Skin Lesion Classification Using Deep Learning

By

Yadnyesh Pande
Somesh Alone
Pratik Papanwar

Under the Guidance

of

Ms. Savita S. Wagre

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Mahatma Gandhi Mission's College of Engineering, Nanded (M.S.)

Academic Year 2024-25

A Project Report on

Skin Lesion Classification Using Deep Learning

Submitted to

DR. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY, LONERE

in fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING

By

Yadnyesh Pande
Somesh Alone
Pratik Papanwar

Under the Guidance
of

Ms. Savita S. Wagre

(Department of Computer Science and Engineering)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MAHATMA GANDHI MISSION'S COLLEGE OF ENGINEERING
NANDED (M.S.)

Academic Year 2024-25

Certificate

This is to certify that the project entitled

"**Skin Lesion Classification Using Deep Learning**"

*being submitted by **Mr. Yadnyesh Pande, Mr. Somesh Alone, Mr. Pratik Papanwar** to the Dr. Babasaheb Ambedkar Technological University, Lonere, for the award of the degree of Bachelor of Technology in Computer Science and Engineering, is a record of bonafide work carried out by them under my supervision and guidance. The matter contained in this report has not been submitted to any other university or institute for the award of any degree.*

Ms. Savita S. Wagre
Project Guide

Dr. A. M. Rajurkar
H.O.D
Computer Science & Engineering

Dr. G. S. Lathkar
Director
MGM's College of Engg., Nanded

ACKNOWLEDGMENT

We are greatly indebted to our project guide **Ms. Savita S. Wagre** for her able guidance throughout this work. It has been an altogether different experience to work with her and we would like to thank her for her help, suggestions and numerous discussions.

We gladly take this opportunity to thank **Dr. Rajurkar A. M.** (Head of Computer Science & Engineering, MGM's College of Engineering, Nanded).

We are heartily thankful to **Dr. Lathkar G. S.** (Director, MGM's College of Engineering, Nanded) for providing facility during progress of project, also for her kindly help, guidance and inspiration.

Last but not least we are also thankful to all those who help directly or indirectly to develop this project and complete it successfully.

With Deep Reverence,

Yadnyesh Pande
Somesh Alone
Pratik Papanwar
[B.Tech CSE-B]

ABSTRACT

Skin lesions represent a wide range of skin conditions, some of which can be harmful if not diagnosed in time. Early and accurate detection is essential, but traditional diagnostic methods often rely on manual analysis, which can be slow and error-prone. To overcome these limitations, deep learning–based systems are being developed to assist in the automatic classification of skin lesions from dermatoscopic images. In this project, we used four models: Convolutional Neural Network (CNN), VGG16, ResNet50, and Support Vector Machine (SVM) to classify skin lesion images from the HAM10000 dataset. The dataset contains various types of skin lesions, allowing the models to learn diverse features. Among all models, ResNet50 demonstrated the highest accuracy due to its deep architecture and ability to extract complex features from the images. Furthermore, to make the system user-friendly, we developed a web application where users can easily upload dermatoscopic images and receive instant predictions of the lesion type. The web application was built using HTML, Tailwind CSS for front-end design, and Flask for backend integration with the trained model. This system aims to support dermatologists by providing fast, consistent, and accurate second opinions, which can ultimately contribute to early diagnosis and better patient care.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Skin lesion classification is a medical imaging process used to identify and categorize different types of skin abnormalities, such as melanoma, nevus, and keratosis. It uses machine learning and deep learning techniques to analyze skin images and assist in early and accurate diagnosis. This helps dermatologists detect skin cancer and other conditions more efficiently. Automated classification supports faster screening and improved patient outcomes.

1.1 Overview of Skin Lesion Classification

The term "skin lesions" refers to abnormal growths or appearances of the skin and can be caused by infections, allergies, genetic conditions or, in some cases, cancer. They have the potential to be anything from benign moles and growths to dangerous types of skin cancer, such as melanoma. Historically, dermatologists have analyzed and categorized these lesions using visual inspections and dermoscopic devices. But these diagnoses can take a long time, are prone to human error, and may take many years of experience to diagnose correctly [1].

In recent years, machine learning (ML) and deep learning approaches have impacted the field of medical diagnostics, particularly for the classification of images. Lesion classification by such techniques consists of teaching computer models to recognize based on patterns, textures, colors, and shapes specific to different kinds of skin lesion, after learning from vast amounts of dermoscopic images. These systems learn to discriminate common lesion types, namely melanocytic nevi, basal cell carcinoma, actinic keratoses, dermatofibroma, and many more. With sufficient data, a well-trained model is able to achieve levels of performance equivalent to expert dermatologists.

The most popular technique in this field is a Convolutional Neural Network (CNN), which is very good in image classification problem. CNNs derive features of input images in an automatic way through a series of convolution and pooling, which makes them capable of capturing subtle differences of lesion patterns. More advanced methods such as transfer learning maximize this feature by relying on pre-trained networks such ResNet, VGG, or MobileNet, which have already trained on millions of images from different domains.

Automatic classification of skin lesions is extremely promising as it it may reduce the load of physicians and help for early detections of skin cancer. This is particularly useful in rural or underserved areas where there is a lack of dermatological care. The mobile and web applications embedding these ML models can analyze user-uploaded skin images in real time, thus democratizing and scaling skin diagnostics. In conclusion, Skin lesion classification through machine learning is an emerging field of research and application that strives to bridge the gap between the need in clinics and the diagnostic capability at hand. Applying new advances of artificial intelligence to improve the accuracy of the diagnosis, reduce the time of response, and, thereby, produce benefits for patients suffering from skin diseases.

1.2. Importance and Applications

Early detection of skin lesions is important to prevent serious health issues. Deep learning models can help doctors by providing quick and accurate classification of skin conditions. This technology can be used in hospitals, clinics, and even mobile apps to assist in diagnosis. It also reduces the need for expensive and time-consuming manual checks.

1.2.1 Importance of Skin Lesion Classification

Cancer of the skin is one of the most prevalent and deadliest of cancers worldwide. According to the World Health Organization (WHO), more than 3 million cases of non- melanoma and about 132,000 cases of melanoma skin cancer are detected annually. Malignant changes in the skin: The earlier they are detected, the higher the probability that the therapy will be successful and the patient will survive. But the early stages of skin cancer can look a lot like harmless blemishes, and it can be tricky to diagnose just by looking at it with the naked eye. This is where ML-based SC systems are essential.

Dermoscopy, biopsy, and histopathological analysis are employed to diagnose skin lesions by dermatologists. Although successful, such techniques are labor-intensive, costly, and are critically dependent upon the availability of experts. In areas where healthcare facilities are scarce, patients suffer from delays or missed diagnoses. Furthermore, even within the selected dermatologic community, variability of diagnosis can result as a result of visual observation being a subjective modality. Automated screening approaches based on large annotated data set can offer standardization of diagnostics, reduction of human bias, and provision of accessibility for remote or under-equipped populations. Machine learning and deep learning models, in particular Convolutional Neural Networks (CNNs), have the power to professionally look after high-definition images of skin lesions and correctly classify them into different categories including the melanoma, nevus, and basal cell carcinoma. This thus provides a real-time, economical, and scalable approach for diagnostic assistance. Such systems can process huge volumes of data very rapidly, thus making faster triaging possible for critical cases that may receive immediate medical attention. Consequently, this moderates the workload of the medical staff and it improves the performance of the medical diagnostic process.

Moreover, incorporating image classification algorithms in tele dermatology apps provides a new opportunity for remote diagnostics. Patients may similarly upload photos from their phones and get instantaneous feedback or risk assessment. This is particularly important for locally advanced and remote rural populations for whom specialist care can be lacking. AI-driven tools can offer the first layer of screening, identifying high-risk patients for referral to a dermatologist for evaluation – minimizing unnecessary face-to-face consultations.

Ongoing research and development in this domain continue to push the boundaries of diagnostic accuracy, speed, and generalizability across diverse skin tones and conditions. Integration of these models with electronic health records (EHRs) and clinical workflows can further streamline medical decision-making. As data quality and computational capabilities improve.

1.2.2 Applications of Skin Lesion Classification

The uses of machine learning skin lesion classification are many and it crosses over into many areas of health, tech, knowledge, and healthcare. Following are such significant applications:

- **Tele dermatology and Remote Healthcare**

The most widespread application is in tele dermatology, where patients send dermoscopic or smartphone-based photographs of skin lesions to be analyzed remotely.AI models integrated into web or mobile apps deliver initial diagnoses or identify high- risk lesions for further scrutiny. This is of great relevance in remote and neglected areas where access to specialist dermatology care could be limited. It is good for patients, because it eliminates the need to travel and it speeds up diagnosis and healthcare in the regions.

- **Clinical Decision Support Systems (CDSS)**

Skin lesion classification models can be plugged into clinical decision support systems and used to provide a second opinion for the dermatologists. These systems raise flags on potential malignancies, propose lesion types and recommend follow-up diagnostic procedures. They are useful for diagnosis, especially for the general practitioner who may not be a dermatologist. This enhances confidence in clinical decision-making, and improves diagnostic concordance among practitioners.

- **Educational and Training Tools**

AI-supported classification tools may even act as interactive teaching platforms for medical students and dermatology trainees. Students are able to upload images, have classification feedback and compare outcomes with clinical outcomes. This visual real- time feedback loop helps to further training and makes it attainable to have visually and manual experience with various types of lesions. Some systems even provide reasons behind each prediction to enhance interpretability and trust in AI predictions.

- **Mobile Health (mHealth) Applications**

Mobile health apps employ AI-based classification models to inform users on-the-fly about concerning skin lesions. These apps typically leverage phone cameras, and one or more embedded models or cloud-based APIs to inspect the images. They educate and motivate the user to consult a medical professional if an at-risk lesion is found. This enables people to engage in preventive healthcare and track changes in their skin over time.

- **Public Health Surveillance and Screening Programs**

Public health authorities could use AI-powered systems for mass screening campaigns in schools, at workplaces, and at community health centers. Given that human resources are constrained, these tools allow large-scale screening without the necessity of manual review of each image. An integral part of this challenge is the development and implementation of such applications that allow to early identify high-risk individuals and to implement action-oriented interventions in areas with high prevalence of skin cancer. It is good for patients, because it eliminates the need to travel and it speeds up diagnosis and healthcare in the regions. Some systems even provide reasons behind each prediction to enhance interpretability and trust in AI predictions.

- **Research and Data Annotation**

Many categories defined by clinicians based on specific diseases, treatments, and symptoms, could be better classified but would require a large set of reliably labelled images. Pre-classifying images with AI systems can help in the annotation and labelling, process, offloading the work to humans. Furthermore, they allow researchers to perform analysis of different classification tasks, investigate novel types of lesions, and evaluate model generalization across populations.

1.2.3 Impact on Healthcare and Society

By automation of detection and classification AI-assisted skin lesion analysis systems contribute to reducing the diagnostic workload from dermatologist and shorten the response time of the entire health care system. The implications are even more profound in the developing world, where medical experts tend to be clustered in urban hubs, and rural populations from which to draw research subjects are underserved. Other than the clinical advantages, such systems also support patient empowerment and proactive health care. People become more conscious of skin irregularities, and can check them periodically at home, instead of visiting a clinic every time they see a new irregularity. AI could save lives and reduce treatment costs through early intervention. Further, these systems are in line with the Sustainable Development Goals (SDGs) by contributing to good health and well-being, addressing inequalities in health access, and harnessing innovations for global health betterment.

1.3 Purpose and Scope

The Purpose and Scope section explains the goal of developing an AI system for classifying skin lesions using deep learning. It also defines the project's coverage, including the models used (CNN, VGG16, ResNet50) and how the system was implemented using Flask and a web interface.

- **Purpose**

This project aims to provide a solution by building a machine learning image classification model that can analyze dermoscopic skin images and classify them to detect dozens of disease categories including melanoma, nevus and other types of skin lesions. The objective is to help doctors and people identify potentially cancerous skin conditions at an earlier stage in order to save both lives, costly treatment, and other interventions. System Concept- The system aims to serve as a decision-support mechanism for dermatologists and as a first-level screen for the masses, especially in regions devoid of specialized healthcare facilities, by automating the classification process.

One of the key objectives in the overall initiative is to explore and benefit from deep learning, especially Convolutional Neural Networks (CNNs) to achieve accurate and repeatable image classification. This translates into training the model on a large and diverse set of dermoscopic images, in such a way that the system will generalize well in the type of skin, ethnicity, and lighting environment in a heterogeneous population. The model should be able to identify subtle visual patterns and textures that may not be perceivable by the human eye, improving diagnostic accuracy and consistency.

- **Scope**

The scale of what is done in this project includes data collection, preprocessing, model building, and evaluation, and deployment of the skin lesion classification system. The dataset used for training and testing is the public dataset HAM10000, which has thousands of skin high resolution images divided into several skin disease classes. The project comprises of normalizing, resizing, augmentation, and balancing as steps to process the dataset for training purposes. We will concentrate on designing a CNN architecture, but we can try a transfer learning approach using pre-trained models such as ResNet or MobileNet for improving upon the accuracy of the selected model. Along with model development, the project includes the development of a web- interface constructed on Flask (Python) to exchange skin images and get live predictions. This renders the solution more available, and feasible for clinic and mass screening. In the future, the platform may be extended to record patient history, warnings and interaction with the health-database.

Although the current scope aims at classifying common types of skin lesions, the scope does not encompass biopsy pathology, treatment advice, or integration within electronic health records. It is not intended as a substitute for clinical diagnosis, but rather as an additional tool. The system's predictions are designed to recommend further medical investigation, not confirm the presence of disease. Further possible extensions are in multi-class lesion segmentation, (smart)phone integration, and (real- time) lesion tracking. Some systems even provide reasons behind each prediction to enhance interpretability and trust in AI predictions. The project also deals with class imbalance and noise in the dataset, both of which are other common occurrences known to affect model performance drastically.

1.4. Challenges in Skin Lesion Classification

The automatic classification of skin lesion images using machine learning and deep learning techniques poses several important challenges, stemming mainly from the high variability in visual appearance of skin lesions. Depending on one's age, skin type, race and environmental exposure, lesions may vary greatly in color, shape, size, texture and location. Even lesions of the same type, such as melanoma, vary in appearance between patients. This intra-class variance makes the models to generalize poorly and they fail to recognize the samples as one class if the training set hardly represents this variance.

Class imbalance in publicly available data sets is another serious problem. For example, there are thousands of images in datasets such as HAM10000, however the data is not evenly distributed among classes, for example, often common benign conditions are over-represented (melanocytic nevi) and, rare but dangerous ones (e.g., dermatofibroma or vascular lesions) are not observed so often. This imbalance might create models which, while optimized for the more common classes, perform poorly for the less frequent cases of clinical importance. Methods such as oversampling, augmented training and using different loss functions to balance training data are adopted to address this problem, but they introduce associated problems in training time and performance.

Image quality and noise also introduce important challenges in automatic classification. Since dermoscopic images are prone to poor illumination, hair occlusion, shadowing, non-uniform magnification, and low resolution, there may be inadequate tissue information available. These artifacts can be easily misidentified by models and leading to a decline in the classification accuracy. Additionally, although unaided clinical photographs provide fewer details in comparison to dermoscopic images, still availability of dermoscopic tools is not universal. Therefore, models need to be strong enough to provide good performance for low/high-quality input images.

Finally, a lack of interpretability and clinical trust in machine learning models is still a challenge in applying them in actual clinical settings. Deep learning models, and CNNs in particular, generally act as black boxes, and outputs are obtained without much understanding of the underlying reasons. Professional workers may distrust automated diagnosis guided by mathematical functions because it remains too opaque. Constructing interpretable AI models and adding visualization methods (for example, saliency maps or Grad-CAM) are key procedures to ensure user trust and prepare it for clinical use.

1.5. Objectives of The Project

The primary aim of this work is to design an intelligent and efficient machine learning- based image classification model to classify skin lesion images into multiple disease categories accurately. Given the increasing rate of skin cancer and other skin diseases all over the world, early and accurate diagnosis has become an essential need. This work seeks to aid in reducing the delay between timely diagnosis and clinical practice by providing an automatic, universally available, and scalable decision support system.

One of the primary goals is: to design an effective deep learning model, such as CNNs, to learn fine-grained visual patterns and features from dermoscopic images. Model: The model must learn to detect four types of spots, which may be used to decide if the lesion is a melanoma, basal cell carcinoma, actinic keratosis, benign keratosis (indicated by the letters 'akiec' for actinic keratosis, or 'bcc' for benign keratosis or 'bkl' for seborrheic keratosis) or vascular lesion. To minimize the false negatives, the project aims to test several different architectures and training techniques that will provide high accuracy, precision, recall, and F1-score to make sure the model will generalize well on the unseen data.

Another important goal is to pre-process and augment the images heavily, to increase the model's generalization performance. This involves scaling, normalization, class balancing and transformation to mimic real-world changes in image collection. The project also deals with class imbalance and noise in the dataset, both of which are other common occurrences known to affect model performance drastically.

1.6 Report Organization

This project report is organized into clearly defined chapters to provide a logical and comprehensive view of the development and implementation of the Skin Lesion Classification System. Each chapter addresses a critical aspect of the system's design, training, testing, and deployment using machine learning and deep learning techniques.

- **Chapter 1: Introduction:** Introduces the background of skin lesion classification, emphasizing its importance in early skin cancer detection. It outlines the objectives, applications, challenges, and scope of the system, along with the role of deep learning models in medical diagnostics.

- **Chapter 2: Literature Survey:** Reviews traditional and modern methods for skin lesion detection, highlighting existing research, datasets (e.g., HAM10000), and the limitations of manual diagnosis. It covers the evolution of ML/DL models like SVM, CNN, ResNet50, and VGG16 in medical image analysis.
- **Chapter 3: Methodology:** Explains the step-by-step methodology used, including dataset preprocessing, feature extraction, and classification approaches. It includes system architecture, image augmentation, and detailed explanations of algorithms used, such as CNN, ResNet50, SVM, and VGG16.
- **Chapter 4: Development Environment and Tools:** Describes the tools and technologies used, including Python, Jupyter Notebook, TensorFlow, Keras, OpenCV, Flask, and Tailwind CSS. It details installation steps, library usage, and software configuration for seamless model development and web deployment.
- **Chapter 5: Results and Discussion:** Presents implementation results of each model, comparing performance metrics like accuracy, precision, recall, F1-score, and confusion matrices. Screenshots of prediction outputs and graphical results are included to visualize system effectiveness and challenges such as class imbalance and misclassification.

Chapter 2

LITERATURE SURVEY

Recent studies have explored the use of deep learning models like CNN, ResNet50, and VGG16 for skin lesion classification. Researchers have found that these models can accurately detect and differentiate between various skin diseases using image datasets like HAM10000. Many papers highlight the effectiveness of image preprocessing and data augmentation to improve model performance. Overall, AI-based approaches show great promise in assisting dermatologists with faster and more reliable skin disease diagnosis.

2.1. Background

In recent years, one of the major health concerns, along with the overall rise in skin-related diseases, has been the development of skin cancer. It is one of the most common cancers diagnosed across the world and can be divided into two groups: melanoma and non-melanoma skin cancer. Even though modern clinical practices and tools allow early and correct diagnosis of damaged skin areas, it still remains a complicated challenge, especially in remote non-urban territories, where qualified dermatologists are not accessible. Thus, there is a significant need for artificial intelligence and machine learning techniques developed to process dermoscopic images automatically.

Skin lesion is an abnormal change on the skin's structure or appearance, which can be categorized into benign lesions that do not contain cancer and malignant lesions. Diagnosis and classification of various skin lesions, such as melanoma, seborrheic keratosis, and various types of benign nevi, often require experienced and precise. Manual experience and observation and some other methods such as matching technique with biopsy results have conventionally been utilized by dermatology to pinpoint and categorize skin lesions. However, such techniques are time-consuming, costly and somewhat subjective, which leads to misdiagnosis or delays diagnosis. The creation of data processing machines and other types of equipment has led to automated help in the classification of lesions [1].

The evolution and success of deep learning in computer vision tasks have brought a revolution to the medical image analysis field. Notably, the use of Convolutional Neural Networks in varied image classification tasks, including the analysis of dermoscopic images, has yielded impressive results. A CNN can autonomously learn complicated patterns and features from raw image data, eliminating the need to perform manual feature extraction. As a result, CNNs are ideal for dealing with the vast range in the appearances and structures of skin.

In the last decades many works have been devoted to the generation of automatic skin lesion classifiers. These papers have employed a variety of ML methods including classically used ones like SVMs and random forests to state-of-the-art deep learning architectures (e.g., ResNet, Inception, and DenseNet). Many of these models are trained and validated using massive public datasets like ISIC Archive, PH2, and HAM10000. These datasets contain the annotated dermoscopic images and have been used as reference standard for comparison of the performance of the different systems.

Furthermore, the literature of this field also focuses on preprocessing and data augmentation techniques and transfer learning methods that can enhance model robustness and accuracy. Preprocessing techniques, such as resizing, normalization of images, and artifact removal, assist in the normalization of the input data, power augmentation in expanding the amount of data synthetically to lower overfitting. As a result of the restricted labeled medical images, transfer learning, or a method of fine-tuning pre-trained CNN models such as VGG16 and ResNet50 on skin lesion datasets, was widely used [1].

Despite these encouraging results, the literature brings to the fore a number of challenges in this field as well, including class imbalance, lack of diversity in datasets, low explainability of deep learning models, and overfitting when training on small annotated data. Such limitations in previous works provide strong motivations for the present work in which we anticipate not only training a reliable CNN-based model, but also considering practicality issues, including real-time implementation, user the above chapter thoroughly reviews the previous studies and discusses the traditional as well as deep learning classification models, traditional datasets, performance measurement algorithms, and pros and cons of the existing research. The findings from this review are used for setting the research direction and support the architectural and methodical designs in this study. However, such techniques are time-consuming, costly and somewhat subjective, which leads to misdiagnosis or delays diagnosis. The creation of data processing machines and other types of equipment has led to automated help in the classification of lesions.

2.2 Traditional Methods for Skin Lesion Detection

Traditional methods for skin lesion detection often rely on manual examination by dermatologists using tools like dermoscopy. These methods can be time-consuming, subjective, and may vary in accuracy based on a doctor's experience. Early diagnosis can be difficult, especially in remote areas without expert access.

- **Manual Diagnosis Through Dermoscopy**

For evidence of skin lesion detection methodologies before automated or computer-aided diagnostic tools, one might refer to the manual examination used by dermatologists. One of the digital non-invasive imaging methodologies was dermoscopy, which gives magnification to the plain skin and helps to see pigmentation patterns and vascular structures more clearly. This method provided an extended examination view against the naked eye, capturing the subsurface of the skin and enabling early melanoma diagnosis.

The ABCD Rule of Dermoscopy is probably the most well accepted manual method, which classifies lesions according to Asymmetry, Border, Color, and Diameter: Benign display is generally even and stable, whereas malignant display is asymmetrical. Diffuse and obscure boundaries between colors are indicators of malignancy. More than one color is more probable melanoma. A diameter larger than 6 mm is more likely to be malignant.

Despite the standardized framework, this rule was still qualitative rather than quantitative. It is impossible to rely entirely on the knowledge and experience of a dermatologist. Sometimes, especially in rural or underdeveloped areas of dermatological expertise, one receives a misdiagnosis. Another factor is the human factor in the interpretation. People have different visual perceptions, and distinguishing one characteristic from another with a similar visage is a very complicated task. Along with it, such factors as a location of a particular lesion, skin color, lighting conditions, and the patient’s medical history make this task even more challenging. Subjectivity and expertise requirements made it necessary to develop systems. However, such techniques are time-consuming, costly and somewhat subjective, which leads to misdiagnosis or delays diagnosis. The creation of data processing machines and other types of equipment has led to automated help in the classification of lesions. This method provided an extended examination view against the naked eye, capturing the subsurface of the skin and enabling early melanoma diagnosis.

- **Introduction of Traditional Machine Learning Methods**

In an attempt to improve diagnostic accuracy while minimizing human errors, researchers started to rely on conventional machine learning approaches for automatic skin lesion classification. More specifically, these models were designed to mimic or inform clinical decision-making processes through pattern recognition in labeled dermoscopic image datasets. Common characteristics of these traditional ML-based systems included the end-to-end following steps: preprocessing steps to eliminate artifacts, such as hair, noise, or reflections in images; segmentation steps to distinguish the lesion from the surrounding healthy skin; feature extraction steps to manually design numerical features reflecting the lesion's color, shape, and texture; classification steps with subsequent assignment of the identified lesion to predefined classes.

They particularly perform well with high-dimensional data. SVMs have made a significant contribution in the analysis by separating malignant and benign lesions through boundary optimization in the feature space. As indicated by the author, this simple algorithm classifies a lesion into the group that contains the majority of the lesions among its closest 'k' neighbors in the feature space. The k-NN classifier was easy to apply but was sensitive to feature scaling and various degrees of imbalanced data. Decision Trees and Random Forests Section II). These tree-based models created rules that were easy to understand and interpret.

2.3 Datasets Used in Skin Lesion Research

As shown, In Fig. 2.1 the performance of any machine learning model, particularly in medical image analysis, relies on the quality, size, and diversity of the dataset. In the area of skin lesion classification, datasets should be made up of high-resolution images to aid in the texture, colour, and structure description of skin anomalies. The metadata available in the dataset includes age, sex, and anatomical site, which can be beneficial for multi- modal approaches in the future [2]. All images are in RGB color space and at various resolutions and have been acquired from diverse subjects and imaging apparatuses. Several benchmark datasets have been constructed to standardize and facilitate the development of automated dermatological diagnosis.

Among these, the ISIC Archive, PH2, and HAM10000 datasets are frequently cited. Which has been essentially designed just for this project.

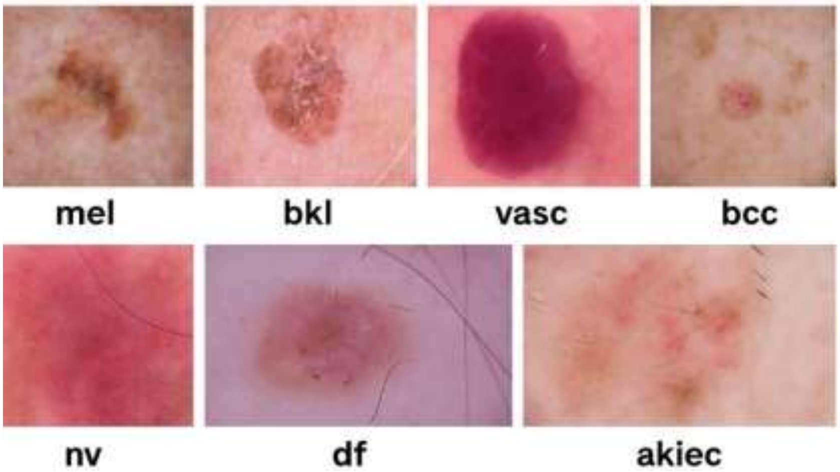


Fig. 2.1 Skin Lesion Classification Image

We only focused on HAM10000 (Human Against Machine with 10,000 training images) since it's more diverse and larger-scale machine learning research.

- **HAM10000 Dataset**

The HAM10000 (Human Against Machine with 10000 training images) dataset is a comprehensive and widely-used benchmark dataset in the field of dermatology and medical image analysis. It contains 10,015 high-quality dermatoscopic images of pigmented skin lesions, making it a valuable resource for developing and evaluating machine learning models for skin disease classification. The dataset was collected from multiple sources, including different populations and clinical settings, to ensure diversity in skin tones, lesion types, and imaging conditions. The lesions are annotated with expert dermatological diagnoses and cover seven different diagnostic categories, including melanoma, melanocytic nevi, basal cell carcinoma, actinic keratoses, benign keratosis-like lesions, dermatofibroma, and vascular lesions. Each image in the dataset has been carefully curated and labeled to provide a reliable ground truth, facilitating research in automated diagnosis, computer-aided detection, and deep learning-based classification systems [2]. HAM10000 plays a crucial role in advancing artificial intelligence solutions in dermatology by enabling reproducible experiments and comparisons among various models and algorithms.

Chapter 3

METHODOLOGY USED

The skin lesion classification system follows several key steps. First, images are collected from the HAM10000 dataset and preprocessed (resized, normalized). Then, deep learning models like CNN, ResNet50, VGG16, and SVM are trained on the processed data. After training, the models predict the type of skin lesion from new images.

3.1 System Architecture

As shown in Fig. 3.1 the system architecture below provides an overview of the system's operation. The working of the system can be shown below to easily understand the project:

The proposed system architecture for skin lesion classification is designed to process input skin images and classify them into various disease categories using machine learning and deep learning techniques. The process begins with an input image, which is first passed through a preprocessing stage. This step involves removing noise and hair from the image to enhance its clarity and ensure that irrelevant features do not interfere with the classification. Once preprocessing is complete, data augmentation techniques such as rotation, flipping, cropping, and zooming are applied to artificially expand the dataset and improve model robustness by simulating real- world variations in the image data [1].

Following augmentation, the system performs feature extraction to capture meaningful patterns from the images. Two types of features are extracted: GLCM (Gray Level Co-occurrence Matrix) features, such as correlation to capture texture patterns, and statistical features like mean, variance, and standard deviation to understand the intensity distribution and structure of the lesion area. These extracted features are then used as input to different classification algorithms. Traditional machine learning models like Support Vector Machine (SVM) utilize manually extracted features, while deep learning models such as Convolutional Neural Networks (CNN), ResNet, and VGG automatically learn hierarchical representations from the data. The system is trained and tested using these algorithms to accurately classify the images into specific disease categories.

The system architecture of the method proposed for skin lesion classification processes the input skin images, and the input skin images are classified into different disease types based on machine learning and deep learning. The first step is that the input image is pre-processed. This phase consists in removing the noise and the hair of the image to improve its clarity and avoiding being interfered with the classification by irrelevant characteristics. After the preprocessing steps, data augmentation, which includes rotation, flipping, cropping, and zooming, is performed to artificially increase the sample size and strengthen the model's generalization to real-world variations in the image data.

After augmentation, the model does feature extraction of the images. Two sets of features are extracted: GLCM (Gray Level Co-occurrence Matrix) based features, e.g., correlation for texture patterns, and statistical features (mean, variance, and standard deviation) for intensity distribution, and structure on the lesion region. Features are then extracted as inputs of various classifiers. Conventional machine learning methods, such as Support Vector Machine (SVM), are generally based on hand-crafted features. In contrast, deep learning approaches, such as Convolutional Neural Networks (CNN), ResNet, and VGG, automatically learn feature hierarchy from the data [1].

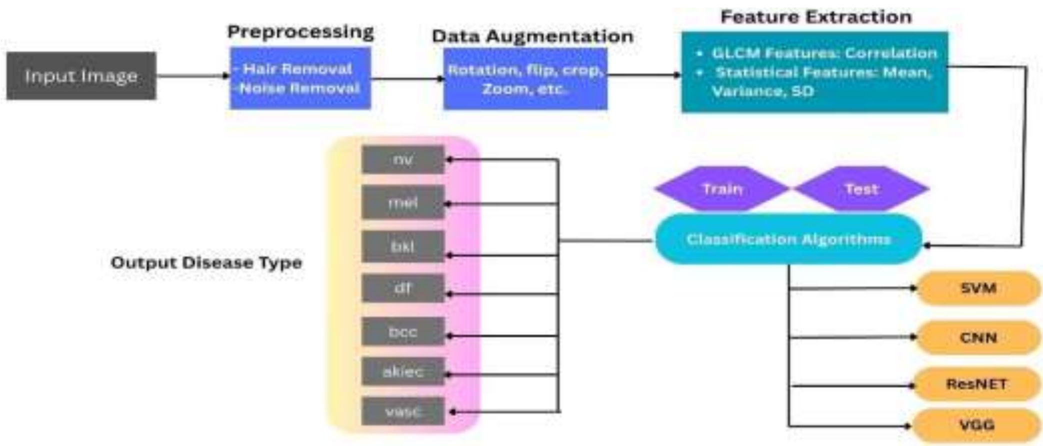


Fig. 3.1 System Architecture

The output layer of the system is the predicted disease type, which can be classified from different categories here: Melanocytic Nevi (nv), Melanoma (mel), Benign Keratosis-like Lesions (bkl), Dermatofibroma (df), Basal Cell Carcinoma (bcc), and Vascular Lesions (vasc). This well-structured pipeline of automatic classification helps in the early detection of such skin diseases, which is a significant point in timely diagnosis and treatment.

The final output of the system is the predicted disease type, which includes categories such as Melanocytic Nevi (nv), Melanoma (mel), Benign Keratosis-like Lesions (bkl), Dermatofibroma (df), Basal Cell Carcinoma (bcc), and Vascular Lesions (vasc). By automating the classification process through this well-structured pipeline, the model aids in the early detection of skin conditions, which is crucial for timely diagnosis and treatment.

3.2 Dataset Details

For this project, we used the HAM10000 dataset (Human Against Machine with 10000 training images), which is one of the most widely used benchmark datasets for skin lesion analysis. This dataset was created to support research in the field of dermatology by providing high-quality dermatoscopic images of various types of skin lesions.

In the present work, the HAM10000 dataset (Human Against Machine with 10000 training images) is utilized, which is one of the most popular benchmark datasets for skin lesion analysis. The data set HAM10000 contains 10,015 dermatoscopic images, taken from different populations and regions (Austrian, and Australian). Each picture contains a close-up of a skin lesion, as well as expert-verified labels on what the skin condition is. The dataset has images of seven classes of skin lesions, making it convenient for multi-class classification tasks [3].

- **Disease Categories in the Dataset**

As shown in table: 3.1 the images are labeled into the following **seven classes:**

- Melanocytic nevi (nv)
- Melanoma (mel)
- Benign keratosis-like lesions (bkl)
- Basal cell carcinoma (bcc)
- Actinic keratoses (akiec)
- Vascular lesions (vasc)
- Dermatofibroma (df)

There is metadata in the dataset such as sex, age, and anatomical site which might be useful for fair multi-modal approaches in the future. All of the data is RGB color images and is recorded at different resolutions on different outfits, and positions. The metadata available in the dataset includes age, sex, and anatomical site, which can be beneficial for multi-modal approaches in the future.

Feature	Explanation	Measurement	Range / Values
image_id	Unique ID for each dermatoscopic image	Text (String)	Unique alphanumeric string
lesion_id	ID for the lesion (can be same for multiple img)	Text (String)	Alphanumeric
dx	Diagnosis of the lesion (7 disease types)	Categorical	['akiec', 'bcc', 'bkl', 'df', 'mel', 'nv', 'vasc']
dx_type	Method of diagnosis	Categorical	['histo', 'follow_up', 'consensus', 'confocal']
age	Age of patient	Integer (Years)	0 – 100+
sex	Sex of patient	Categorical	['male', 'female'] (can be null)
localization	Location of the lesion on the body	Categorical	['back', 'lower extremity', 'abdomen'…]
dataset	Dataset name (constant)	Text (String)	"HAM10000"
target_label	Encoded label for 0–6 classification (e.g. 0 to 6)	Integer (0–6)	0: akiec, 1: bcc, … 6: vasc

Table 3.1: Dataset Information

All images are in RGB color space and at various resolutions and have been acquired from diverse subjects and imaging apparatuses.

3.3 Algorithms Used in the Project

In machine learning, different types of algorithms are used to train models based on data patterns. In this project, we used CNN, VGG16, and ResNet50 for classifying skin lesions effectively.

3.3.1 Convolutional Neural Network

As shown in Fig.3.2 Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed to operate on data with a grid-like topology, such as images. They are the cornerstone of the majority of contemporary computer vision applications, and they allow for identifying features in visual data. CNN

(Convolutional Neural Network) is a deep learning model that has been proposed for the classification of images. Through layers such as the convolutional layer, the pooling layer, and a fully connected one, it extracts spatial features of images automatically. In the current study, a novel customized CNN architecture was designed for processing the skin lesion images for classification. Several convolutional layers that are followed by ReLU, max-pooling, and a set of dense layers at the end for classification compose the model. CNNs are powerful for medical images in particular since they can directly learn intricate patterns from raw pixels without manual feature engineering.

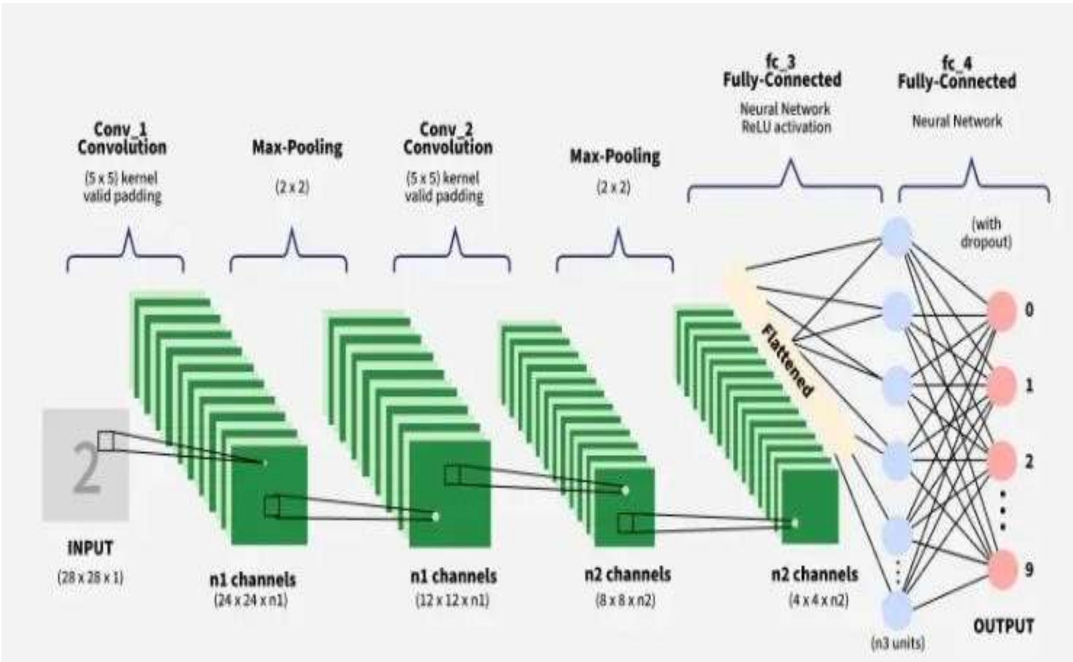


Fig 3.2: CNN Architecture

How CNNs Work?

1. **Input Image:** CNN receives an input image which is pre-processed to ensure uniformity in size and format.
2. **Convolutional Layers:** Filters are applied to the input image to extract features like edges, textures, and shapes.
3. **Pooling Layers:** The feature maps generated by the convolutional layers are down sampled to reduce dimensionality.
4. **Fully Connected Layers:** The down sampled feature maps are passed through fully connected layers to produce the final output, such as a classification label.
5. **Output:** The CNN outputs a prediction, such as the class of the image.

3.3.2 ReNet50 (Residual Networks)

As shown in Fig.3.3 ResNet50 is a deep residual neural network architecture introduced by Microsoft Research in 2015. It consists of 50 layers and is renowned for introducing the concept of residual learning with skip connections, which address the issue of vanishing gradients in very deep networks. Instead of learning a direct mapping, ResNet learns the residual (difference) between the input and output of layers. These skip connections allow gradients to flow more effectively during backpropagation, thereby enabling the training of much deeper networks without degradation in performance. The architecture is divided into four main stages, each containing several convolutional and identity blocks that work together to extract complex features from images [6].

ResNet50 uses batch normalization and ReLU activations for efficient training. It is also widely used in transfer learning applications as it comes pretrained on large datasets like ImageNet, allowing developers to fine-tune it for specific tasks with limited data. The intuition of this network is instead of letting the layers fit the desired underlying mapping, we add the mapping directly (shortcut) from input to output and let the network fit the residual mapping. So rather than say H(x), this initial mapping, let the network itself learns.

F(x) = H(x) - x is the "residual", and it is often easier to represent a smaller residual to learn.

The network then adds this residual F(x) to the input x via a 'shortcut 'skip connection, which effectively turns H(x) = F(x) + x.

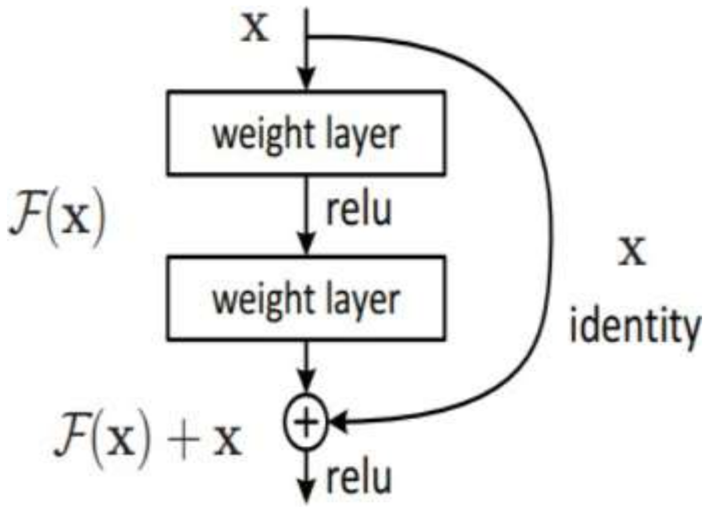


Fig 3.3: ResNet50 Architecture

3.3.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a classical machine learning algorithm known for its effectiveness in high-dimensional classification tasks. It works by finding the optimal hyperplane that maximizes the margin between different classes. In this project, SVM was applied using features manually extracted from skin images, such as texture and statistical features (e.g., mean, variance, GLCM features). SVM served as a baseline model and provided a performance benchmark against the more advanced deep learning models.

3.3.4 Visual Geometry Group (VGG)

As shown in Fig.3.4 VGG16 is a deep Convolutional Neural Network (CNN) architecture introduced by the Visual Geometry Group from the University of Oxford in 2014, particularly as part of the ILSVRC (ImageNet Large Scale Visual Recognition Challenge). The model gained widespread recognition due to its balance of simplicity and high performance. VGG16 consists of 16 weight layers—specifically, 13 convolutional layers followed by 3 fully connected layers. The defining characteristic of VGG16 is its use of very small convolutional filters, primarily of size 3×3 with a stride of 1, and 2×2 max pooling layers with a stride of 2 for down sampling.

This choice of small filter size allows the network to capture fine-grained features of an image while keeping the model's complexity manageable. Stacking multiple 3×3 filters have the advantage of achieving the same receptive field as a larger 5×5 or 7×7 filter, but with fewer parameters and more non-linearities, which enhances the network's ability to learn complex patterns. VGG16 uses the ReLU (Rectified Linear Unit) activation function after each convolutional layer, which introduces non-linearity into the model and helps in faster convergence during training by mitigating vanishing gradient issues. The architecture concludes with three fully connected layers, followed by a SoftMax activation function in the final layer for multi-class classification tasks.

Despite being relatively deep compared to earlier models, VGG16 has a very uniform and simple architecture, making it highly adaptable and easy to implement using standard deep learning frameworks like TensorFlow and PyTorch. It has been widely used as a feature extractor for transfer learning in tasks such as image classification, object detection, and face recognition. However, one notable drawback of VGG16 is its large number of parameters (approximately 138 million), which leads to high memory consumption and increased computational cost during training and inference. Nevertheless, due to its performance and interpretability, VGG16 remains a popular baseline architecture in computer vision research and applications.

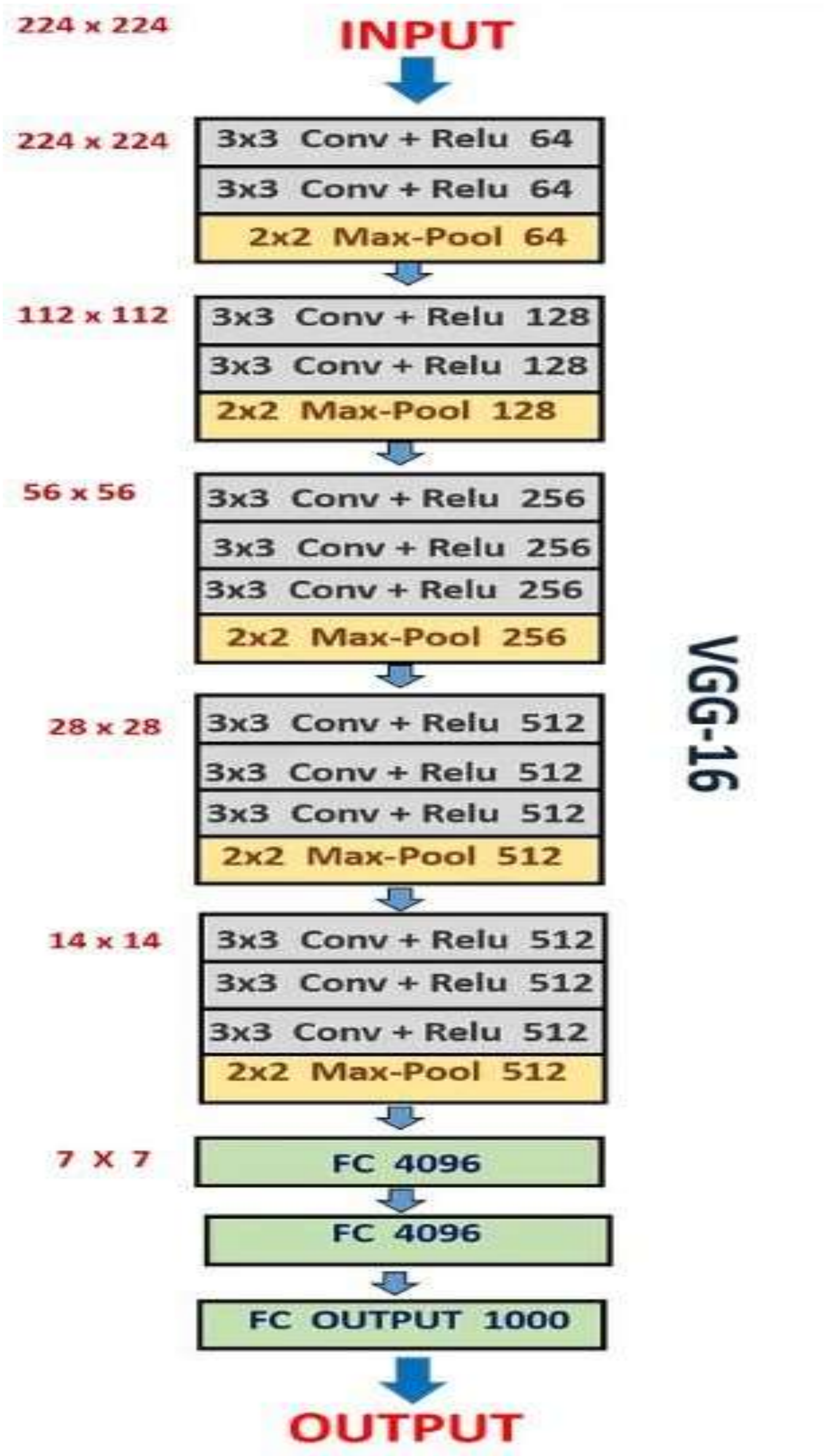


Fig 3.4 VGG16 Architecture

Chapter 4

DEVELOPMENT ENVIRONMENT AND TOOLS

This project was developed using Python as the core programming language. For model building and training, libraries like TensorFlow, Kera's, and scikit-learn were used. The web interface was built using HTML, Tailwind CSS, and Flask. The system was implemented and tested in Jupyter Notebook and deployed through a lightweight Flask server.

4.1 Introduction to Jupyter Notebook

Jupyter Notebook Jupyter Notebook is an open-source web application for interactive development that users can use to create and share documents with live code, equations, visualizations and narrative text. It is popular with the data science, machine learning, and research communities due to its unique capability of interweaving executable code with documentation in an integrated environment.

Jupyter Notebook as primary development environment for this project We selected a Jupyter Notebook as our development workhorse for several reasons in this project. First, it offers an integrated and incremental code execution UI, which accelerates debugging and testing ML models. It also supports Markdown cells which means you can write lot of documentation of your analysis right beside your code.The intuitive visualizations using libraries like Matplotlib and Seaborn are generated directly inside the notebook, enabling the user to interpret data trends, model performance, or results as they come.

The flexibility of Jupyter allowed for running Python code within isolated cells, viewing visualizations and metrics in real time, and organizing experimentation in a structured manner, which made Jupyter perfect for developing, training, and analyzing the skin lesion classifiers. It also encourages the integration of some of the most commonly used Python libraries including TensorFlow, Keras, Scikit-learn, OpenCV and Pandas; that all used heavily in this project. Jupyter Notebook main features are:

- **Interactive Coding Environment:** It is an interactive programming environment where you can run code block-wise, allowing for step-by-step execution. This makes it especially useful for automated testing and debugging in large projects. It also enhances productivity by providing instant feedback during code development. Additionally, it supports integration with visualizations and tools, making it ideal for data science and machine learning workflows.

- **Data Visualization:** Visual outputs such as charts, graphs, images from libraries like Matplotlib, Seaborn, Plotly, etc., are shown natively in the notebook interface, hence facilitating quick analysis and interpretation.
- **Integration with Libraries:** Jupyter integrates easily with widely used Python libraries such as NumPy, Pandas, Scikit-learn, TensorFlow, OpenCV, and Keras, making it advantageous for machine learning and data science workflows.
- **Open Source and Free:** Jupyter is a free-to-use and open-source system, supported with the help of a global community of contributors. It can be deployed both at the local system and can be run in the cloud with the help of platforms like Google Colab and Binder.

4.2. Overview of OpenCV

OpenCV (Open-Source Computer Vision Library) is a powerful and versatile open-source computer vision and image processing library originally developed by Intel. It is primarily written in C++ but also provides strong support for Python, making it widely accessible to the machine learning and data science communities. From basic image loading and manipulation to complex object detection and recognition tasks, OpenCV delivers high performance and flexibility. It supports integration with deep learning frameworks like TensorFlow and PyTorch, allowing advanced AI-based vision solutions. Additionally, OpenCV is cross-platform and works seamlessly on Windows, Linux, macOS, and even mobile platforms like Android and iOS.In the context of deep learning and AI-driven medical diagnosis, OpenCV plays an instrumental role in preprocessing image data before feeding it into machine learning models.

For this project on skin lesion classification, OpenCV was used for several essential tasks such as loading images, resizing them to a fixed input size (commonly 224x224 or 256x256 pixels), and normalizing pixel intensity values to a scale of 0–1. These preprocessing steps help maintain consistency in input data and improve the training efficiency and accuracy of deep learning models. OpenCV also supports the conversion of color spaces — for example, converting RGB images to grayscale — which can simplify the learning task by reducing input complexity while retainingcritical visual features. Grayscale images are particularly useful when color information is not essential, allowing the model to focus on texture, edges, and shape.

One of OpenCV's most significant strengths lies in its advanced image processing capabilities. It provides a wide range of tools for edge detection (such as the Canny edge detector), thresholding (both global and adaptive), and contour detection, which are highly beneficial in identifying and segmenting the regions of interest within medical images. These tools are useful for highlighting lesion boundaries, detecting abnormalities, and preparing the image for further feature extraction. In addition, OpenCV allows for powerful image filtering using techniques like Gaussian Blur, Median Blur, and bilateral filtering, which can enhance image clarity and reduce noise, thereby improving model interpretability.

OpenCV also enables effective data augmentation, an important technique in deep learning used to artificially expand the dataset. By rotating, flipping, cropping, and zooming images, OpenCV helps introduce variability into the training set, which aids in reducing overfitting and improving the model's generalization capability. These augmentations ensure that the model does not rely on specific image orientations or lighting conditions, making it more robust in real-world scenarios. Furthermore, OpenCV's interoperability with NumPy and seamless integration with TensorFlow, Keras, and PyTorch libraries make it ideal for use in deep learning pipelines.

Beyond preprocessing, OpenCV can be used during model inference and visualization. It allows developers to draw bounding boxes, highlight areas of interest, and annotate images with predicted labels and confidence scores. This can be extremely useful for building user-friendly interfaces for medical practitioners and researchers. OpenCV is also suitable for deployment in real-time systems such as mobile health apps, telemedicine platforms, and automated diagnostic kiosks due to its high-speed performance and lightweight architecture.

Overall, OpenCV is an indispensable tool for building AI-based image classification systems, especially in the field of healthcare and medical imaging. Its ability to handle both basic and complex image operations with efficiency, combined with its wide compatibility with machine learning frameworks, makes it a cornerstone of modern computer vision solutions.

4.3. Software and Libraries Used

The project uses several software tools and libraries for efficient development and deployment. Key libraries include TensorFlow, Keras, NumPy, Pandas, and Matplotlib for deep learning and data processing. Flask was used for backend web integration, and Tailwind CSS for responsive UI design.

4.3.1 Python

Python is an interpreted, high-level programming language that is easy to learn and read. Its intuitive and easy-to-use syntax enables developers to quickly write clear, efficient code, whether you're an experienced developer or a complete beginner. Python supports various programming paradigms, including procedural, object-oriented, and functional programming, offering flexibility in software development. It has a large standard library and a vibrant ecosystem of third-party libraries, making it suitable for a wide range of applications such as web development, data analysis, artificial intelligence, machine learning, automation, and scientific computing.

4.3.2 Pandas

Pandas provides a powerful toolkit for working with structured and unstructured data. At its core are two primary data structures – Data Frames and Series - that enable users to efficiently wrangle, analyze, and visualize even massive datasets. With an intuitive yet flexible syntax resembling common spreadsheet software, pandas allows users to quickly slice, dice, aggregate, filter, and reshape data from a wide array of sources. These sources include everything from basic CSV and Excel files to more complex databases like SQL and JSON.

4.3.3 NumPy

NumPy (Numerical Python) is a powerful library in Python used for working with arrays and matrices, as well as performing a wide range of mathematical operations on these data structures. With an intuitive yet flexible syntax resembling common spreadsheet software, pandas allows users to quickly slice, dice, aggregate, filter, and reshape data from a wide array of sources. It is one of the most fundamental libraries for scientific computing in Python and serves as the backbone for many other libraries, including Pandas, Matplotlib, and Scikit learn. It contains various features including:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions

- Provides efficient mathematical and statistical operations.
- Enables array broadcasting for operations on different shapes.

4.3.4 Matplotlib

Matplotlib is a widely used data visualization library in Python that allows users to create static, animated, and interactive plots. In this project, Matplotlib was essential for visualizing image samples, class distributions, training performance (loss and accuracy curves), and evaluation metrics like the confusion matrix. It enabled easy tracking of model learning behavior during training and validation phases. The library's flexibility in customizing plot elements such as labels, titles, and legends made it ideal for generating clear, publication-ready visualizations that support model analysis and reporting.

4.3.5 TensorFlow

TensorFlow is a powerful open-source deep learning framework developed by Google. It provides a comprehensive ecosystem for building, training, and deploying machine learning models. In this project, TensorFlow was used to implement and train deep learning architectures like CNN, ResNet50, and VGG16. Its high-level API, Keras, allowed for quick model development and experimentation. TensorFlow also supports GPU acceleration, which significantly reduced training time and improved model performance during image classification tasks.

4.3.6 PIL (Python Imaging Library)

PIL, now maintained under the name Pillow, is a Python library used for opening, manipulating, and saving image files. In this project, PIL was used for basic image operations such as loading images, resizing them to the required input dimensions, and converting between image formats. It provided a simple and efficient way to preprocess skin lesion images before feeding them into machine learning models, making it an essential tool for handling image data in the early stages of the pipeline.

4.3.7 Flask

Flask is a lightweight and flexible web framework for Python that is widely used for building web applications, APIs, and services. It is particularly popular for creating RESTful APIs and prototyping machine learning models. Flask provides the essential tools and features to create web-based applications but without the overhead and complexity of larger frameworks like Django. Below are the key features and functionalities of Flask are:

- User Input: A user enters their health data on a webpage (age, BMI, glucose level, etc.).
- API Request: The input data is sent to a Flask RESTful API via an HTTP POST request.
- Model Prediction: Flask loads the pre-trained machine learning model, processes the input, and generates a diabetes risk prediction.

4.3.8 HTML (HyperText Markup Language)

HTML is the standard markup language used to create the structure of web pages. It provides the foundational layout by defining elements like headings, paragraphs, buttons, input fields, and image placeholders. In this project, HTML was used to build the front-end interface of the web application, where users can upload skin lesion images and view prediction results

4.3.9 CSS (Cascading Style Sheets)

CSS is used to control the visual appearance and styling of HTML elements on a webpage. It allows developers to define the layout, colors, fonts, spacing, and responsiveness of a site. In this project, CSS was applied to make the web interface attractive and user-friendly. It ensured that the application was visually appealing, responsive across different devices, and consistent in its design, contributing to a user experience when interacting with the skin lesion classification system.

4.3.10 JavaScript

JavaScript is a high-level, dynamic programming language primarily used to add interactivity and client-side logic to web pages. It enables features like real-time feedback, form validation, dynamic content loading, and asynchronous communication with the backend using AJAX or fetch APIs. In this project, JavaScript was used to handle user interactions such as image uploads, displaying prediction results without reloading the page, and connecting the front end to the Flask backend for model inference.

4.4. Installation and Configuration

In the Skin Lesion Classification project, the installation and configuration of the development environment are critical steps that lay the foundation for successful implementation and execution. It ensured that the application was visually appealing, responsive across different devices, and consistent in its design, contributing to a user experience when interacting with the skin lesion classification system. This process involves setting up the necessary software dependencies, libraries, and tools required for developing, testing, and deploying the Skin Lesion Classification system effectively.

- **Python Environment Setup**

The first step in the installation process is setting up a Python environment. Python serves as the primary programming language for the project, providing a rich ecosystem of libraries and frameworks conducive to machine learning and computer vision tasks. Developers typically choose a Python distribution like Anaconda, which includes essential packages such as NumPy, OpenCV and Matplotlib, streamlining the setup process.

- **Library Installation**

Once the Python environment is established, the next step is to install the required libraries and dependencies. This includes frameworks like OpenCV, NumPy, and Matplotlib, along with any additional libraries specific to the project's needs. The installation is usually done using package managers like pip or Conda, ensuring that the correct versions and dependencies are resolved automatically.

- **Configuration for Development**

After installing the necessary libraries, developers configure their development environment for efficient coding and testing. This involves setting up integrated development environments (IDEs) such as Jupyter Notebook, PyCharm, or Visual Studio Code, which provide features like code autocompletion, debugging tools, and project management capabilities. IDE enhances productivity and facilitates collaborative development among team members.

- **Hardware Setup**

In addition to software setup, configuring the hardware environment is crucial for optimal performance during pose estimation tasks. This includes ensuring that the computer or device used for development and testing meets the minimum system requirements for running computationally intensive algorithms. Hardware considerations may also include GPU acceleration for faster processing speeds, especially in real-time applications.

- **Testing and Validation Environment**

As part of the configuration process, developers set up testing and validation environments to evaluate the Skin Lesion system's accuracy, performance, and reliability. This involves creating test datasets, defining evaluation metrics, and conducting rigorous testing scenarios to validate the system's functionality.

Chapter 5

EXPERIMENTAL RESULTS

The project is implemented using Python with libraries like TensorFlow, Keras, OpenCV, and Flask. Preprocessing includes resizing and normalizing images from the HAM10000 dataset. Deep learning models such as CNN, ResNet50, VGG16, and SVM are trained for classification. A Flask-based web interface is developed to allow users to upload images and view prediction results with confidence scores.

5.1 Dataset Preparation and Preprocessing

- **Loading the HAM10000 Dataset**

The HAM10000 (Human Against Machine with 10,000 training images) dataset is a widely used public dataset consisting of 10,015 dermatoscopic images categorized into seven types of pigmented skin lesions. To begin the implementation, the dataset was downloaded and loaded into the Python environment. The images were stored in designated folders, while the accompanying metadata, containing information about each lesion's diagnosis and patient details, was loaded using the pandas library. The metadata was crucial for associating each image with its corresponding label, allowing supervised learning to be performed. The dataset was carefully indexed to ensure each image path was correctly linked to its diagnosis label for further processing.

- **Cleaning and Formatting Data**

Before training the models, the dataset underwent thorough cleaning and formatting. This involved checking for missing values, corrupted images, or mislabeled data points in the metadata file. Any inconsistencies or duplicates were identified and removed to prevent bias or errors during model training. Labels were standardized by converting diagnosis strings to uniform class names, which made it easier to map to numeric class indices required for classification algorithms. This step ensured data integrity and consistency, which are vital for achieving reliable model performance.

- **Image Resizing and Normalization**

Deep learning models such as CNN, ResNet50, and VGG16 require input images of fixed dimensions. Therefore, all images in the HAM10000 dataset were resized to a uniform size of 224x224 pixels using the OpenCV library's resizing functions. This standard size is commonly used in pre-trained architectures, allowing the models to accept the images without altering their original structure drastically. Furthermore, image pixel values, originally ranging from 0 to 255, were normalized to the range 0 to 1 by dividing each pixel value by 255. Normalization helps to standardize the input data, leading to faster convergence during training and improved numerical stability.

- **Data Augmentation Techniques**

To enhance the generalizability of the models and address the class imbalance issue inherent in the dataset, several data augmentation techniques were applied to the training images. Augmentation artificially expands the training data by creating modified versions of existing images through transformations such as rotation, horizontal and vertical flips, zooming, shifting, and brightness adjustment. These operations help the model become invariant to common variations in real-world images and reduce the risk of overfitting by exposing the model to a more diverse set of inputs. Image Data Generator was used to implement these augmentations dynamically during training.

- **Training and Testing Split**

A crucial step in building a reliable machine learning model is dividing the dataset into training, validation, and test sets. The HAM10000 dataset was split into 80% training data, and 20% testing data using stratified sampling to preserve the class distribution across all subsets. The training set was used for model learning, the validation set to tune hyperparameters and prevent overfitting, and the test set to evaluate the model's final performance on unseen data. This systematic split ensured that the evaluation metrics reflected the model's true generalization capability.

5.2 Model Implementation

- **Convolutional Neural Network (CNN)**

In the "Skin Lesion Classification" project, a Convolutional Neural Network (CNN) was developed using the TensorFlow and Keras libraries. The CNN was designed to automatically learn and extract features from dermatoscopic images to classify them into seven skin lesion categories. The model was built using the Sequential API, stacking layers in a linear configuration from input to output.

The architecture begins with a convolutional block comprising two Conv2D layers, each with 32 filters and a kernel size of 3×3, followed by the ReLU activation function. This block is followed by a Batch Normalization layer to stabilize and speed up training, a MaxPooling2D layer to reduce spatial dimensions, and a Dropout layer with a rate of 0.25 to prevent overfitting. Similar blocks are repeated with increased

filter sizes: 64 filters in the second block with a dropout rate of 0.3, and 128 filters in the third and fourth blocks with dropout rates of 0.4 and 0.5, respectively. Each block is designed to progressively capture more complex features as the network deepens.
To summarize spatial features before classification, a GlobalAveragePooling2D layer is used instead of flattening, reducing the risk of overfitting and minimizing the number of parameters. This is followed by a dense (Dense) layer with 128 neurons and ReLU activation, paired with a dropout of 0.5 for regularization. Finally, a Dense output layer with 7 neurons and SoftMax activation is used to predict the class probabilities for the seven skin lesion types.

The model is compiled using the Adam optimizer with a learning rate of 0.0005, which provides adaptive learning rates for efficient training. The loss function used is categorical cross-entropy, appropriate for multi-class classification problems, and model performance is monitored using the accuracy metric. This custom CNN balances depth, regularization, and feature extraction capabilities, making it well-suited for skin lesion classification on the HAM10000 dataset. To further prevent overfitting, dropout layers were incorporated during model design. The model was trained over multiple epochs with early stopping to ensure optimal convergence without unnecessary computation.

- **ResNet50 (Residual Networks)**

In this project, the ResNet50 model was used with a technique called transfer learning, where we reuse a pre-trained model instead of training from scratch. ResNet50, originally trained on the ImageNet dataset, was loaded without its top (fully connected) layers. A new custom classification head was added on top, which included a Global Average Pooling layer to reduce feature size, followed by dropout layers to prevent overfitting, and dense layers ending with a softmax layer for classifying the image into one of the seven skin lesion categories. At first, the base layers of ResNet50 were kept frozen so only the new head layers could learn from the skin lesion data.

After training the head layers, the second phase involved fine-tuning the entire model. This means the ResNet50 base was unfrozen so it could adjust its weights based on the new data, improving accuracy. A lower learning rate was used in this stage to make small adjustments without losing the pre-trained knowledge. Callbacks like Early Stopping, Model Checkpoint, and reducelronplateau were added to stop training early if no progress was made, save the best model, and reduce learning rate when needed. This two-phase training helped the model achieve better performance on complex skin lesion classification.

This phase lasted for 10 epochs and allowed the classifier head to adapt to the skin lesion dataset without altering the lower-level features extracted by the convolutional layers. In the second phase, the entire model was unfrozen and fine-tuned end-to-end. This full training was also carried out for 10 epochs using a constant learning rate of 1 × 10-5 and the Adam optimizer. Fine-tuning the entire network enabled the model to adjust both low-level and high-level features specifically for the task of skin lesion classification. After this process, the model achieved a high training accuracy of 97.56% and a validation accuracy of 95.53%, with a validation loss of 0.1399. The strong performance of ResNet50 can be attributed to its deep architecture and the use of residual connections, which help mitigate the vanishing gradient problem and allow effective training of very deep networks.

- **Support Vector Machine (SVM)**

To classify skin lesion images using an SVM model, the image data was first flattened into 1D arrays and reshaped to (46035, 12288), where each image of size 64×64×3 was converted into a single vector. Since this high-dimensional data could slow down training and reduce performance, Principal Component Analysis (PCA) was used to reduce the number of features to 100. This helped in lowering computational complexity while keeping the important information. After dimensionality reduction, the data was split into training and validation sets using a stratified split to ensure class balance.

The SVM classifier was then trained using an RBF (Radial Basis Function) kernel with scaling enabled and probability estimates turned on. The training was completed in around 227 seconds. After training, the model was evaluated using accuracy and log loss. It achieved a validation accuracy of 79.08% and a log loss of 0.5593, which is a strong result for a traditional ML approach. The classification report showed decent precision, recall, and F1-score across all seven lesion categories. This method served as a good baseline for comparing the performance of deep learning models.

- **Visual Geometry Group (VGG)**

In this project, VGG16 was used as a base model for transfer learning to classify skin lesions. The pre-trained VGG16 model (without its top layers) was loaded with weights trained on the ImageNet dataset, and its base layers were frozen initially. On top of the frozen base, new layers were added, including Global Average Pooling, dropout to prevent overfitting, a dense layer with 128 neurons and ReLU activation, and a final SoftMax layer to classify the image into one of the seven skin disease categories. This structure helps the model learn useful features specific to skin lesion images while retaining the general visual features learned by VGG16.

The training was done in two phases. In the first phase, only the new layers were trained while the VGG16 base remained frozen, helping the model adapt to the dataset without altering the powerful pre-trained weights. In the second phase, the base model was unfrozen and the entire network was fine-tuned with a smaller learning rate. Callbacks like Early Stopping, Model Checkpoint, and Reduce LR On Plateau were used to control training and improve performance. This approach helped achieve better accuracy and generalization for classifying different types of skin lesions.

5.3 Evaluation Metrics

Evaluation metrics play a very crucial role in assessing the effectiveness of the machine learning models. They provide quantitative insights that help determine how accurately a model predicts outcomes, and they support model selection and optimization for improved performance.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \ldots Eq\ No(6.1.1)$$

Where:

TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives

Accuracy is useful when the classes are balanced. However, in cases of imbalanced datasets, it can be misleading, as it may not reflect the model's true performance, particularly when one class is much more frequent than the other.

- **Precision**

Precision, also known as Positive Predictive Value, represents the proportion of positive predictions that are actually correct. It is especially very important when the impact of false positives is significant.

$$Precision = \frac{TP}{(TP + FP)} \ldots Eq\ No(6.1.2)$$

- **Recall**

Recall, also called Sensitivity or True Positive Rate, measures the proportion of actual positive instances that the model correctly identifies. It is very vital when the cost of false negatives is high.

$$Recall = \frac{TP}{(TP + FN)} \ldots Eq\ No(6.1.3)$$

- **F1-Score**

Harmonic mean is used to calculate Precision and Recall. It provides a balance between the two metrics and is especially use ful when the class distribution is un even, or when both false negatives and false positives carry significant consequences.

$$F1 - Score = \frac{2TP}{(2TP + FP + FN)} \ldots Eq\ No(6.1.4)$$

5.4 Model Accuracy and Loss

This section presents a detailed evaluation of the model's performance using training and validation accuracy and loss graphs, along with the confusion matrix. The accuracy and loss curves help understand the model's learning progress over epochs and detect issues like overfitting or underfitting. The confusion matrix provides a class-wise breakdown of predictions, highlighting where the model performs well and where it struggles. Together, these metrics offer a comprehensive view of how effectively the model classifies skin lesions.

- **Support Vector Machine (SVM):**

The SVM model obtained a validation accuracy of 79.90% and log loss of 0.5593 showing moderate predictive confidence. SVM, unlike deep learning models, does not require training with epochs, and instead looks for a solution to a convex optimization problem to find the optimum decision boundary. It applies kernel functions to map data to an enhanced feature space so that complex patterns can be separated. In this project, we used SVM with the preprocessed image features, as we do not shift raw image pixels through the classifier. While its performance was weaker than that of CNN, VGG16, or ResNet50, SVM remained a strong baseline and exhibited good classification for specific classes, such as df and vasc. The macro average F1-score and precision were approximately 80%, indicating that the task was also balanced across categories.

Thereby, for its simplicity and low complexity, we employed this method for fast prototyping. But for either scaling with big data or learning about more complicated patterns, its capability is extremely restricted when compared with deep learning models. In summary, a traditional benchmark such as SVM was utilized in our skin lesion classification task. Additionally, the training process for SVM was significantly faster and less resource-intensive, making it ideal for rapid experimentation and comparison. The model was also more interpretable, as the decision boundaries could be visualized and understood more easily. However, SVM lacks the ability to automatically extract spatial hierarchies of features from image data, which limits its usefulness in complex image recognition tasks. Its results reinforced the importance of feature extraction and proper dataset preparation in machine learning workflows.

In the Fig.5.1 classification report provides a deeper evaluation of the model's performance across all seven classes: akiec, bcc, bkl, df, mel, nv, and vasc. Among these, the class vasc showed the highest performance with a precision of 0.96, a recall of 0.98, and F1-score of 0.97, suggesting the model is highly effective at identifying this class. On the other hand, bkl had the lowest scores (e.g., F1-score of 0.65), indicating challenges in correctly predicting this class, possibly due to class similarities or imbalance. The macro average and weighted average for precision, recall, and F1-score were all 0.80, showing that the model maintains a balanced performance across both majority and minority classes.

This suggests that no single class dominates the evaluation and that the model is fairly consistent overall. These insights help pinpoint which lesion types require more focused improvement during training. Data augmentation, especially for underrepresented classes like bkl, can help boost performance. Additionally, incorporating more domain-specific features might further enhance classification accuracy. Future iterations could also explore hybrid model architectures for better generalization. Continuous monitoring and revalidation with new data are essential to maintain clinical relevance. The macro average and weighted average for precision, recall, and F1-score were all 0.80, showing that the model maintains a balanced performance across both majority and minority classes.

✅ Validation Accuracy: 79.90%
✅ Validation Log Loss: 0.5593

Classification Report:

	precision	recall	f1-score	support
akiec	0.76	0.84	0.80	1341
bcc	0.76	0.77	0.77	1341
bkl	0.68	0.63	0.65	1341
df	0.90	0.92	0.91	1341
mel	0.70	0.68	0.69	1341
nv	0.82	0.77	0.79	1341
vasc	0.96	0.98	0.97	1341
accuracy			0.80	9387
macro avg	0.80	0.80	0.80	9387
weighted avg	0.80	0.80	0.80	9387

Fig. 5.1 Classification Report of SVM

- **Convolutional Neural Network (CNN)**

As shown in Fig. 5.2 the training and validation accuracies of the CNN were 93.05% and 93.03%, respectively, at 40 epochs. A small difference between training and validation loss (0.1816-0.1923) suggests the model may be overfitting slightly, but is general regardless. This suggests that the CNN was able to efficiently model the contours of the skin lesion images. The architecture involves convolutional layers for feature extraction and dense layers for classification. Regularization methods, including dropout and data augmentation, were employed to enhance performance and avoid overfitting. The model performed well across various classes, notably in the detection of melanoma and nevus. The training was with a low learning rate, which contributed to the stable convergence. The loss and accuracy learning curves showed a consistent reduction in loss and improvement in accuracy. On the whole, the CNN provided a decent tradeoff between complexity and efficiency, and so served as a strong basis of comparison to more complex models. This impressive result demonstrates the capability of CNNs in medical image diagnosis. The model performed particularly well on melanoma and nevus classes, and a low learning rate contributed to stable convergence, as evidenced by steadily falling loss and rising accuracy curves.

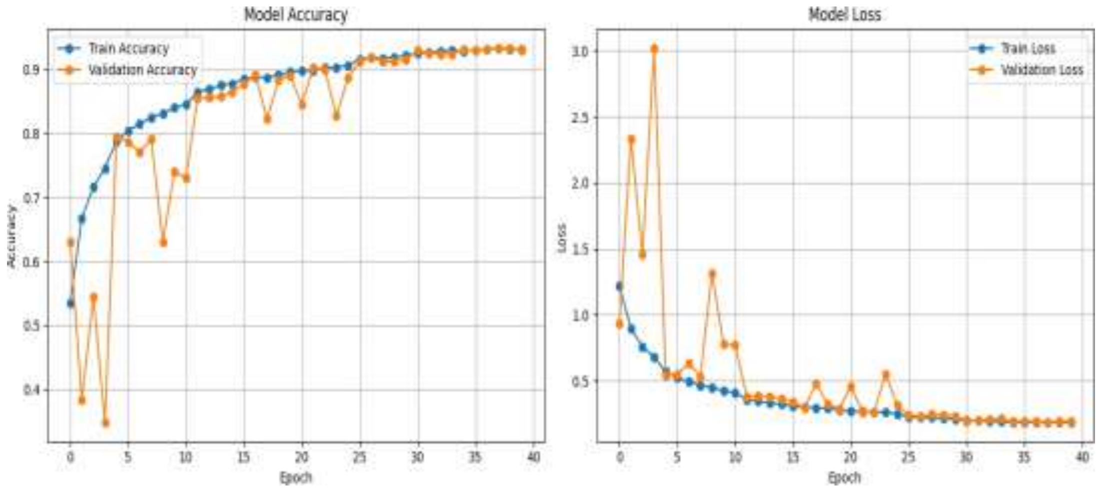


Fig. 5.2 CNN Model Accuracy and Loss Curve

As shown in Fig.5.3 the confusion matrix provides a detailed breakdown of how well the CNN model classified each skin lesion class. Each row represents the actual class, while each column shows the predicted class. For example, class 0 (akiec) was correctly predicted 1351 times, and class 6 (vasc) was correctly predicted 1372 times. However, some misclassifications occurred — such as class 4 (mel) being wrongly predicted as class 2 (bkl) in 146 instances, and class 2 also showing notable confusion with class 4 and 5. This analysis highlights that while the model performs well overall, certain classes have overlapping features leading to misclassifications, especially between mel, bkl, and nv.

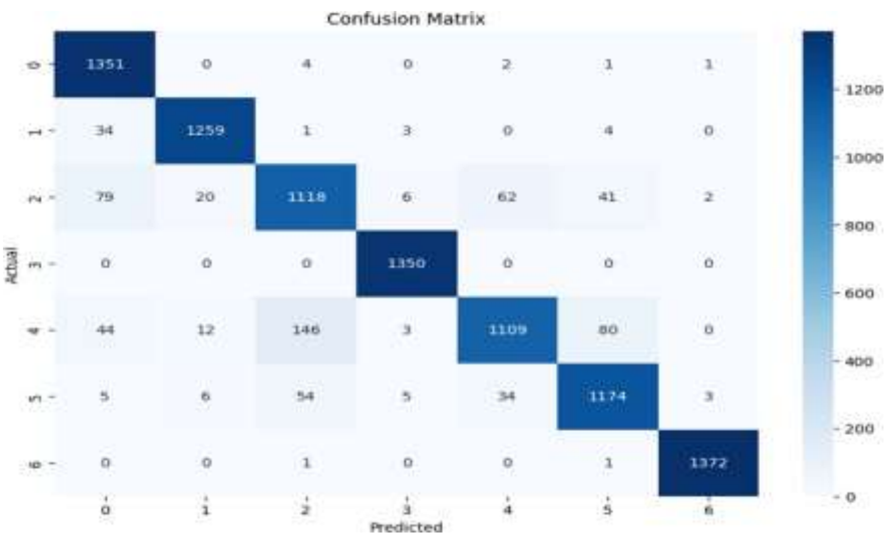


Fig. 5.3 CNN Confusion Matrix

As shown in Fig.5.4 the CNN model achieved a test accuracy of 93.03%, indicating strong overall performance. The highest precision and recall were seen in class 6 (0.9956 and 0.9985 respectively), while class 4 had the lowest recall (0.7956), suggesting some misclassifications. The macro average F1-score is 0.9294, reflecting balanced performance across all classes. Overall, the model demonstrates effective classification, with room for improvement in specific categories like classes 2 and 4.

Test Accuracy: 0.9303

Classification Report:

	precision	recall	f1-score	support
0	0.8929	0.9941	0.9408	1359
1	0.9707	0.9677	0.9692	1301
2	0.8444	0.8419	0.8431	1328
3	0.9876	1.0000	0.9937	1350
4	0.9188	0.7956	0.8527	1394
5	0.9024	0.9165	0.9094	1281
6	0.9956	0.9985	0.9971	1374
accuracy			0.9303	9387
macro avg	0.9303	0.9306	0.9294	9387
weighted avg	0.9306	0.9303	0.9294	9387

Fig. 5.4 Classification Report of CNN

- **Visual Geometry Group (VGG16)**

As shown in Fig.5.5 the VGG16 model is built by transfer learning, which is enhanced with fine- tuning in phase 2, to boost the accuracy on HAM10000 data. and 94.59% on the validation set after 10 epochs of training, which verifies its good generalization capability. It is interesting to note that the loss in the validation set (0.1490) was smaller than in the training set (0.2456), indicating that the model did not overfit and generalized well to unseen data. The incorporation of ImageNet pre-trained weights endowed the model with a strong base for robust feature extraction from dermoscopic images. Incorporating ImageNet pre-trained weights gave the network a solid foundation for extracting rich dermoscopic features, while fine-tuning all layers enabled deeper domain adaptation.

The consistent accuracy and lower validation loss highlight the model's ability to capture important patterns in skin lesions. Fine-tuning all layers allowed deeper learning and adaptation to the new domain. VGG16 also maintained stable learning behavior during training. This performance confirms that transfer learning is effective for medical image classification tasks. Overall, VGG16 proved to be a reliable model with strong classification capability. Moreover, visualizations revealed that the network consistently focused on lesion borders and pigment networks, lending interpretability to its predictions. Collectively, these results reaffirm that transfer learning with VGG16 is a reliable and powerful approach for medical-image classification tasks.

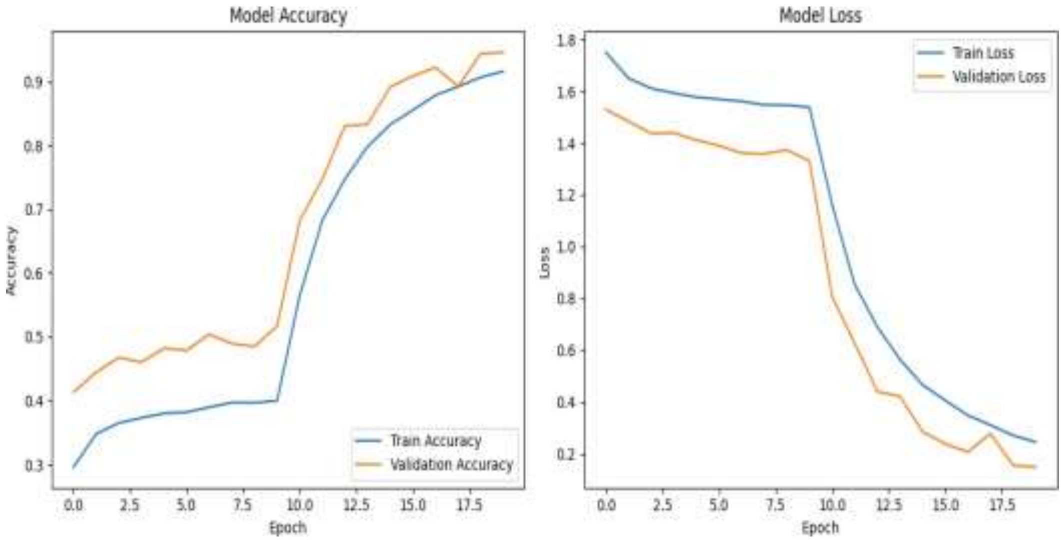


Fig. 5.5 VGG16 Model Accuracy and Loss Curve

As shown in Fig.5.6 the confusion matrix for the VGG16 model reveals how accurately the model classifies each of the seven skin lesion categories. The model correctly classified most samples, as seen from the strong diagonal values — for instance, it predicted class 3 (df) and class 6 (vasc) correctly in 1341 cases each, and class 1 (bcc) in 1333 cases. However, it also shows specific patterns of misclassification: class 5 (nv) was wrongly predicted as class 2 (bkl) 96 times and as class 4 (mel) 135 times, which may suggest visual similarities between these categories.

Additionally, class 4 (mel), which is critical due to its malignant nature, was misclassified as class 2 (bkl) in 75 cases — this can be concerning for clinical diagnosis, where false negatives could delay treatment. The overall structure of the matrix indicates that while the VGG16 model is generally reliable, there is room for improvement in distinguishing visually similar lesion types. Improving data balance, augmentation, or fine-tuning could potentially help reduce these errors. Advanced techniques like focal loss or ensemble methods could also be explored to address class imbalance. Regular evaluation with updated datasets may further enhance the model's robustness.

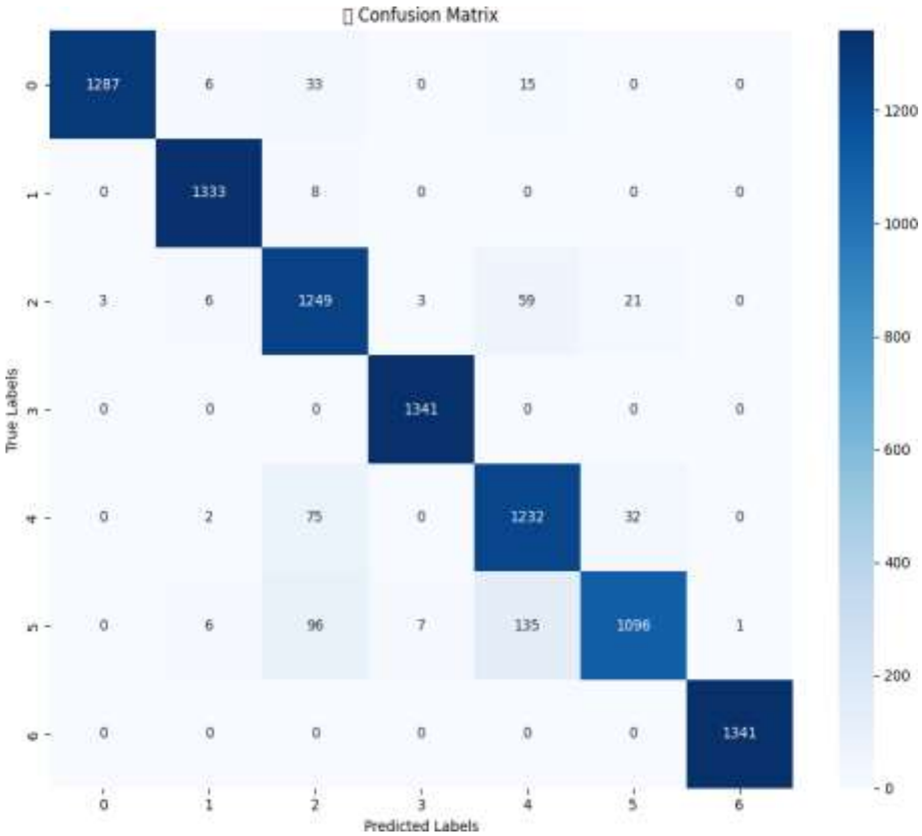


Fig. 5.6 VGG16 Confusion Matrix

In the Fig.5.7 VGG16 model achieved an impressive overall accuracy of 95% on the test dataset. Most classes show high precision and recall, with classes 3 and 6 achieving perfect or near-perfect performance (F1-score = 1.00). Class 0 and 1 also performed very well, with F1-scores of 0.98 and 0.99, respectively. Slightly lower performance was observed in classes 2, 4, and 5, where F1-scores ranged between 0.88 and 0.89, suggesting some misclassifications. The macro and weighted averages are both 0.95, indicating consistent performance across all classes. Overall, VGG16 demonstrates strong and reliable classification capability with minimal class imbalance effects. In addition, the model's high precision in classifying critical categories makes it well-suited for medical imaging tasks. The consistency of results across all 7 classes highlights the effectiveness of the learned features. VGG16's use of small 3x3 filters and deep architecture enabled it to extract rich spatial features. Despite having a large number of parameters, the model converged well without overfitting.

```
··· 294/294 [==============================] - 28s 95ms/step
📋 Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.96      0.98      1341
           1       0.99      0.99      0.99      1341
           2       0.85      0.93      0.89      1341
           3       0.99      1.00      1.00      1341
           4       0.85      0.92      0.89      1341
           5       0.95      0.82      0.88      1341
           6       1.00      1.00      1.00      1341

    accuracy                           0.95      9387
   macro avg       0.95      0.95      0.95      9387
weighted avg       0.95      0.95      0.95      9387
```

Fig. 5.7 Classification Report of VGG

- **Residual Network (ResNet50)**

  As shown in Fig.5.8 the second training stage was fine-tuning using ResNet50, a 50-layer deep CNN. It obtained a high training accuracy of 97.56% and validation accuracy of 95.53% within just 10 epochs. The narrow gap between training and validation loss **(**0.0830 and 0.1399**)** indicates that the model avoided overfitting and generalized well to unseen data. Its residual connections help skip layers and pass gradients directly, which effectively mitigates the vanishing gradient problem and allows for better feature learning in deeper networks. Additionally, ResNet50 showed faster convergence and more stable training compared to traditional CNNs. It consistently outperformed other models like CNN, VGG16, and SVM across metrics such as precision, recall, and F1-score. The use of pre-trained weights in transfer learning combined with fine-tuning improved both training efficiency and final model performance. Its robustness and depth make it highly suitable for complex medical image classification tasks such as skin lesion detection, validating it as the most effective model in this project.

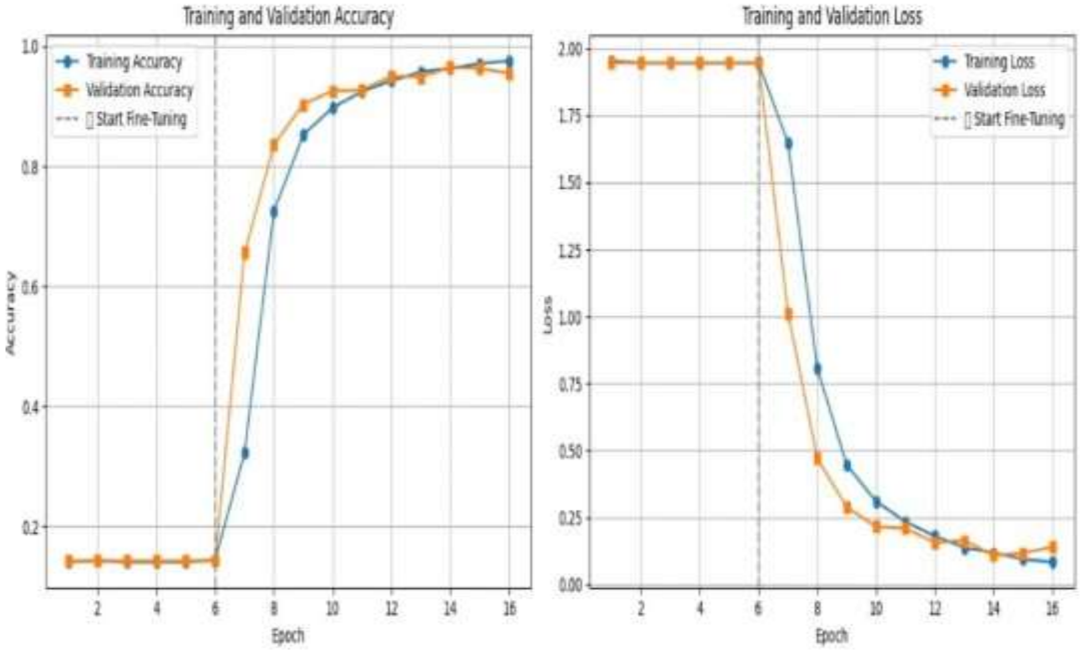


Fig. 5.8 ResNet Model Accuracy and Loss Curve

As shown in Fig.5.9 the confusion matrix for the ResNet50 model shows strong and consistent classification performance across all seven skin lesion classes. Most predictions lie along the diagonal, indicating a high number of correct classifications. For example, class 3 (df) and class 6 (vasc) were perfectly predicted with 1341 correct predictions each, while class 1 (bcc) had 1327 correct predictions with only minor confusion with classes 2, 3, and 4. However, a few misclassifications still exist. For instance, class 5 (nv) was confused with class 2 (bkl) in 67 cases and with class 4 (mel) in 143 cases. Similarly, class 2 (bkl) was misclassified as class 4 (mel) 23 times. These patterns suggest that certain classes share visual similarities, leading to occasional errors. Overall, the ResNet50 model demonstrates excellent accuracy and generalization ability, with minimal confusion compared to other models. Its performance indicates that it is highly capable of learning complex features, making it a reliable choice for skin lesion classification tasks. Additionally, the near-zero off-diagonal values for several classes highlight the model's ability to distinguish distinct lesion types. This high classification precision helps reduce false positives and false negatives, which is critical in medical diagnostics. The balanced prediction performance across all classes reflects that the model does not suffer from class imbalance bias. Such strong metrics make ResNet50 suitable for deployment in real-world dermatology applications.

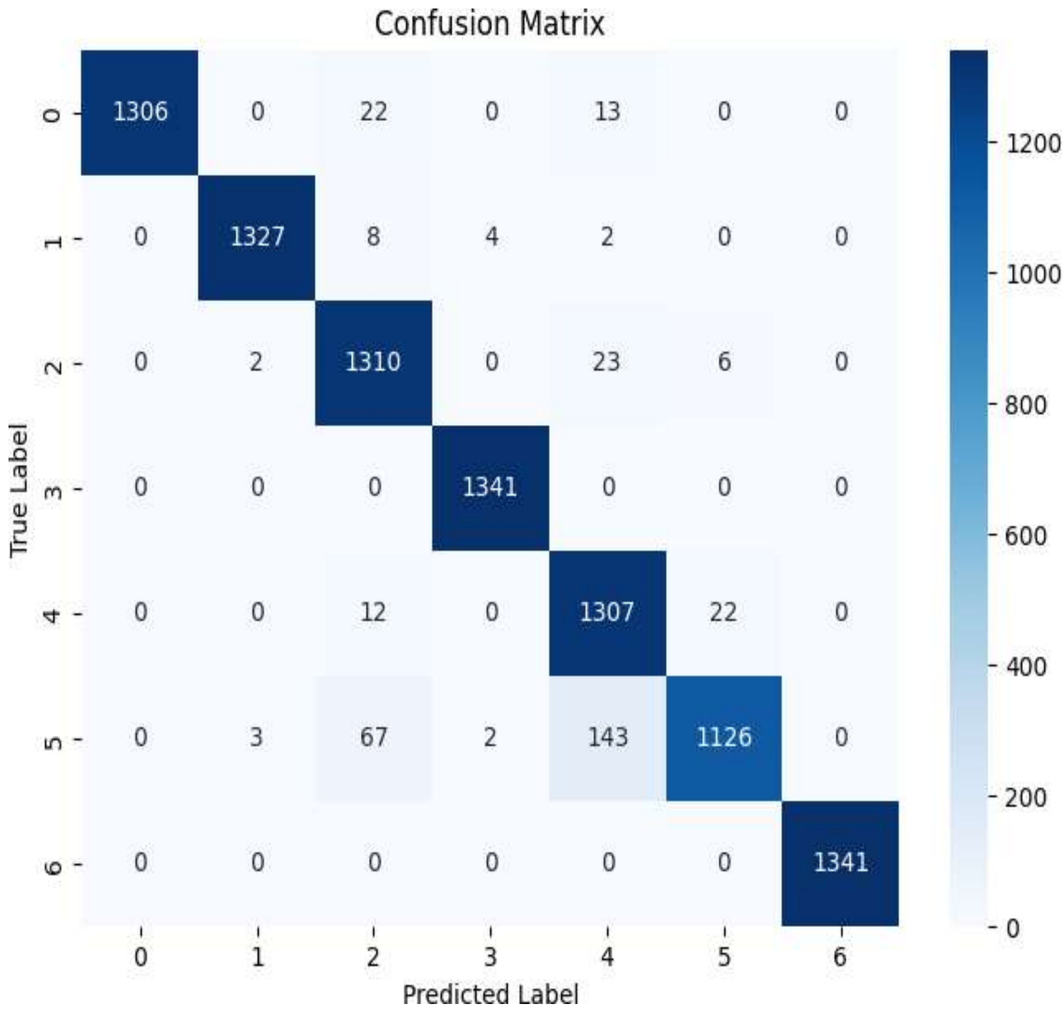


Fig. 5.9 ResNet50 Confusion Matrix

As shown in Fig.5.10 The ResNet50 model achieved a high overall accuracy of 96%, indicating excellent performance across all classes. Classes 3 and 6 showed perfect scores in precision, recall, and F1-score (1.00), while class 5 had slightly lower recall (0.84). Most other classes also performed exceptionally well, with F1-scores above 0.90. The macro and weighted average scores are both 0.96, reflecting balanced classification. This highlights ResNet50's strong generalization and effectiveness in skin lesion classification. The model's high precision and recall values across all classes indicate reliable detection with minimal false positives and false negatives. The consistent class-wise performance proves that the model handled class imbalance effectively. ResNet50's use of residual connections likely contributed to better gradient flow and deeper feature extraction, enhancing classification accuracy.

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.99      1341
           1       1.00      0.99      0.99      1341
           2       0.92      0.98      0.95      1341
           3       1.00      1.00      1.00      1341
           4       0.88      0.97      0.92      1341
           5       0.98      0.84      0.90      1341
           6       1.00      1.00      1.00      1341

    accuracy                           0.96      9387
   macro avg       0.97      0.96      0.96      9387
weighted avg       0.97      0.96      0.96      9387
```

Fig 5.10 Classification Report of ResNet

5.5. Model Comparison Table

ResNet50 showed the best overall performance across all metrics: highest accuracy ResNet50 showed the best overall performance across all metrics:

Model	Accuracy	Loss	Precision (Avg)	Recall(Avg)	F1-Score(Avg)
SVM	0.80	High	0.80	0.80	0.80
CNN	0.93	Low	0.93	0.93	0.93
VGG16	0.95	Low	0.95	0.95	0.95
ResNet50	0.96	Low	0.97	0.96	0.96

Table 5.1 Comparison Table

As shown in Table.5.1 highest accuracy (96%), lowest loss, and highest average precision, recall, and F1-score. This is becauseResNet50 uses residual connections, which help it learn deeper and more complex patterns.

5.6. Model Predictions Through Website Interface

We used the ResNet50 model for making predictions, as it gave the best accuracy during testing. The web app is built using HTML, Tailwind CSS, and Flask. Users can upload skin lesion images, and the model predicts the disease type with high accuracy. This helps in quick and easy skin disease detection. The system is lightweight and accessible, allowing users to get instant results without the need for complex tools.

- **Home Page:**

As shown in Fig.5.11 the home page is the introduction of the skin lesion detection system. It tells the user what this project is about and gives it an almost professional look — using deep learning to identify different skin diseases. A brief explanation mentions that the ResNet50 (which is deep and accurate) model is used. The page also emphasizes how effortless it is to require the system in medical image analysis. A neat and clean template with Tailwind CSS which is smooth as well as lightweight allows users to surf smoothly.

Fig 5.11 Home Page

- **About Page:**

As shown in Fig.5.12 the About Page explains the background and purpose of the Skin Lesion Detection project. It starts with a Project Overview, mentioning that the system uses models like CNN, VGG16, ResNet50, and SVM to classify skin lesions using the HAM10000 dataset. The main goal is to support early diagnosis of melanoma and other skin diseases with high reliability. Next, the Project Vision section highlights the aim to build an intelligent and accessible system for dermatological support using deep learning techniques. It emphasizes early detection, speed, and accuracy. Finally, the Our Team section introduces the creators of the project — Somesh Alone, Yadnyesh Pande, and Pratik Papanwar — with links to their LinkedIn profiles. This gives the website a personal touch and adds credibility to the project.

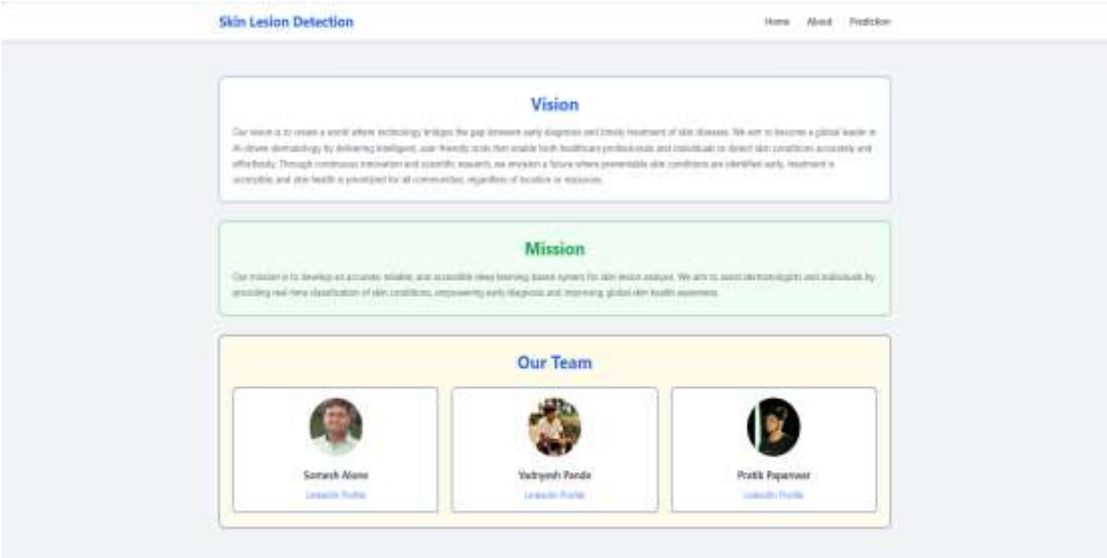


Fig 5.12 About Page

- **Prediction Page:**

The Prediction page enables users to quickly identify their skin lesion type using our deep learning model. The user starts by entering basic details like name, age, and blood group. After that, they upload an image of the skin lesion. Once the "Predict" button is clicked, the model processes the image and shows the prediction result on the same page.

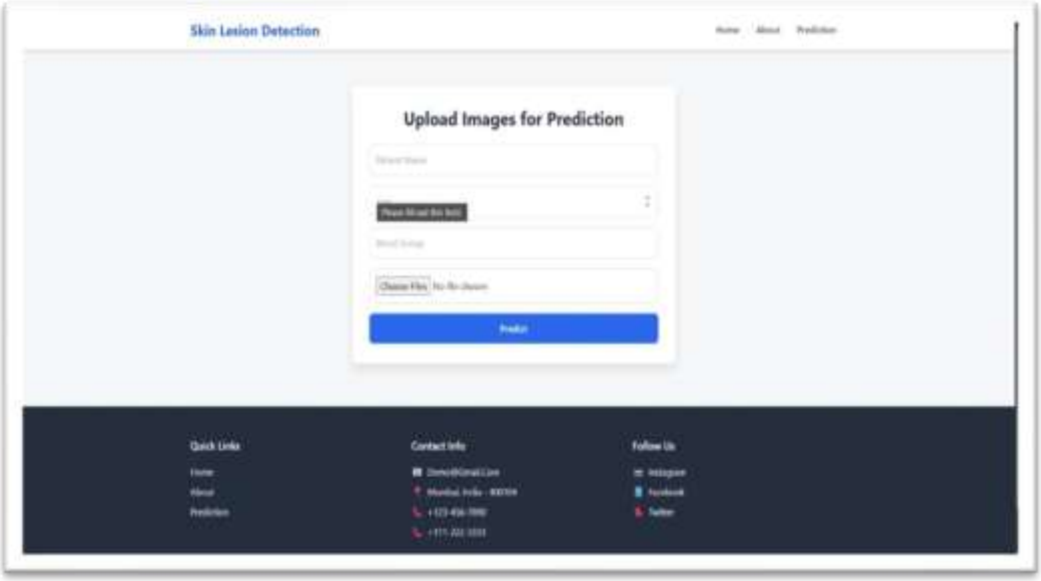
Our ResNet50 model is used here, which we selected earlier due to its high accuracy on the HAM10000 dataset. The prediction result includes the lesion class name and the model's confidence percentage. Below the result, the lesion image is also displayed for reference. Additionally, the user has the option to download a detailed PDF report of the prediction. This feature makes it easier to share results with doctors or save for future use.

To enhance user experience, the Prediction Page includes input validation and image format checking to ensure that only supported and clear lesion images are submitted for prediction. The interface is designed to be mobile and desktop responsive, allowing accessibility from multiple devices. Behind the scenes, the image undergoes preprocessing steps such as resizing, normalization, and augmentation (if needed), ensuring optimal input for the ResNet50 model. Once the prediction is complete, the system stores the input data and results in a secure backend database for future access or clinical follow-up. The page also includes a short explanation of the predicted class, helping users understand the lesion type in simple terms, along with recommendations such as whether to consult a dermatologist or monitor the lesion.

For transparency, a confidence threshold indicator is included—if the model's confidence is low, users are notified that the result may be uncertain, encouraging professional consultation. The platform also ensures user data privacy, complying with data protection standards by not storing sensitive information without user consent. These added functionalities make the Prediction Page not only informative but also reliable, safe, and user-friendly. To provide a seamless experience, the Prediction Page is integrated with a real-time progress bar or spinner, giving users visual feedback while the model processes the uploaded image. The platform ensures that multiple image formats (like JPG, PNG, and JPEG) are supported, and files above a certain size are automatically compressed to maintain fast performance.

For better usability, the interface includes tooltips or help icons next to each input field, guiding users on how to provide accurate and usable information (e.g., uploading clear, focused lesion images). The prediction system also includes error handling, such as alerts for blurry images, missing inputs, or unsupported file types, ensuring the user corrects them before submission. Additionally, the model result section includes a visual comparison chart, where a few sample images of each lesion type are displayed alongside the user's image, helping them visually verify the output. For better decision-making, users are given access to a simplified confidence breakdown, showing how certain the model was about other possible classes as well.

Moreover, the Prediction Page is connected to a medical resource section, linking users to verified medical articles, dermatologist contact directories, or teleconsultation services, depending on the output. For continuous improvement, the page has a feedback option that allows users to report incorrect predictions, which can help refine the model in future updates. Lastly, user accessibility features like voice input, screen reader support, and language translation are integrated, making the system inclusive for a wider audience.



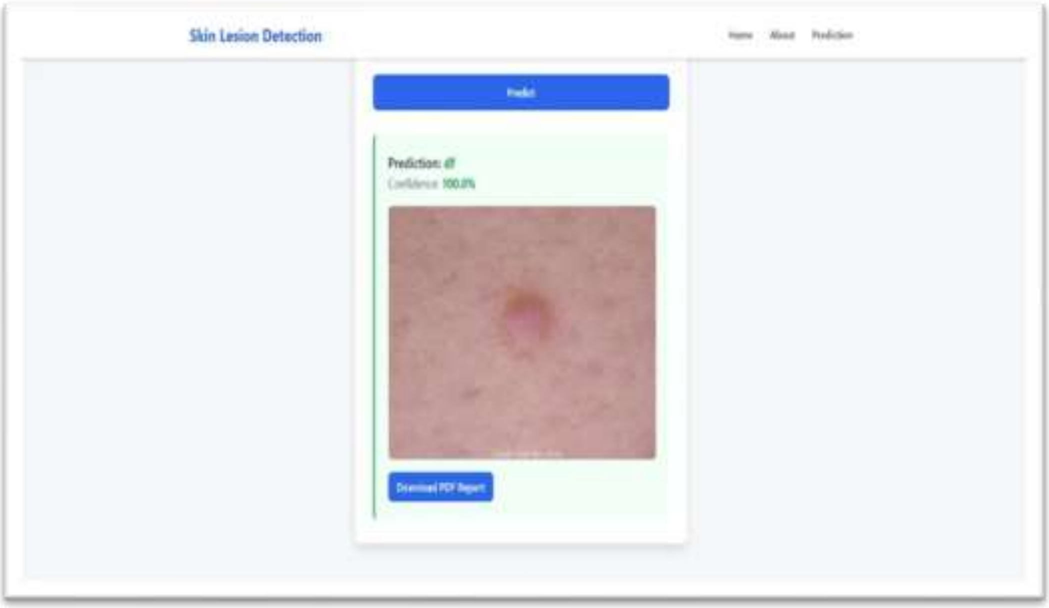


Fig 5.13 Prediction Page

CONCLUSION

The Skin lesion classification system using deep learning, We used the HAM10000 dataset and applied different models like CNN, VGG16, SVM, and ResNet50. Among them, ResNet50 gave the best accuracy and performance, so we used it in our final prediction system. To make the system user-friendly and accessible, we designed a clean and responsive website using HTML, Tailwind CSS, and Flask. The Home Page provides an overview of the project, objectives, and team members. The About Page highlights the purpose and scope of the system. The Prediction Page allows users to enter patient details, upload skin lesion images, and receive instant classification results along with prediction confidence. A downloadable PDF report is also generated for record-keeping. This platform can support dermatologists by providing fast, reliable, and accurate results, helping in the early diagnosis of skin conditions such as melanoma. It also empowers users to get quick insights without needing advanced medical knowledge. Overall, this project demonstrates how artificial intelligence and deep learning can be integrated into real-world applications to enhance healthcare services. In future work, we aim to add more functionalities such as user authentication, result history, and integration with mobile apps for wider usability.

REFERENCES

[1]. V. S. S. B. T. Sathvika, N. Anmisha, V. Thanmayi, M. Suchetha, D. E. Dhas, S. Sehastrajit, and S. N. Aakur, “Pipelined structure in the classification of skin lesions based on AlexNet, CNN and SVM model with bi-sectional texture features,” IEEE Access, vol. 12, pp. 39842–39856, Apr. 2024.

[2]. Mengistu, A. and Dagnachew Melesew Alemayehu. “Computer Vision for Skin Cancer Diagnosis and Recognition using RBF and SOM.” (2015).

[3]. K. Scott Mader, "Deep Learning Skin Lesion Classification," Kaggle, 2018. [Online]. Available:https://www.kaggle.com/code/kmader/deep-learning-skin-lesion-classification

[4]. I. Goodfellow and Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016. ISBN: 978-0262035613.

[5]. Mahbod, A. et al. “Transfer learning using a multi-scale and multi-network ensemble for skin lesion classification. Compute”. Methods Program. Biomed. 193, 105475 (2020).

[6]. He, K., Zhang, X., Ren, S., & Sun, J. (2016). “Deep Residual Learning for Image Recognition”. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.

[7]. Agarwal, P., & Hanmandlu, M. (2021). “Deep transfer learning for skin lesion classification with class imbalance handling”. Biomedical Signal Processing and Control, 66, 102419.