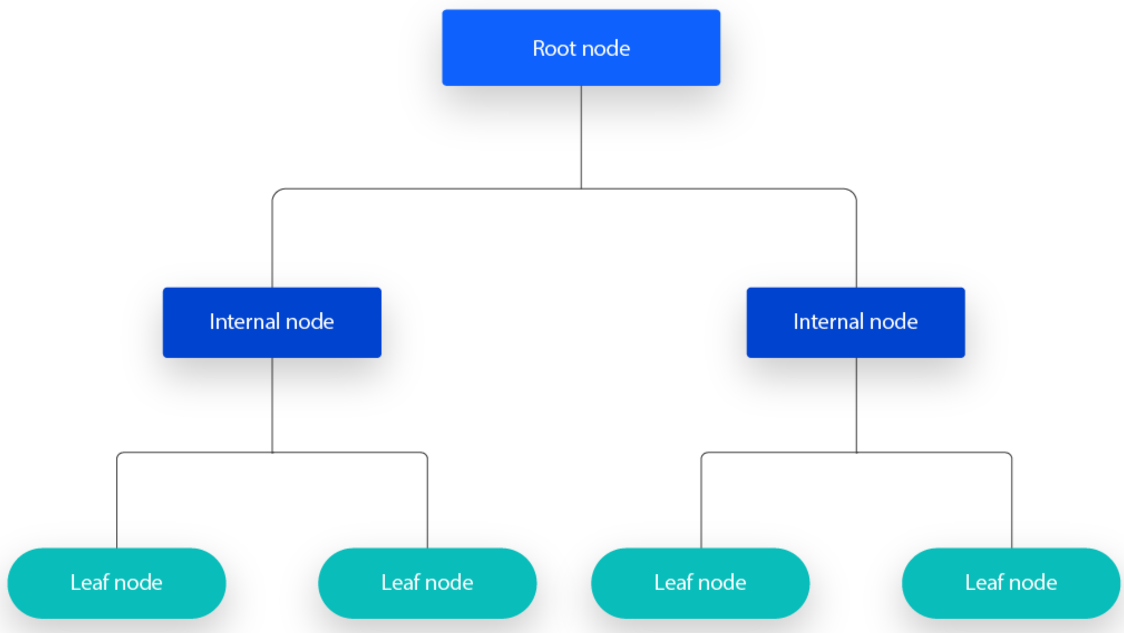

Figure 4 Decision Trees

Random Forest (RF)

Random Forest algorithm is a powerful tree learning technique in Machine Learning. It works by creating a number of Decision Trees during the training phase. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance. In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks) This collaborative decision-making process, supported by multiple trees with their insights, provides an example stable and precise results. Random forests are widely used for classification and regression functions, which are known for their ability to handle complex data, reduce overfitting, and provide reliable forecasts in different environments.[38].

Figure 5 Random Forest

Convolutional Neural Networks (CNNs)

CNNs are a type of deep learning model specifically designed for processing grid-like data, such as images. They consist of several types of layers:

1. **Convolutional Layers**: These layers apply convolution operations to the input, using filters to extract features such as edges, textures, and patterns.
2. **Pooling Layers**: These layers reduce the spatial dimensions of the data, thereby decreasing the computational load and helping to generalize the model.
3. **Fully Connected Layers**: These layers combine the features extracted by previous layers to make predictions.

CNNs have been highly successful in image classification tasks due to their ability to automatically learn and extract relevant features from raw image data. In the context of skin lesion detection, CNNs are trained on labeled dermoscopic images to identify patterns and characteristics indicative of various skin conditions [1].

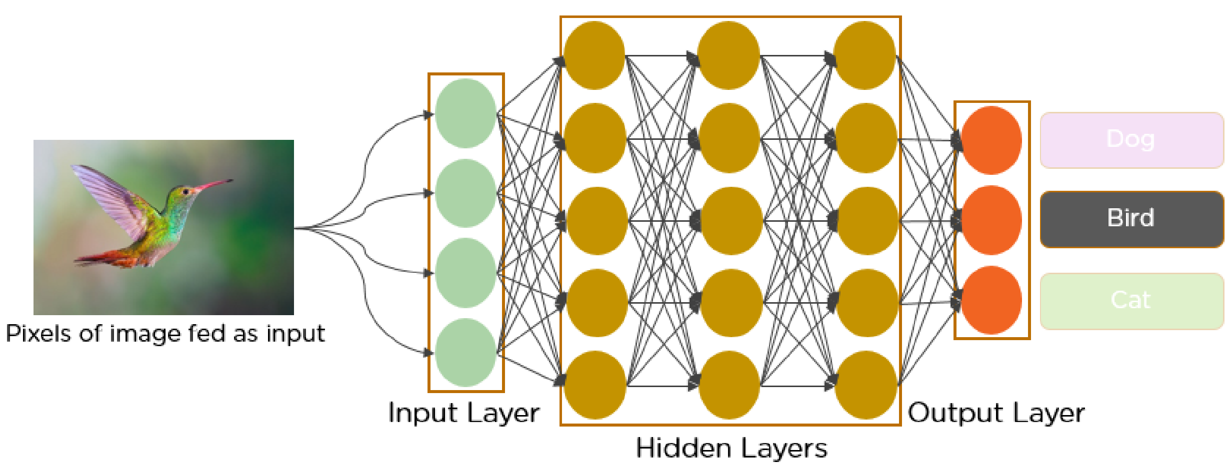

Figure 6 CNN

Inception Net

The Inception Network is an advanced architecture for deep learning models, designed to address detection and classification problems efficiently. It was developed as a result of the ILSVRC Competition, known for pioneering innovative deep learning models.[37].

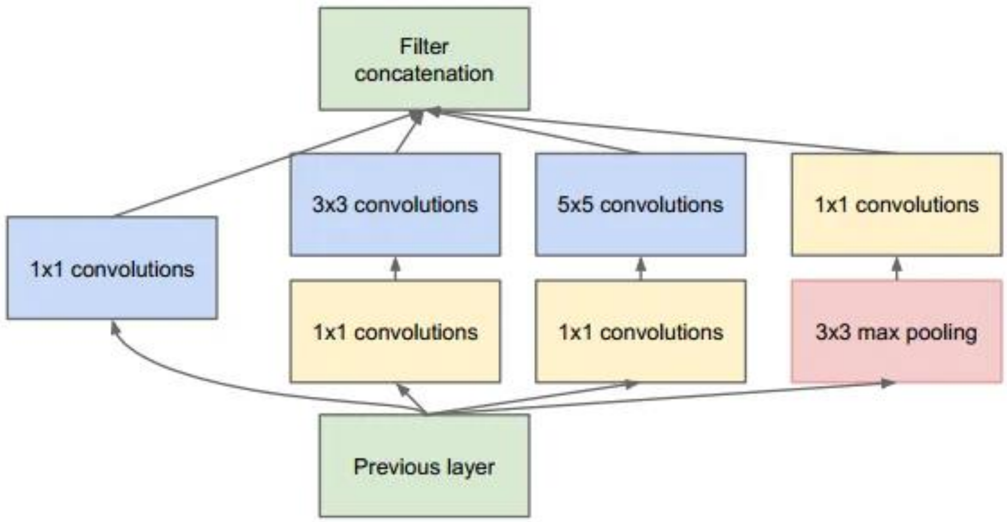

Figure 7 Inception Block

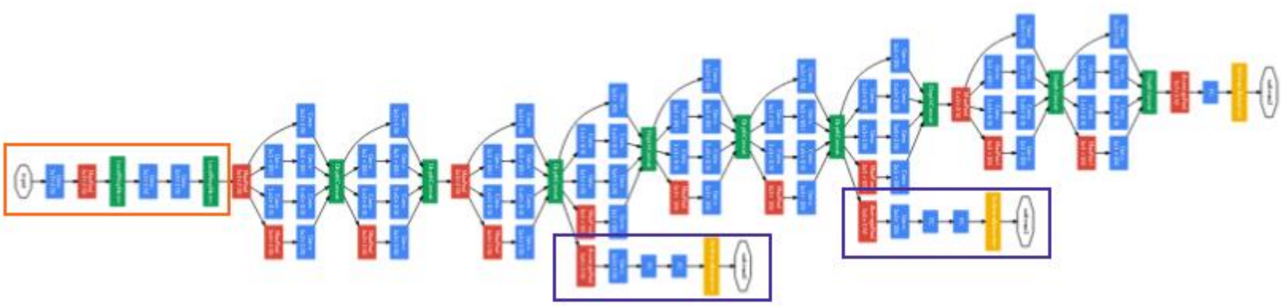

Figure 8 Inception Net

InceptionRestNet

Inception-ResNet, introduced by Google in 2016, stands as a pivotal development in convolutional neural network (CNN) architectures, combining elements from the Inception and ResNet models to address key challenges in deep learning. The Inception architecture, known for its efficient use of multi-level feature extraction through parallel convolutional pathways, was enhanced by integrating residual connections inspired by ResNet. These connections facilitate the training of very deep networks by alleviating the degradation problem caused by increased depth, where adding more layers can lead to diminishing performance due to vanishing gradients. Inception-ResNet achieves this by allowing gradients to flow more directly through the network during backpropagation, thereby enabling more effective training of deeper models. This innovation not only improves convergence rates but also enhances overall accuracy on tasks such as image classification and object detection. By combining the strengths of Inception's multi-scale feature extraction with ResNet's residual learning, Inception-ResNet has become a cornerstone in modern CNN architectures, setting new benchmarks in performance across various computer vision tasks and inspiring further advancements in deep learning research.[36]

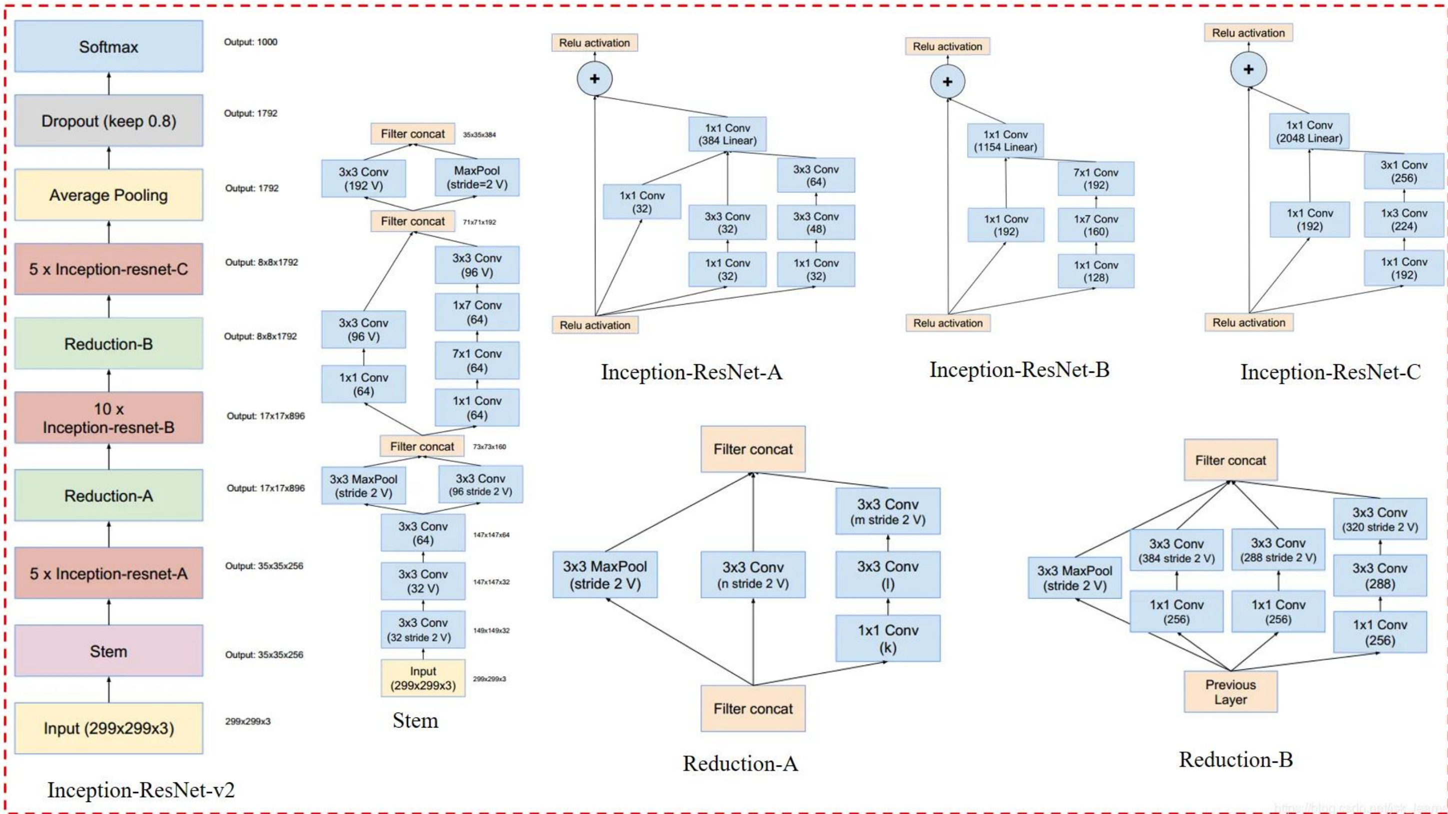

Figure 9 Inception ResNet

Survey of Work Done in the Field

Numerous studies have explored the application of deep learning for skin lesion classification:

1. Dildar et al. reviewed various deep learning techniques applied to skin cancer detection, highlighting the potential of CNNs to achieve high accuracy rates [2].
2. Zhang et al. developed an optimized CNN model that significantly improved the diagnostic accuracy of skin cancer [3].
3. Daghrir et al. proposed a hybrid approach combining deep learning and classical machine learning techniques for melanoma detection, achieving promising results [4].
4. Jenitha et al. explored the use of machine learning algorithms for skin cancer recognition, emphasizing the importance of feature selection and model training [5].
5. Bhatt et al. provided a comprehensive review of state-of-the-art machine learning techniques for melanoma detection, highlighting the advancements and challenges in the field [6].

Other notable contributions include:

- Agarwal and Singh's work on using CNNs for skin cancer image classification [7].
- Naqvi et al.'s review on deep learning applications in skin cancer detection [8].
- Nawaz et al. utilized deep learning and fuzzy k-means clustering for dermoscopic image analysis, further demonstrating the potential of AI in dermatology [9].
- Gouda et al. developed a deep learning-based system for detecting skin cancer from skin lesion images, showcasing the effectiveness of these techniques in clinical settings [10].

Paper	Methodologies	Datasets	Evaluation
Skin lesion classification of dermoscopic images using machine learning and convolutional neural network “2022”	Decision Tree, Random Forest , SVM, KNN, CNN	HAM10000	DT : 68%, RF : 87%
Skin Lesion Classification on Imbalanced Data Using Deep Learning with Soft Attention”2022"	DenseNet201, InceptionResNetV2, ResNet50, VGG16, Upsizing: data augmentation	HAM10000	86% combination of MobileNetV3,Soft-Attention 92% InceptionResNetV2 with Soft-Attention
Detection of Skin Cancer Based on Skin Lesion Images Using Deep Learning “Health Care 2022”	CNN, Resnet50, InceptionV3, Inception Resnet, ESRGAN, Validation patience, Upsizing: data augmentation	HAM10000 ISIC2018	the Inception model had an overall accuracy rate of 85.7%
Multi-class multi-level classification algorithm for skin lesions classification using machine learning techniques”2020”	RCNN along with FKM, Polesel, Ramponi, Mathews	ISIC-2016, ISIC-2017, and PH2	0.945, 0.963, and 0.971 respectively with datasets
State-of-the-art machine learning techniques for melanoma skin cancer detection	DCNN, AlexNet, VGG16, FCNs based on VGG16 and GoogleNet, Deep RCNN and Fuzzy C-Mean Clustering, DenseNet with	PH2 , ISIC 2017 , ISBI 2016, ISBI 2017 , ISBI 2017, ISIC Archive	Optimized CNN : 93% DCNN : 95% LDA with CNN : 85%

and classification: a comprehensive review “2023”	IcNR, Optimized CNN, Upsizing: data augmentation		
Skin Cancer Detection Using Deep Learning— A Review “2023”	AlexNet, VGG, ResNet, DenseNet, MobileNet, Upsizing: data augmentation	HAM10000, PH2, ISIC”2016 to 2020”, Atlas of Dermoscopy, Dermofit, BCN20000, PAD-UFES-20	VGG-16 achieving the best accuracy of 88% Xception achieving the highest accuracy of 90.48%
The Development of a Skin Cancer Classification System for Pigmented Skin Lesions Using Deep Learning ”2020”.	VGG-16,stochastic gradient descent. data augmentation	approximately 120,000 clinical images taken from 2001 to 2017 at the Department Dermatologic Oncology in the National Cancer Center Hospital, they extracted 5846 clinical images of brown to black pigmented skin lesions from 3551 patients. “Not Accessible”	86.2% for the FRCNN.
Skin Lesions Classification Based on Deep Learning Approach “2020”.	VGG16, Inception_V3, ResNet-50.	ISIC Archive Dataset	85.3%,84.3% and 81.7 respectively with methodologies.
Classification of Skin Cancer Images using Convolutional Neural	DenseNet,Resnet, XceptionNet, MobileNet, Augmentation	Skin Cancer: Malignant vs. Benign	DenseNet : 86% Resnet : 86% XceptionNet: 82% MobileNet : 80%

Networks"2022"			
Soft-Attention Improves Skin Cancer Classification Performance"2021"	Deep CNN, Deep CNN with soft-Attention Up sizing: Data augmentation	ISIIC 2017 HAM 10000	HAM10000 + CNN + Soft : 93.7% ISIC2017 + CNN + Soft : 91.6%

Table 1 Abstract of Related Work

Description of Existing Similar Systems

Several existing systems leverage AI and deep learning for skin lesion detection and classification. For instance, the International Skin Imaging Collaboration (ISIC) has hosted challenges to promote the development of automated systems for skin lesion analysis, including tasks such as lesion segmentation, attribute detection, and disease classification. These challenges have provided valuable datasets and benchmarks for researchers to develop and evaluate their models [21].

Commercial applications and research prototypes have been developed to assist dermatologists and general practitioners in diagnosing skin lesions. These systems typically involve a mobile application that allows users to capture images of their skin lesions, which are then analyzed by a deep learning model to provide a preliminary diagnosis.

3- Analysis and Design

3.1 System Overview

3.1.1 System Architecture

visualizes the system architecture of the project. The system architecture is composed of three layers. The three-tier architecture is divided into Application, Processing and Data layers. Section 3.2 discusses each layer in detail.

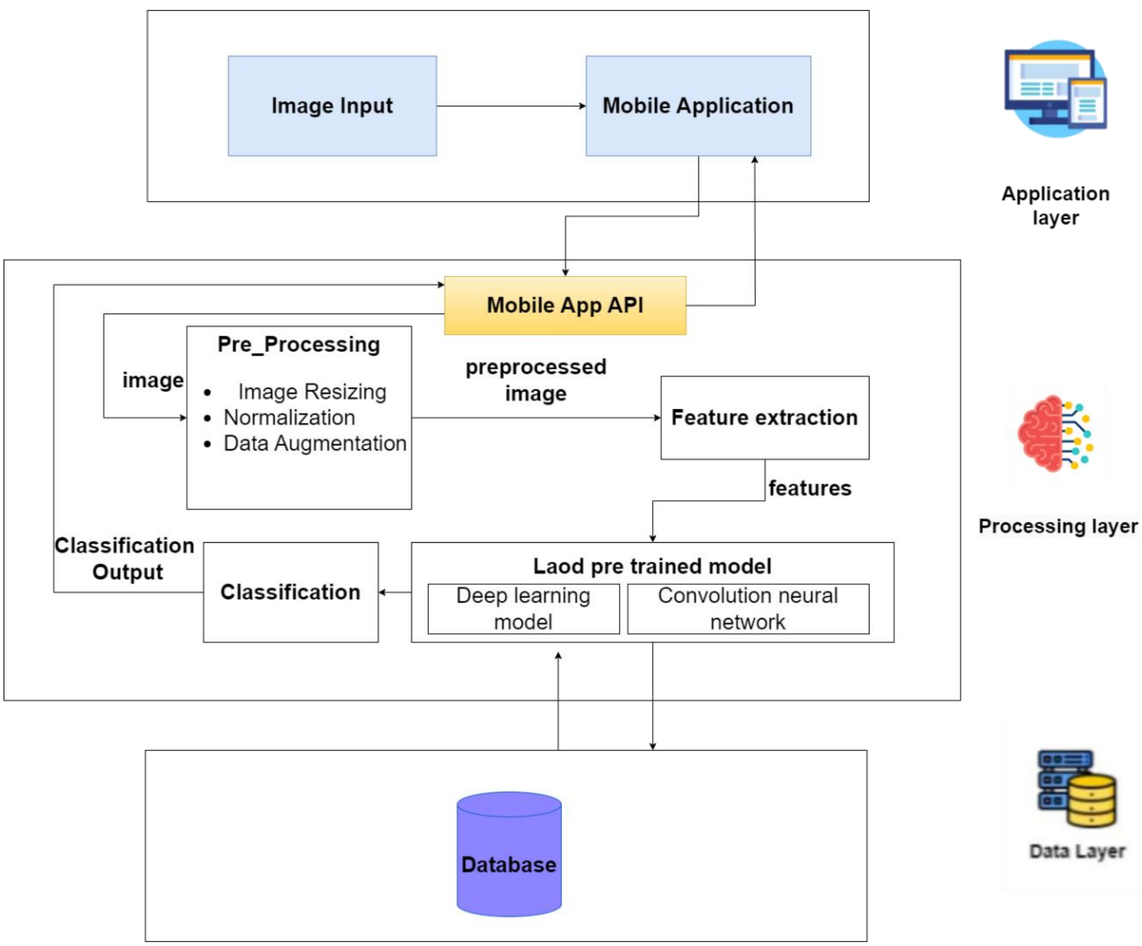

Figure 10 System Architecture

3.2 Description of Methods and Procedures

1. Application Layer

User Interface and communication layer of the application where the end user interacts with the application.

Mobile Application

- User Interaction: The mobile application provides the interface through which users interact with the system. Users can capture or upload images, submit them for classification, and view the results.
- Image Capture: Users can take a photo using the device's camera or select an existing image from the gallery.
- Sending Data: Once an image is captured or selected, it is sent to the backend (Mobile App API) for processing.
- Displaying Results: After the backend processes the image and returns the classification result, the mobile app displays the result to the user.

2. Processing Layer

The "**Mobile App API**" module which is in the application layer**-** is responsible for the data flow between the three layers and most of their modules.

Preprocessing

- Image Resizing: Adjusts the dimensions of the image to fit the requirements of the deep learning model
- Normalization: Adjusts the pixel values to a standard scale (e.g., values between 0 and 1 or -1 and 1) to improve the model's performance.
- Data Augmentation: Applies transformations like rotation, flipping, and cropping to increase the variability of the training data and improve the model's robustness.

Feature Extraction

- **This module** extracts relevant features from the preprocessed image.
- This step reduces the dimensionality of the data and retains essential information needed for classification.

Load Pre Trained Model

Convolutional Neural Networks: The CNNs are extensively utilized in the classification of skin lesion images due to their exceptional performance and remarkable outcomes. CNNs, being a type of multilayer perceptron, form a fully connected network where each neuron is connected to all neurons in the subsequent layer.

These networks employ convolutional operations to extract significant features from the input data.

They consist of various layers, such as convolutional layers, pooling layers, and fully connected layers.

In the context of skin lesion image classification, the neurons in CNNs possess three dimensions → width, height, and depth.

Additionally, they establish selective connections with a small region of the previous layer.

Consequently, CNNs have gained immense popularity as highly effective Deep Neutral Network arch have rained in related to skin lesion image classification.

Classification

- The model processes the features and produces a classification output.
- The classification result is sent back to the Mobile App API to be returned to the mobile application.

3. Data Layer

- includes a **database** that is primarily used to store and retrieve the pretrained model and its weights. This ensures that the model and its associated weights are securely stored and can be accessed efficiently for processing images.

3.1.2 System Users

A. Intended Users:

The skin lesions detection and classification system is designed for the general public and healthcare providers. The general public can use the system to upload images of skin lesions from their camera or gallery ,and receive immediate detection and classification results. Healthcare providers can utilize the system as a preliminary screening tool to aid in clinical evaluations.

B. User Characteristics:

There is no prior experiences or skills required for operating the system. This system is designed to be operated by any non-technical user. Users do not need any medical knowledge, as the system provides clear and understandable results. Basic reading ability in the supported language is sufficient.

3.2 System Analysis & Design

3.2.1 Use Case Diagram

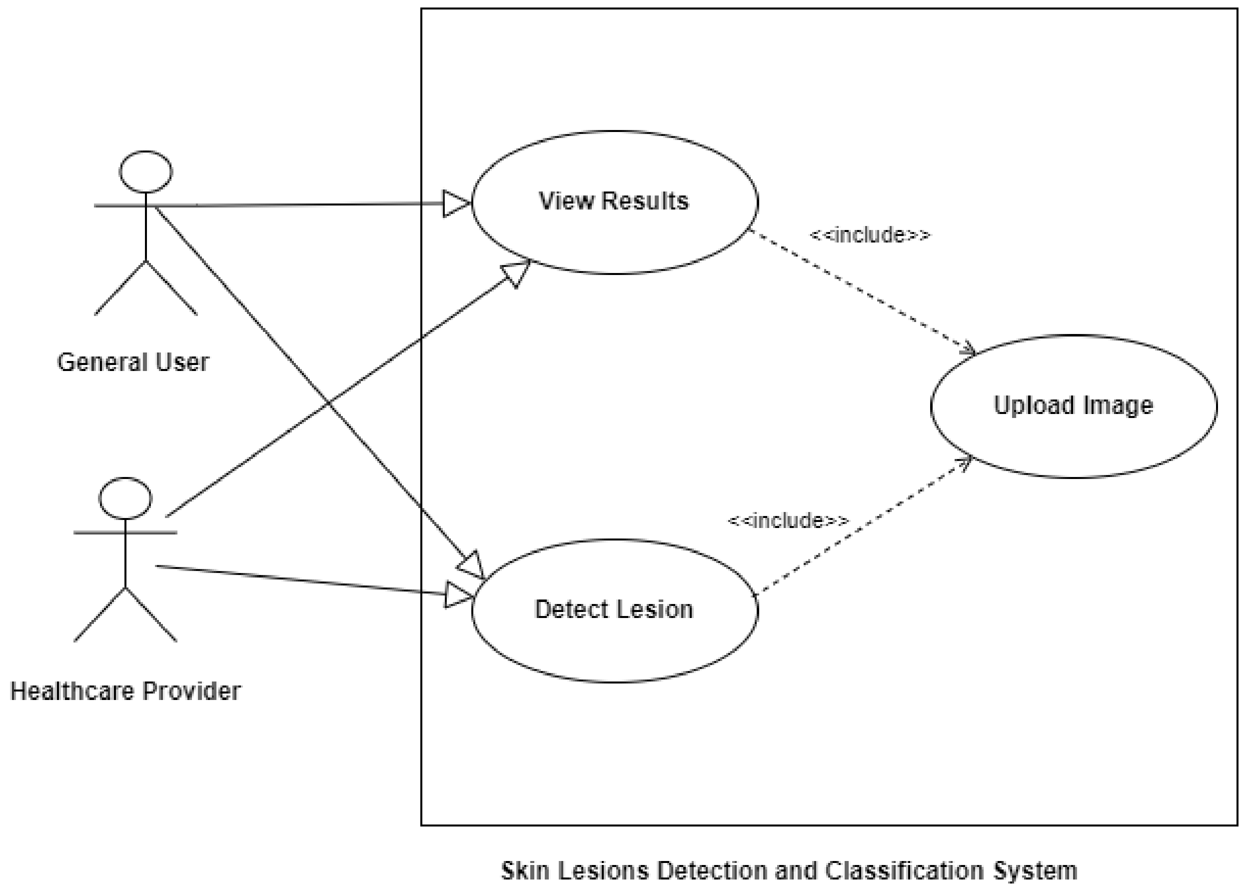

Figure 11 Use Case Diagram

There are two main functions in the use case diagram: upload image and view results.

Upload Image function is where the user can upload an image of a skin lesion either from their device's camera or gallery, and to do so the function include detect lesion function where the system will check the existence of the username in the accounts stored in the database.

3.2.2 Class Diagram

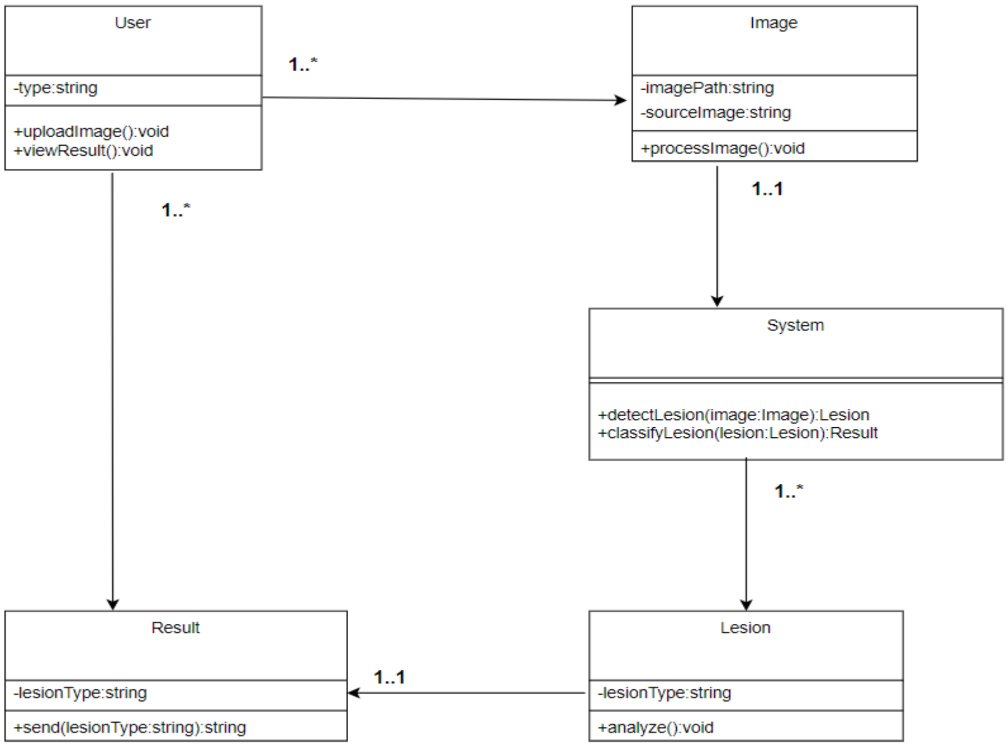

Figure 12 Class Diagram

The class diagram for the Skin Lesion Detection and Classification System illustrates the architecture and interaction between various components in the system. This system is specifically designed to allow users to upload images of skin lesions, process these images to detect and classify the lesions, and subsequently present the results to the users. Below is a detailed description of each class and the relationships between them, providing a comprehensive understanding of the system's functionality and structure.

The **User** class represents the individuals interacting with the system. It contains a single attribute ‘ type’ which specifies the category of the user either a general user or a healthcare provide. The User class has two methods: `uploadImage()` and `viewResult()`. The `uploadImage()` method allows users to upload images of skin lesions, while the `viewResult()` method enables them to view the results of the lesion analysis. This design ensures that users can easily interact with the system without requiring any technical expertise.

The **Image** class encapsulates the details of the images uploaded by the users. It has two attributes: `imagePath`, which stores the file path of the image, and `sourceImage`, which indicates the source of the image, such as whether it was taken from a camera or selected from the gallery. The `processImage()` method in this class handles the preprocessing of the image, preparing it for further analysis by the system.

The **System** class is central to the functionality of the skin lesion detection and classification process. It contains two key methods: `detectLesion()` and `classifyLesion()`. The `detectLesion(image: Image): Lesion` method is responsible for identifying potential lesions in the given image, while the `classifyLesion(lesion: Lesion): Result` method classifies the detected lesion into specific types and generates a corresponding result. This class orchestrates the core operations, ensuring that images are processed accurately and efficiently.

The **Lesion** class represents the identified lesions in the images. It has a single attribute, `lesionType`, which specifies the type of lesion detected. The `analyze()` method is used to analyze the lesion, providing necessary data that aids in the classification process. This class plays a crucial role in detailing the characteristics of the lesions detected in the images.

The **Result** class encapsulates the outcome of the lesion classification process. It has one attribute, `lesionType`, which indicates the type of lesion as determined by the system. The `send(lesionType: string): string` method sends the classification result back to the user. This class ensures that the results are communicated effectively to the users, completing the cycle of interaction.

The relationships and multiplicities between these classes define how they interact with each other. A **User** can upload multiple **Images** (1..*), but each **Image** is associated with only one **User** (1). Each **Image** is processed by one **System** instance (1), and the **System** can handle multiple **Images** (1..*). The **System** detects and classifies multiple **Lesions**(1..*), but each **Lesion** is associated with a single **System** instance (1). Each **Lesion** generates a single **Result**(1), and each **Result** is tied to one **Lesion** (1). Finally, a **User** can view multiple **Results**(1..*), and each **Result** can be viewed by multiple **Users** (1..*).

3.2.3 Sequence Diagram

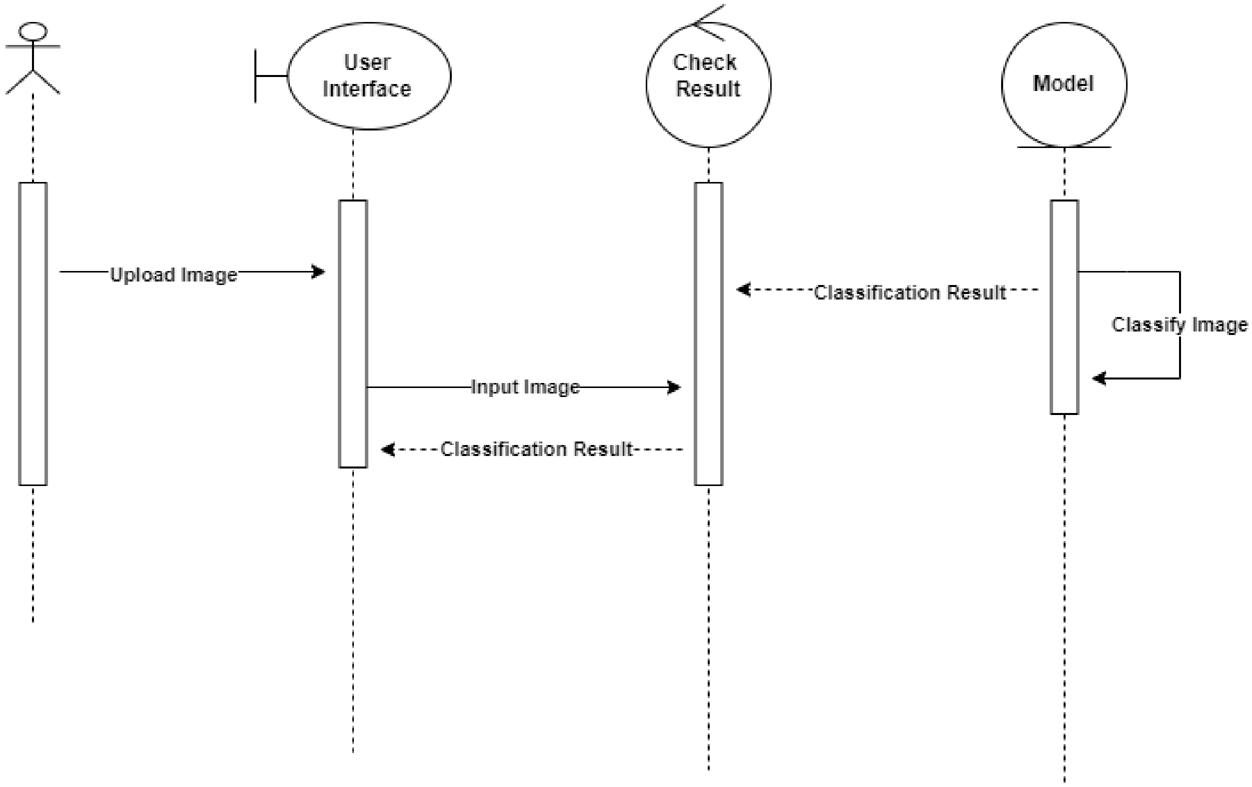

Figure 13 Sequence Diagram

Overview

This sequence diagram represents the process of image classification in a system that involves four main entities: the user,
user interface, check result component, and the model. The interactions between these entities are detailed as follows:

Entities

User: The person interacting with the system.

User Interface: The component through which the user uploads the image.

Check Result: The component responsible for handling the results of the classification.

Model: The component that performs the actual image classification.

Interaction Flow

The user uploads an image to the system via the user interface The user interface sends the uploaded image to the check result control.

The check result component forwards the image to the model for classification.

The model processes the image and returns the classification result to the check result component.

Finally, the check result component sends the classification result back to the user interface.

The user interface displays the classification result to the user. This sequence diagram illustrates the process of image classification, from the user uploading an image to receiving the classification results.

4- Implementation and Testing

4.1 Datasets

Two datasets are used to train and evaluate the different models implemented in this project. Each of the datasets is explained below, outlining the data format, number of entries, etc. The datasets are: HAM10000 , ISIC2018

HAM10000: Dermoscopic Skin Lesion Classification Dataset:-

The HAM10000 dataset is a collection of dermoscopic images designed to train machine learning models for classifying pigmented skin lesions.

Content:

- **Images:** 10015 dermoscopic images of pigmented skin lesions acquired from diverse populations and sources.
- **Labels:** Each image is assigned one of seven diagnostic categories:
    - Actinic keratoses and intraepithelial carcinoma (akiec)
    - Basal cell carcinoma (bcc)
    - Benign keratosis-like lesions (bkl)
    - Dermatofibroma (df)
    - Melanoma (mel)
    - Melanocytic nevi (nv)
    - Vascular lesions (vasc)
- **Ground Truth:** Over 50% of lesions have histopathological confirmation. The remaining diagnoses are confirmed through follow-up examination, expert consensus, or in-vivo confocal microscopy.
- **Multi-Image Lesions:** The dataset includes lesions with multiple images linked by a unique identifier (`lesion_id`) within the metadata file.

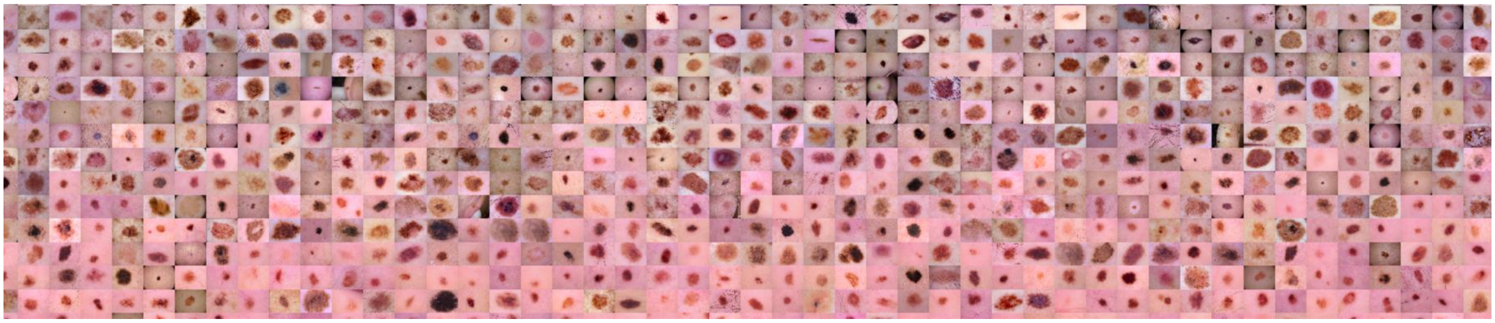

Figure 14 HAM10000 Dataset

ISIC2018 :-

The HAM10000 dataset provides 10015 labeled dermoscopic images of pigmented skin lesions for training machine learning models in classification. This dataset, along with the ISIC 2018 challenge that builds upon it, serves as a valuable resource for researchers developing and evaluating automated systems for skin lesion classification, particularly melanoma detection. The ISIC 2018 challenge offered tasks in lesion segmentation, attribute detection, and disease classification, providing ground truth data for evaluation and fostering exploration of human-computer collaboration in this field.

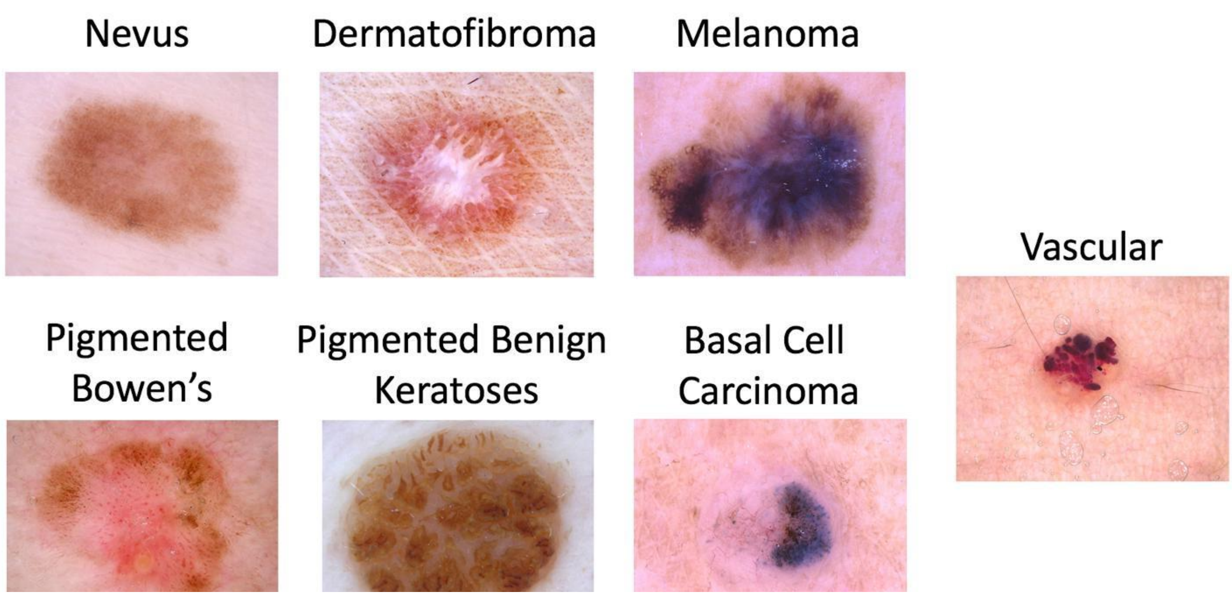

Figure 15 ISIC 2018 Dataset

4.2 Software Tools Used

Flutter: Flutter is an open-source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase which used in building the user interface.

Python: Python is a programming language that lets you work quickly and
integrate systems more effectively. In the project, we leverage certain frameworks and applications of Python, such as:

• TensorFlow: Used in building Classification model.
• Flask: Used in building the API for the application.

Colab: is a hosted Jupyter Notebook service that requires no setup to use and provides free access to computing resources, including GPUs and TPUs. Colab is especially well suited to machine learning, data science, and education.

4.3 Setup Configuration(Software)

Git: Git is a distributed version control system used for tracking changes in source code during software development. It allows multiple developers to collaborate on a project, manage different versions of the code, and facilitate code merging.

Flask: Flask is a popular web framework in Python used for building web
applications. It provides a simple and flexible way to handle HTTP requests, define routes, and render templates.

Pandas: Pandas is a powerful library for data manipulation and analysis in Python. It provides data structures and functions for efficiently handling and analyzing structured data, particularly tabular data.

Numpy: Numpy is a fundamental package for scientific comp uting in Python. It provides support for large, multi-dimensional arrays and matrices, along with a wide range of mathematical functions to operate on these arrays efficiently.

Tensorflow-gpu: Tensorflow-gpu is a version of the TensorFlow library optimized for running on Graphical Processing Units (GPUs). TensorFlow is an open-source machine learning framework that enables the creation and training of various types of neural networks.

Keras: Keras is a high-level neural networks API that runs on top of TensorFlow (or other backend engines). It provides an easy-to-use interface for building and training neural networks.

4.4 Image Preprocessing

4.4.1 Image resizing

All the images are resized to 299*299 before processing into different models.

4.4.2 Data augmentation

Data augmentation is a technique for generating new “data”. To train the machine learning models, the proposed method used Horizontal Flip augmentation i.e., shifting all pixels of an image in a horizontal direction. As a result, models with data augmentation are more likely to learn more differentiating characteristic features than models without data augmentation.

In the CNN models we have applied random transformations (rotations, shearing, etc.) to train and balance the classes. Overfitting is also avoided by data augmentation.

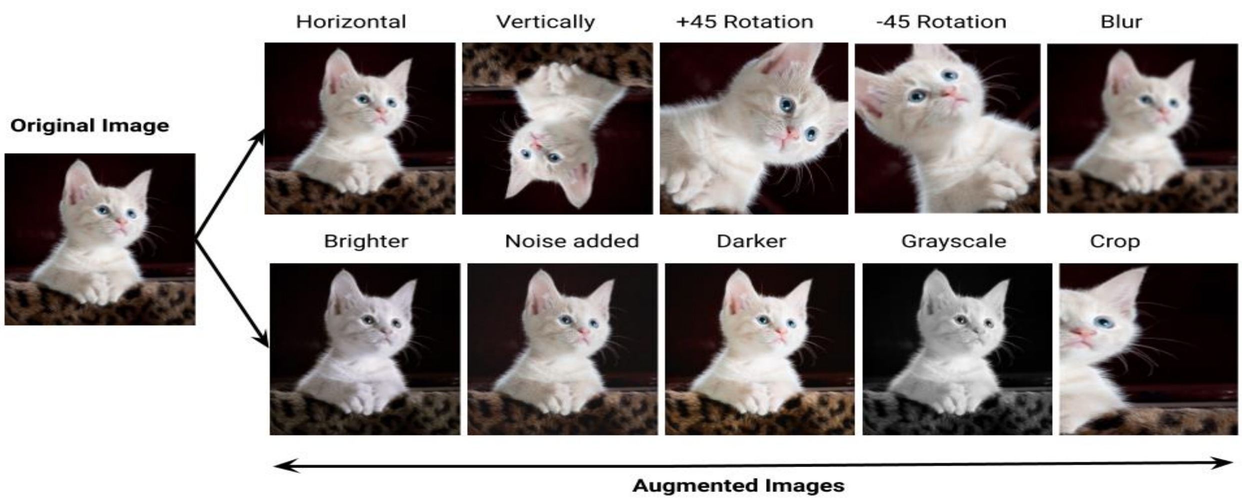

Figure 16 Data Augmentation

4.5 Models and Results

4.5.1 Support Vector Machine

Support Vector Machines (SVMs) were tried in our project for their robust performance in binary and multiclass classification tasks. SVMs are particularly effective in high-dimensional spaces and can handle nonlinear relationships between features by using kernel functions. In the context of skin lesion classification, SVMs can efficiently separate different classes of lesions by finding the optimal hyperplane that maximizes the margin between classes. This makes SVMs suitable for distinguishing between benign and malignant lesions. Moreover, their ability to work well with smaller datasets and their strong theoretical foundations in statistical learning theory make them a valuable tool in our ensemble of classification methods.

Data	**Hyperparameters values**	**Accuracy**
HAM10000	Kernel = ’linear’, c = 1	42%

Table 2 SVM Results

4.5.2 Decision Trees (DT)

Decision Trees were incorporated into our project for their simplicity and interpretability. They work by recursively splitting the data into subsets based on feature values, creating a tree-like model of decisions. This method is advantageous in skin lesion classification as it allows for easy visualization and understanding of the decision-making process. Decision Trees can capture complex interactions between features without requiring extensive data preprocessing. Their ability to handle both numerical and categorical data makes them versatile for our diverse dataset. Additionally, Decision Trees can be used as a building block for more complex ensemble methods, such as Random Forests, enhancing the overall classification performance.

Data	Hyperparameters values	Accuracy
HAM10000	Max_depth = 100	44%

Table 3 DT Results

4.5.3 Random Forest (RF)

Random Forests were tried for their superior performance in classification tasks due to their ensemble nature, which combines multiple Decision Trees to improve accuracy and generalization. By averaging the predictions of several trees, Random Forests reduce the risk of overfitting and increase robustness. In skin lesion classification, Random Forests can effectively handle large datasets with many features and provide reliable predictions by considering various feature subsets. Their ability to capture non-linear relationships and interactions between features is particularly beneficial for our project. Additionally, Random Forests provide insights into feature importance, helping to identify which aspects of the dermoscopic images are most significant for distinguishing between different types of lesions.

Data	Hyperparameters values	Accuracy
HAM10000	n_estimators = 200	68%
HAM10000 with Data Augmentation	n_estimators = 200	80%

Table 4 RF Results

4.5.4 Convolutional Neural Network

Convolutional Neural Networks (CNNs) were tried in our project due to their exceptional performance in image processing tasks. CNNs can automatically learn and extract hierarchical features from raw image data, making them ideal for complex visual recognition tasks such as skin lesion classification. Their architecture, which includes convolutional, pooling, and fully connected layers, allows them to capture spatial hierarchies and patterns within images. This makes CNNs highly effective for identifying subtle differences and patterns in dermoscopic images, which are crucial for distinguishing between various types of skin lesions. Additionally, their ability to leverage large datasets and perform transfer learning enhances their accuracy and robustness in medical image analysis.

Data	Hyperparameters values	Accuracy
HAM10000	Learning rate = 0.001 Epochs = 70	86%

Table 5 CNN Results

4.5.5 InceptionNet

In the context of skin lesion classification, the key idea behind the Inception Network is its ability to extract both global and local features from dermoscopic images simultaneously. This is achieved through inception modules, which consist of multiple convolutional blocks that can be spatially repeated. Each block outputs correlation statistics, identifying high-correlation units, which are then concatenated and fed into the next inception module. This approach enhances computational efficiency and avoids alignment issues inherent in region-based methods. Ultimately, a SoftMax activation function is applied in the final layer to provide probabilistic class assignments for different skin lesion types. Training the network typically involves over 50 epochs, using the Adam optimizer with a learning rate set to a value from 0.0001 to 0.001.

Data	Hyperparameters values	Accuracy
HAM10000	Learning rate = 0.0001 Epochs = 50 Fine tuning exclude 10 layers	85%
HAM10000	Learning rate = 0.001 Epochs = 50 Fine tuning exclude 5 layers	87%
HAM10000 with Data Augmentation	Learning rate = 0.001 Epochs = 50 Fine tuning exclude 5 layers	84%

Table 6 InceptionNet Results

4.5.6 InceptionResNet

The InceptionResNet model was chosen for this project due to its advanced architecture, which combines the strengths of Inception modules and Residual connections. This hybrid structure allows for efficient feature extraction by capturing both local and global features through multiple convolution filters of different sizes. The Residual connections help mitigate the vanishing gradient problem, enhancing training stability and effectiveness, especially in deep networks. The model's proven performance in various image classification tasks, including medical imaging, ensures its ability to accurately classify skin lesions. Additionally, its success in previous research and benchmarks, such as the ISIC challenges, supports its application in our project, aiming for high accuracy in skin lesion classification.

Data	Hyperparameters values	Accuracy
HAM10000	Learning rate = 0.001 Epochs = 70	92.3%
HAM10000 with Data Augmentation	Learning rate = 0.001 Epochs = 70 L2 regularization	84%
HAM10000 with Data Augmentation	Learning rate = 0.01 Epochs = 70 Retrain except 30 layers	94.22%
HAM10000 with Data Augmentation	Learning rate = 0.01 Epochs = 70 Retrain except 28 layers - Using Step Decay - Add 2 Dense layers	96.13%

Table 7 InceptionResNet Results

Hyperparameter	Value
Optimizer	Adam
Loss Function	Categorical Cross Entropy
Epochs	70
Batch size	32, 64
Learning rate	0.01-0.0001

Table 8 Hyperparameters

4.6 GUI

The graphical user interface (GUI) of the skin lesion classification system is built using Flutter, an open-source framework by Google for developing natively compiled applications across various platforms. The Flutter framework allows us to create a visually appealing and responsive user interface from a single codebase, ensuring consistent user experience on both Android and iOS devices.

The GUI serves as the frontend of the application, providing users with intuitive controls to interact with the system. It includes features such as:

- **Image Upload**: Users can upload images of skin lesions directly from their mobile devices or taking a picture using camera.
- **Classification Results**: The GUI displays the classification results obtained from the backend deep learning model, indicating the predicted skin lesion category (e.g., melanoma, nevi) .

By integrating a user-friendly GUI with powerful backend functionalities, the application aims to empower both healthcare professionals and individuals to efficiently diagnose and monitor skin lesions using advanced AI techniques.

Wireframes

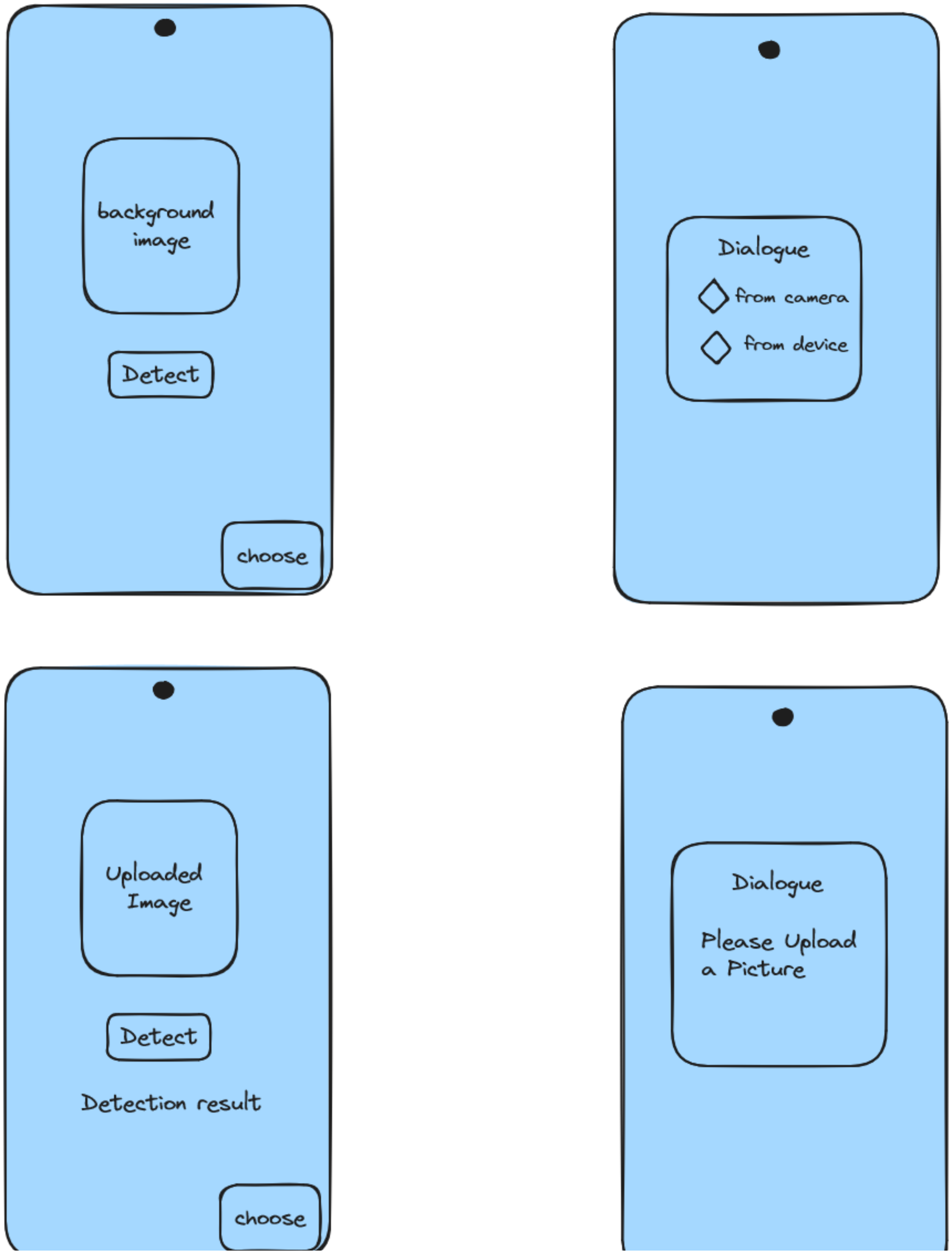

Figure 17 Wireframes

5- User Manual

5.1 Application Overview

Our production is mobile application used to Detect the uploaded image contian beging or maligant skin lesion .

5.2 Application Screens

5.2.1 Home Screen

Figure 18 Mobile App Home Screen

This screen appears when the app is launched.

5.2.2 User Options

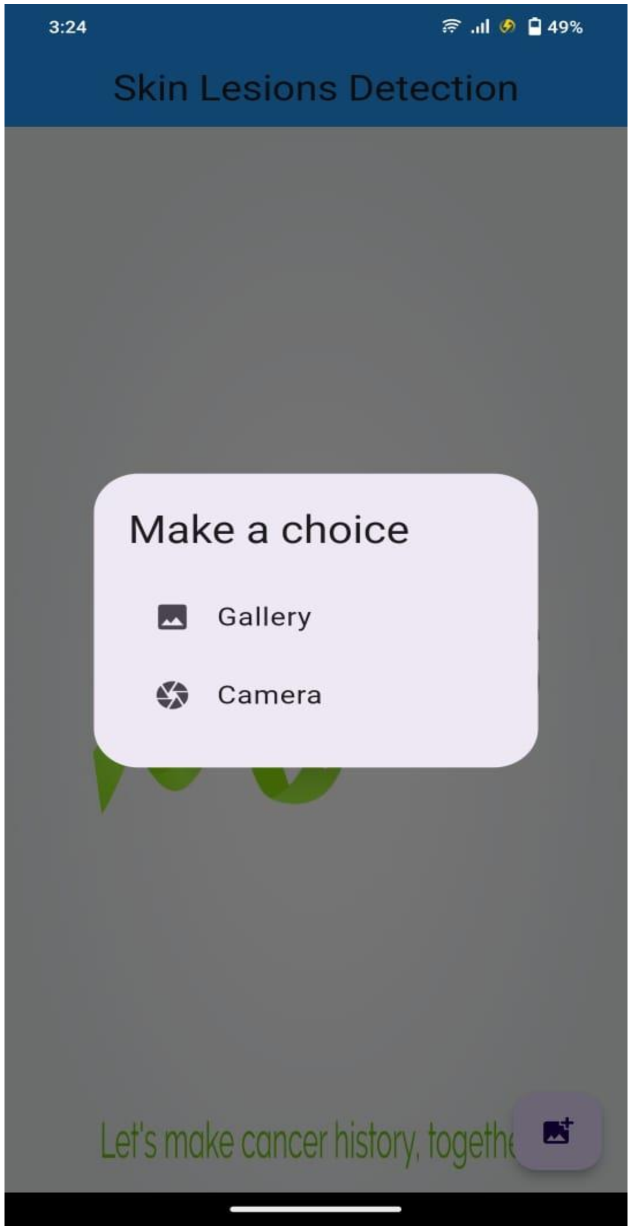

Figure 19 Mobile App Choices Screen

The user has the option to either select an image from the gallery or capture a new image using the camera**.**

5.2.3 Result Screen

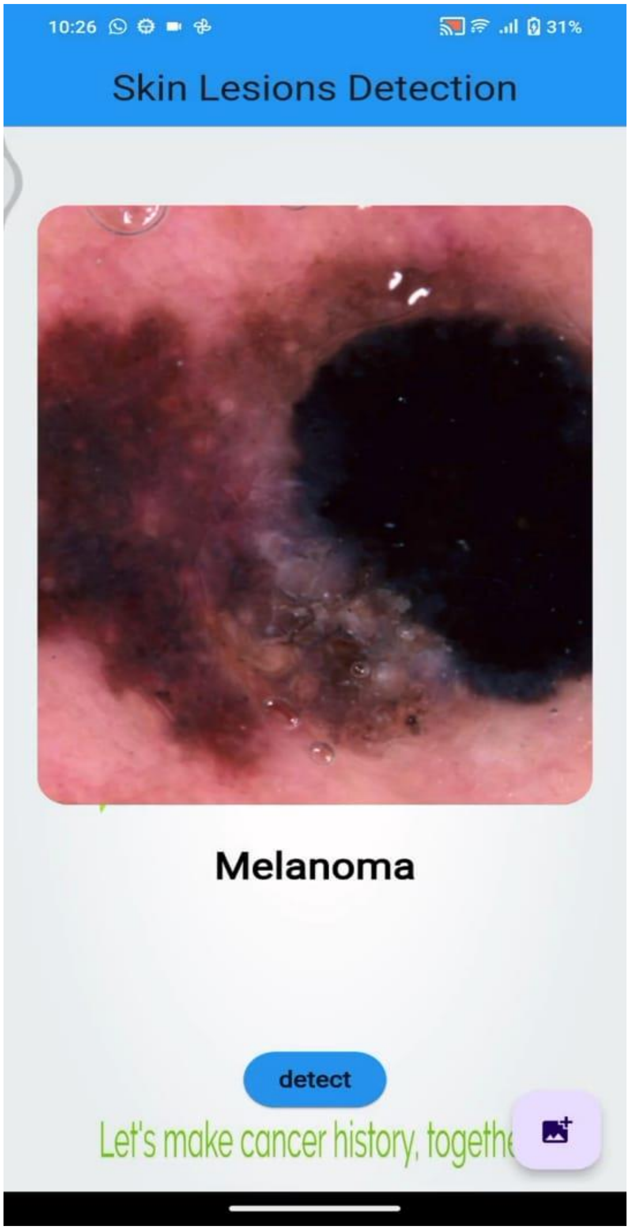

Figure 20 Mobile App Result Screen

The classification result is displayed on this screen.

5.2.4 Alert Screen

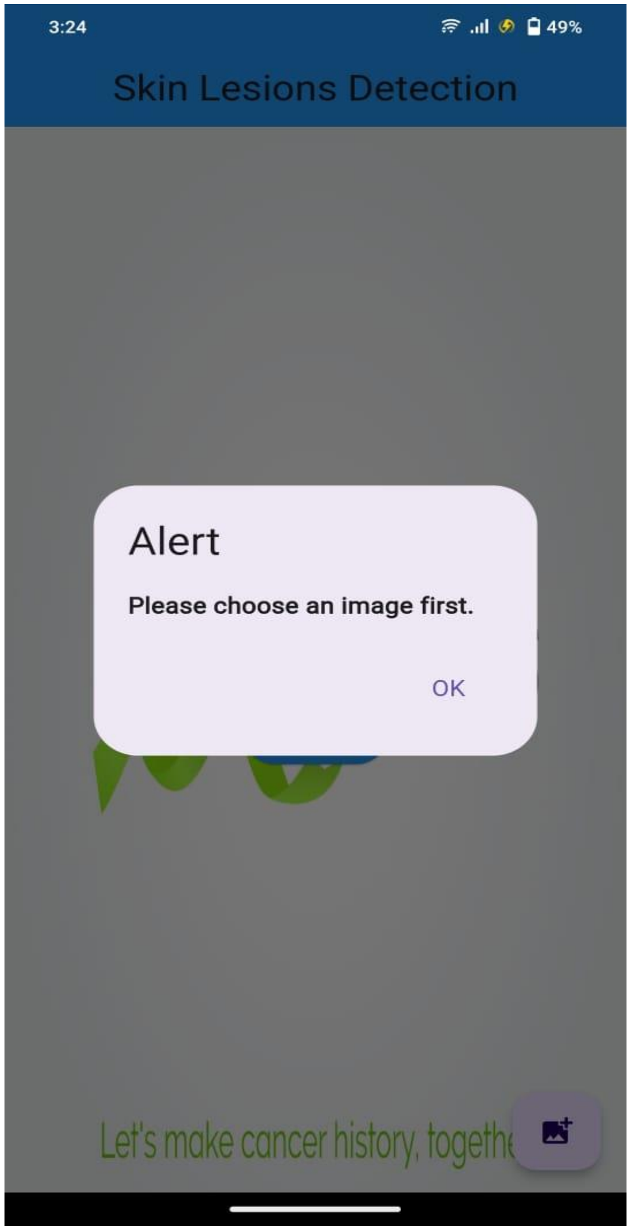

Figure 21 Mobile App Alert Screen

The alert appears if the user tries to detect without uploading image

5.3 Installation Guide:

To install the Flutter application on a mobile device, start by transferring the executable file (APK for Android or IPA for iOS) to the device. For Android devices, the user can transfer the APK file to the device using methods such as email, cloud storage, or direct download. Once the APK file is on the device, navigate to the file location using a file manager app and tap on the APK file to initiate the installation process. Ensure that the device settings allow installations from unknown sources. For iOS devices, the IPA file must be signed and distributed through TestFlight or a similar service, or the device must be configured for development and the app installed through Xcode.

After the installation is complete, open the application on the mobile device. To ensure proper communication with the Flask API, verify that the Flask server is running and accessible over the network. The API endpoint configurations within the Flutter application should point to the correct address of the Flask server. Once the application is running, it should be able to interact with the Flask API to access the deep learning model and perform the required functions.

6- Conclusion and Future Work

6.1 Conclusion

We utilized the Human Against Machine Dataset (HAM10000) and the International Skin Imagaing Dataset (ISIC2018) for our work. The images were resized to a resolution of 299 x 299. Additionally, we applied data augmentations techniques to improve accuracy. Several machine learning and deep learning models, including SVM , Decision Trees, Random Forest, basic CNN, InceptionNet and InceptionResNet.

Among these models, InceptionResNet yielded the best overall results.

We explored various hyperparameters, optimizers, batch sizes, number of epochs, and activation functions for each mentioned model. Our Flutter mobile application was developed and linked with the model by using Flask API to deploy it on portable mobile devices.

Overall, the results obtained were promising, demonstrating the accurate identification of different types of skin lesions from images. The accuracy achieved by the different models ranged from 42% to 96.13%

6.2 Future Work

Future work on this project will focus on making the skin lesion classification model more accurate with higher accuracy and reliable by using advanced techniques and larger datasets. Conducting trials and meeting medical standards will ensure the system works well and is safe in real-world use.

Improving images preprocessing by many steps such as: image segmentation process to remove the background of the image and eliminate noise in the form of hair and air bubbles.

Adding many features to the application:

- Connect the user with the best doctors.
- The user has medical record that can be shared with his doctor.
- Using the image processing techniques to take excellent shot affected area of the skin when the area captures a photo.
- Notify the user to do some periodic checkups.

References

1. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436-44. doi: https://doi.org/10.1038/nature14539.

2. Dildar M, Akram S, Irfan M, et al. Skin cancer detection: a review using deep learning techniques. Int J Environ Res Public Health 2021;18(10):5479. doi: https://doi.org/10.3390/ijerph18105479.

3. Zhang N, Cai YX, Wang YY, et al. Skin cancer diagnosis based on optimized convolutional neural network. Artif Intell Med 2020;102:101756. doi: https://doi.org/10.1016/j.artmed.2019.101756.

4. Daghrir J, Tlig L, Bouchouicha M, et al. Melanoma skin cancer detection using deep learning and classical machine learning techniques: A hybrid approach. Proceedings of 5th international conference on advanced technologies for signal and image processing (ATSIP). IEEE Xplore; 2020. doi: https://doi.org/10.1109/ATSIP49331.2020.9231544.

5. Jenitha A, Amrutha G, Kishore KL, et al. Skin Cancer Recognition and Detection Using Machine Learning Algorithm. Proceedings of the Fist International Conference on Advanced Scientific Innovation in Science, Engineering and Technology. ICASISET; 2020. doi: https://doi.org/10.4108/eai.16-5-2020.2304046.

6. Bhatt H, Shah V, Shah K, Shah R, Shah M. State-of-the-art machine learning techniques for melanoma skin cancer detection and classification: a comprehensive review; 2023. doi: https://doi.org/10.1016/j.imed.2022.08.004.

7. Agarwal K, Singh T. Classification of Skin Cancer Images using Convolutional Neural Networks; 2022.

8. Naqvi M, Gilani SQ, Syed T, Marques O, Kim HC. Skin Cancer Detection Using Deep Learning-A Review; 2023. doi: https://doi.org/10.3390/diagnostics13111911.

9. Nawaz M, Mehmood Z, Nazir T, Naqvi RA, Rehman A, Iqbal M, Saba T. Skin cancer detection from dermoscopic images using deep learning and fuzzy k-means clustering; 2021. doi: https://doi.org/10.1002/jemt.23908.

10. Gouda W, Us Sama N, Al-Waakid G, Humayun M, Zaman Jhanjhi N. Detection of Skin Cancer Based on Skin Lesion Images Using Deep Learning; 2022. doi: https://doi.org/10.3390/healthcare10071183.

11. Deif MA, Hammam RE. Skin Lesions Classification Based on Deep Learning Approach; 2020. doi: https://doi.org/10.1097/JCE.0000000000000405.

12. Hajiarbabi M, Chandra R. Skin Lesion Detection Using Deep Learning. ResearchGate; 2023. doi: https://doi.org/10.1016/j.imed.2022.08.004.

13. Afza F, Sharif M, Khan MA, Naz S, Fernandes SL. A framework of skin lesion detection and recognition based on the deep learning network. Comput Mater Continua 2020;62(3):1313-31. doi: https://doi.org/10.32604/cmc.2020.010340.

14. Tschandl P, Rosendahl C, Kittler H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of

common pigmented skin lesions. Sci Data 2018;5:180161. doi: https://doi.org/10.1038/sdata.2018.161.

15. Codella NCF, Nguyen QB, Pankanti S, et al. Deep learning ensembles for melanoma recognition in dermoscopy images. IBM J Res Dev 2017;61(4):5-1. doi: https://doi.org/10.1147/JRD.2017.2708299.

16. Jafari MH, Karimi N, Samavi S, Soroushmehr SM, Nasr-Esfahani E, Ward K. Skin lesion segmentation in clinical images using deep learning. Proceedings of the 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); 2017 Jul. doi: https://doi.org/10.1109/EMBC.2017.8037112.

17. Khamparia A, Pandey B, Gupta D, et al. Automated early detection of skin cancer using ensemble of deep neural networks. J Ambient Intell Humaniz Comput 2020;11(12):5827-38. doi: https://doi.org/10.1007/s12652-020-01977-8.

18. Pham T, Luong M, Xuan Truong N, et al. An empirical study on leveraging Transfer Learning for skin lesion classification using a highly imbalanced dataset. Proceedings of the 2020 International Conference on Machine Learning and Cybernetics (ICMLC); 2020 Nov. doi: https://doi.org/10.1109/ICMLC50913.2020.9269788.

19. Nida, Naeem MA, Saleem M. Hybrid deep learning model for skin lesion classification using optimized Convolutional Neural Networks and Support Vector Machine. Pattern Recognit Lett 2020;138:88-96. doi: https://doi.org/10.1016/j.patrec.2020.07.026.

20. Gessert N, Nielsen M, Shaikh M, et al. Skin lesion classification using ensembles of multi-resolution EfficientNets with meta data. MethodsX 2020;7:100864. doi: https://doi.org/10.1016/j.mex.2020.100864.

21. Bajwa MN, Nguyen TB, Lee J, et al. A survey on deep learning in medical image analysis: From the perspective of medical imaging. Comput Med Imaging Graph 2020;87:101802. doi: https://doi.org/10.1016/j.compmedimag.2020.101802.

22. Hekler A, Utikal JS, Enk AH, et al. Superior skin cancer classification by the combination of human and artificial intelligence. Eur J Cancer 2019;120:114-21. doi: https://doi.org/10.1016/j.ejca.2019.07.019.

23. Codella NCF, Nguyen QB, Pankanti S, et al. Deep learning ensembles for melanoma recognition in dermoscopy images. IBM J Res Dev 2017;61(4):5-1. doi: https://doi.org/10.1147/JRD.2017.2708299.

24. Jafari MH, Karimi N, Samavi S, Soroushmehr SM, Nasr-Esfahani E, Ward K. Skin lesion segmentation in clinical images using deep learning. Proceedings of the 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); 2017 Jul. doi: https://doi.org/10.1109/EMBC.2017.8037112.

25. Khamparia A, Pandey B, Gupta D, et al. Automated early detection of skin cancer using ensemble of deep neural networks. J Ambient Intell Humaniz Comput 2020;11(12):5827-38. doi: https://doi.org/10.1007/s12652-020-01977-8.

26. Pham T, Luong M, Xuan Truong N, et al. An empirical study on leveraging Transfer Learning for skin lesion classification using a highly imbalanced dataset. Proceedings of the 2020 International Conference on Machine Learning and Cybernetics (ICMLC); 2020 Nov. doi: https://doi.org/10.1109/ICMLC50913.2020.9269788.

27. Nida, Naeem MA, Saleem M. Hybrid deep learning model for skin lesion classification using optimized Convolutional Neural Networks and Support Vector Machine. Pattern Recognit Lett 2020;138:88-96. doi: https://doi.org/10.1016/j.patrec.2020.07.026.

28. Gessert N, Nielsen M, Shaikh M, et al. Skin lesion classification using ensembles of multi-resolution EfficientNets with meta data. MethodsX 2020;7:100864. doi: https://doi.org/10.1016/j.mex.2020.100864.

29. Bajwa MN, Nguyen TB, Lee J, et al. A survey on deep learning in medical image analysis: From the perspective of medical imaging. Comput Med Imaging Graph 2020;87:101802. doi: https://doi.org/10.1016/j.compmedimag.2020.101802.

30. Isic Archive. https://isic-archive.com/

31. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature. 2017;542(7639):115–8. doi: 10.1038/nature21056 . - DOI - PMC – PubMed

32. Gupta AK, Hall DC.J Cosmet Dermatol. 2022 Feb;21(2):530-535. doi: 10.1111/jocd.14681.

33. Google Cloud. https://cloud.google.com/learn/what-is-artificial-intelligence

34. IBM. https://www.ibm.com/topics/machine-learning

35. IBM. https://www.ibm.com/topics/deep-learning

36. C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," arXiv preprint arXiv:1602.07261, Feb. 2016, revised Aug. 2016. doi:https://doi.org/10.48550/arXiv.1602.07261

37. https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202

38. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/#:~:text=Random%20Forest%20is%20a%20widely,both%20classification%20and%20regression%20problems

39. https://scikit-learn.org/stable/modules/tree.html

40. https://medium.com/@davidfagb/understanding-the-support-vector-machine-svm-model-c8eb9bd54a97