# CHAPTER 1

# INTRODUCTION

### OBJECTIVE

The primary objective of this project is to design and implement an automated skin cancer detection and classification system using deep learning methodologies. Skin cancer is one of the most common and potentially life-threatening forms of cancer globally, and its early detection significantly improves the chances of successful treatment. Therefore, the proposed system aims to leverage advanced computational techniques to assist in early diagnosis, thereby reducing the dependency on dermatologists for manual interpretation and minimizing human error in diagnosis[1].

A key goal of this study is to utilize Convolutional Neural Networks (CNNs) for image classification. CNNs are well-suited for handling complex image data and can automatically learn and extract significant features from dermoscopic images[4]. By employing a CNN-based model, the system is trained to distinguish between benign and malignant skin lesions, making it a reliable diagnostic aid. To enhance the model’s ability to classify accurately, a pre-trained AlexNet model is incorporated using the transfer learning approach, allowing for better feature representation and reducing the training time required for convergence.

Another major objective is to implement image preprocessing techniques to improve the input quality of skin lesion images[3]. These preprocessing steps include hair removal, glare elimination, and shadow correction, all of which help in removing artifacts that may hinder accurate classification. By improving the clarity and uniformity of images, the system becomes more robust and reliable in handling diverse real-world dermoscopic data.Furthermore, the study aims to tackle the common problem of class imbalance within medical datasets through data augmentation techniques. Since some classes in the skin lesion dataset may have fewer images than others, augmentation methods such as cropping, flipping, and rotation are applied to artificially increase the diversity and quantity of training data. This ensures that the model does not become biased toward any particular class and is capable of generalizing better to unseen data.Finally, the performance of the proposed model is evaluated using well-defined metrics, including accuracy, precision, recall, specificity, and F1-score.

These metrics provide a comprehensive understanding of the system's classification ability and help in benchmarking its effectiveness against existing methodologies. The ultimate goal is to build a Computer-Aided Diagnosis (CAD) system that is not only accurate but also scalable and ready for deployment in clinical environments, thereby aiding dermatologists and healthcare providers in the early detection and monitoring of skin cancer.

### PROBLEM STATEMENT

Skin cancer is one of the most prevalent forms of cancer globally, and its incidence rate has been increasing steadily over the past few decades. Despite advancements in medical technologies, early detection of skin cancer remains a significant challenge due to the complexity and variability in lesion appearance across different patients and skin types. Traditional diagnostic methods largely depend on the visual assessment skills of dermatologists, which can be subjective, time- consuming, and prone to human error.

Furthermore, in rural and under-resourced regions, access to skilled dermatologists and diagnostic facilities is limited, often resulting in delayed diagnoses and poor patient outcomes.Another major concern is the similarity in visual features between benign and malignant lesions, which makes accurate classification difficult even for experienced practitioners. The increasing complexity and variability of skin cancer lesions across diverse patient populations call for an objective, automated solution that can deliver consistent, accurate, and early diagnoses to support medical professionals. Manual diagnosis by dermatologists, though highly skilled, is inherently subject to variability due to human factors such as experience level, fatigue, and subjective interpretation. Given the life-threatening nature of malignant lesions like melanoma, and the subtle differences between benign and malignant skin features, reliance solely on manual assessment risks delayed or inaccurate diagnoses. The advent of deep learning, especially the remarkable success of Convolutional Neural Networks (CNNs) in visual pattern recognition tasks, has opened promising new avenues in medical image analysis. CNNs can learn complex features directly from pixel data without manual feature engineering, making them particularly suited to identifying the fine-grained patterns and textures present in dermoscopic skin images[6].

Despite this progress, developing reliable AI-driven models for skin cancer classification remains a challenging task. One major obstacle is the limited availability of large, annotated datasets. Annotating medical images requires significant time and expertise from trained dermatologists, leading to datasets that are often smaller and less diverse compared to datasets in general computer vision tasks. Additionally, dermoscopic images frequently suffer from artifacts such as hair occlusions, lighting glare, shading, and low contrast, which can degrade image quality and obscure critical lesion features. Without careful preprocessing, these artifacts can significantly impair model learning and performance. Another notable challenge is class imbalance, where common skin conditions are overrepresented in datasets while rare but deadly cancers, such as amelanotic melanoma, have very few examples. This imbalance can cause models to become biased towards more frequent classes, leading to poor detection rates for rarer but clinically significant conditions.

Therefore, there is an urgent and pressing need to design and develop an efficient, scalable, and fully automated skin cancer detection system that addresses these challenges systematically. Such a system must employ advanced deep learning strategies—such as transfer learning from large-scale pre-trained networks, data augmentation to synthetically balance class distributions, and fine-tuned CNN architectures optimized specifically for medical images.

Moreover, a robust preprocessing pipeline must be established to enhance image quality by removing noise, normalizing contrast, correcting illumination inconsistencies, and eliminating obstructive artifacts like hairs and bubbles. Together, these improvements would allow the AI system to focus purely on relevant lesion features, thereby improving the model’s sensitivity and specificity.

Ultimately, the goal is to create a tool that not only matches but potentially exceeds the diagnostic performance of human experts when applied at scale. Such a solution would enable earlier detection of skin cancers, improve patient outcomes through timely interventions, reduce diagnostic burdens on dermatologists, and make quality dermatologic care accessible even in resource-limited settings.

By leveraging the full potential of deep learning and meticulously addressing real-world data challenges, this vision of a highly accurate, automated skin cancer classification system can be realized, fundamentally transforming the future of dermatological diagnostics.

### CHAPTER-WISE SUMMARY

#### Chapter 1: Introduction

Chapter 1 lays the foundation for the project by introducing the significance of skin cancer detection and the role of deep learning in medical diagnostics. It presents the background of the study, problem statement, objectives, and the motivation behind automating the detection process. The chapter also explains the scope and limitations of the system, highlighting the urgent need for Computer-Aided Diagnosis (CAD) systems to support medical professionals in the early identification of malignant skin lesions.

#### Chapter 2: System Analysis

This chapter delves into the analysis of the existing systems and identifies the key limitations in current diagnostic approaches. It reviews related works and research studies to establish the relevance of deep learning-based solutions in dermatology. The system requirements—both functional and non- functional—are outlined here, along with a detailed problem analysis, feasibility study (technical, operational, and economic), and risk assessment. The analysis justifies the selection of Convolutional Neural Networks (CNNs) and transfer learning for this project.

#### Chapter 3: System Design

Chapter 3 focuses on the architecture and design of the proposed system. It includes block diagrams, flowcharts, and data flow diagrams that outline how data is processed through various stages— preprocessing, augmentation, feature extraction, classification, and result interpretation. The design of the deep learning model, specifically the adaptation of the AlexNet architecture, is explained in terms of layers, parameters, and data flow. The system also integrates preprocessing techniques like hair removal and glare correction to improve image quality before feeding into the network.

#### Chapter 4: System Implementation

This chapter provides a detailed explanation of the implementation process, tools, and technologies used. It describes how the dataset (from ISIC) is collected, processed, and augmented to address class imbalance. The training process of the CNN model, application of transfer learning, and optimization techniques are discussed in-depth. Evaluation metrics such as accuracy, precision, recall, and F1-score are used to assess the model’s performance.

The results of experiments with and without data augmentation are presented through confusion matrices and graphical visualizations, demonstrating significant improvement in classification performance.

#### Chapter 5: Conclusion and Future Scope

The final chapter summarizes the outcomes of the study and reiterates the effectiveness of the proposed deep learning model for skin cancer detection. It reflects on the key findings, such as the increased accuracy achieved through transfer learning and data augmentation. The limitations encountered during the research are acknowledged, and possible improvements are discussed. The chapter concludes with future scope, suggesting the integration of more advanced architectures like InceptionV3 or Vision Transformers (ViT), multi-class classification for multiple skin diseases, and the development of a real- time mobile or web-based diagnostic tool for broader clinical use.

### LITERATURE SURVEY

The evolution of artificial intelligence, particularly deep learning, has brought about transformative changes in the field of medical diagnostics. Skin cancer, due to its high global prevalence and mortality rates when detected late, has become a prime area of research for the development of automated diagnostic systems[1]. Conventional diagnostic procedures rely predominantly on dermatological expertise and dermoscopic image analysis, which can be subjective and vary between specialists[2]. These traditional approaches are also time-consuming and often inaccessible in underdeveloped or remote regions. As a result, researchers have turned to deep learning-based methods to improve the accuracy, consistency, and availability of diagnostic tools for skin cancer detection and classification.

One of the landmark studies in this field was conducted by Esteva et al., where they used a deep Convolutional Neural Network (CNN) model based on the InceptionV3 architecture, trained on over 130,000 clinical images[3]. Their work demonstrated that the model achieved classification performance comparable to that of experienced dermatologists, setting a new benchmark for artificial intelligence in dermatology. Another significant contribution was made by Kawahara et al., who employed a fully convolutional AlexNet model to detect and segment skin lesions, establishing the foundation for end-to-end deep learning approaches in this domain[1][2]. These studies highlighted the strength of CNNs in automatically learning complex visual patterns and textures present in skin lesions.

Further developments in the field have explored more sophisticated architectures and ensemble methods. For example, Ge et al. proposed a hybrid model combining high-level feature extraction from ResNet and VGG networks, which were then fused bilinearly and classified using Support Vector Machines (SVMs)[3][4]. This fusion strategy allowed for better generalization and accuracy across multiple types of skin lesions. Meanwhile, Yu et al. presented a tiered model based on deep residual networks (ResNets) as part of the ISBI 2016 challenge, which focused on hierarchical classification and enhanced lesion detection performance through residual learning. These innovations underscored the importance of utilizing deeper networks and feature fusion techniques to tackle the complexity of skin cancer classification[7].

Another important challenge addressed in the literature is the scarcity of annotated data and class imbalance in skin cancer datasets. Since medical image datasets are often limited and skewed toward certain classes (e.g., benign lesions being more prevalent than malignant), many researchers have applied data augmentation techniques to synthetically expand the dataset. Techniques such as flipping, rotation, scaling, cropping, and color jittering have been employed to increase diversity and reduce overfitting. Moreover, transfer learning has emerged as a powerful strategy, where models pre-trained on large-scale image datasets like ImageNet are fine-tuned on medical images[4]. This allows models such as AlexNet, VGG16, ResNet50, and DenseNet121 to leverage learned visual representations and adapt them to the domain of skin lesion analysis, even with relatively small medical datasets.

In recent developments, Vision Transformers (ViTs) and attention-based models have also been explored to capture global contextual information in images, which CNNs might overlook due to their local receptive fields[5]. These architectures segment input images into patches and apply transformer layers to understand inter-patch relationships, proving especially useful in identifying irregular patterns or asymmetries in skin lesions. Ensemble learning methods like stacking, which combine predictions from multiple base models (e.g., ViT, ResNet50, DenseNet121), have further enhanced prediction accuracy and stability by balancing the strengths and weaknesses of each model.

In summary, the literature emphasizes a shift toward automated, deep learning-based approaches for skin cancer detection due to their high accuracy, objectivity, and scalability. While earlier efforts focused on traditional machine learning with handcrafted features, modern systems rely on CNNs and hybrid architectures for end-to-end learning from dermoscopic images,CNNs and hybrid architectures for end-to-end learning from dermoscopic images.

Despite the challenges of data imbalance, variability in lesion appearance, and limited annotations, the application of transfer learning, data augmentation, and ensemble methods has significantly improved the performance of these systems. This project builds upon the foundation laid by previous studies and aims to enhance skin cancer detection further by integrating advanced deep learning models, optimized preprocessing, and thorough performance evaluation on real-world datasets.

# CHAPTER 2

# SYSTEM ANALYSIS

### EXISTING SYSTEM

The existing systems for skin cancer detection and classification predominantly rely on traditional diagnostic methods supported by human expertise. Over time, some computational tools have been introduced to aid diagnosis, but these systems still face several limitations in terms of accuracy, accessibility, and efficiency[2]. Below are the key aspects of existing systems used in clinical and research environments:

#### 2.1.1 Manual Diagnostic Techniques

In the conventional approach, dermatologists perform a visual examination of the skin lesion using dermoscopy or a magnifying tool to assess its features[1]. The diagnosis is primarily based on criteria such as the ABCD rule (Asymmetry, Border, Color, Diameter), 7-point checklist, or Menzies method[8]. These assessments are highly subjective and vary based on the clinician's experience. Additionally, biopsies are performed for further confirmation, which are invasive, time- consuming, and often not feasible for large-scale screening.

#### 2.1.2 Computer-Aided Diagnosis (CAD) Tools

Some existing CAD systems have been developed that utilize image processing and machine learning techniques to support dermatologists. These tools extract handcrafted features such as color histograms, texture descriptors, and shape parameters, and classify lesions using traditional algorithms like Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Random Forests. However, these methods heavily depend on manually selected features, which may not be optimal for capturing the complexity and variation of skin lesions.

#### 2.1.3 Limitations of Existing Systems

#### Despite their usefulness, existing systems come with several limitations. Human-based evaluation is prone to subjectivity and inconsistency, particularly in early-stage lesions where visual differences are minimal, leading to unreliable diagnoses. Traditional machine learning methods heavily rely on manual feature extraction, a process that is both time-consuming and often inadequate for handling the complexity of skin lesion classifications. Additionally, many datasets used in traditional systems suffer from data imbalance, with a predominance of benign cases, which causes poor model generalization when detecting malignant lesions. Another significant challenge is the presence of image artifacts in dermoscopic images, such as hair, glare, shadows, and uneven lighting, all of which can severely impact detection accuracy. Furthermore, the limited accessibility of advanced diagnostic equipment and trained medical professionals in rural or low-resource settings hampers timely and effective diagnosis, making it crucial to develop more accessible and automated solutions.

#### 2.1.4 Motivation for Deep Learning Solutions

With the advent of deep learning, particularly Convolutional Neural Networks (CNNs), there has been a paradigm shift in how skin cancer detection can be approached. Deep learning models automatically learn hierarchical features from raw images, eliminating the need for manual feature engineering. Additionally, with access to large public datasets such as ISIC (International Skin Imaging Collaboration), it has become feasible to train models that can match or even surpass human-level performance.

However, even deep learning-based existing systems face challenges like overfitting due to small datasets, imbalanced classes, and lack of standardized evaluation metrics. These issues highlight the need for a robust and efficient system that leverages deep learning techniques while incorporating preprocessing, augmentation, and model optimization strategies to achieve higher accuracy and broader applicability.

### PROPOSED SYSTEM

The proposed system aims to provide a reliable, automated, and highly accurate solution for the early detection and classification of skin cancer using deep learning techniques. Leveraging the power of Convolutional Neural Networks (CNNs), transfer learning, and advanced image preprocessing, this system is designed to overcome the shortcomings of traditional diagnostic methods and existing computer-aided tools. It processes dermoscopic images to distinguish between benign and malignant skin lesions, making it a valuable tool for clinical settings, especially where expert dermatologists are not available.

## 2.2.1 Key Features and Capabilities of the Proposed System

The core of the proposed system is built around Convolutional Neural Networks (CNNs), which are capable of automatically extracting relevant features from dermoscopic images. Unlike traditional machine learning models that require manual feature engineering, CNNs learn spatial hierarchies of patterns through layers, capturing both low-level features such as edges and textures, and high-level features like shapes and contours, all of which are crucial for accurate skin lesion classification.To address the issue of limited labeled medical image data, the system integrates transfer learning by utilizing a pre-trained AlexNet model. This approach enables the reuse of learned features from large datasets like ImageNet, significantly enhancing performance and reducing the required training time. The final layers of AlexNet are fine-tuned with the skin lesion dataset to adapt the model specifically for binary classification tasks, distinguishing between benign and malignant lesions.

Advanced image preprocessing techniques are applied to improve the quality of dermoscopic images and minimize classification errors caused by visual artifacts. These preprocessing steps include hair removal to eliminate hair strands that may obscure lesion regions, glare reduction to minimize light reflections, and shading and illumination correction to normalize lighting conditions, thereby enhancing contrast. These techniques ensure that the input images maintain consistent and clear lesion features for better analysis.The system also addresses the common issue of class imbalance in medical datasets through data augmentation strategies. Techniques such as image rotation, flipping, scaling, cropping, and color transformations are employed to artificially expand and diversify the dataset. This augmentation not only increases dataset size but also improves the model's generalization ability and reduces overfitting, particularly benefiting the underrepresented malignant class.

High classification accuracy and robustness are key strengths of the proposed system. The CNN model demonstrates impressive performance across metrics such as accuracy, precision, recall, and F1-score, especially when trained on the augmented dataset. It shows strong capabilities in minimizing both false negatives, where malignant cases might be missed, and false positives, where benign lesions are incorrectly classified as malignant, making it highly reliable for early diagnosis and screening applications.Scalability and real-time diagnostic potential are also integral to the system's design. It is lightweight and optimized for deployment across various platforms, including web-based diagnostic tools, mobile applications, and integrated clinical software systems. This flexibility ensures that the system can be effectively utilized not only in urban hospitals but also in rural clinics where advanced diagnostic resources may be limited.For thorough model evaluation, a detailed performance analysis is carried out using a confusion matrix, providing insights into true positives, false positives, true negatives, and false negatives. The system's performance is rigorously assessed using key metrics such as accuracy, precision, recall, specificity, and F1-score, ensuring its readiness for real-world clinical applications.

Currently, the model supports binary classification, efficiently determining whether a skin lesion is benign or malignant. This capability is crucial for prioritizing patients for further biopsy or treatment. Looking forward, the system can be extended to support multi-class classification to differentiate among various types of skin cancers, such as melanoma, basal cell carcinoma, and squamous cell carcinoma, further broadening its diagnostic utility.

## USE CASE ANALYSIS

Use case analysis is a crucial part of system analysis, as it defines how different users will interact with the proposed system. It provides a clear picture of the system’s functionalities from the end- user’s perspective. For the automated skin cancer detection system, use cases are designed to identify how the system is expected to behave in real-world clinical or diagnostic scenarios. These interactions involve various stakeholders such as dermatologists, medical assistants, researchers, and even patients using mobile or web-based interfaces.

The proposed system leverages deep learning and image processing to analyze dermoscopic images and classify them as benign or malignant. The use cases are primarily driven by the goal of improving early skin cancer detection through automation, accuracy, and accessibility.

#### Actors Involved in the System

The primary user of the system is the dermatologist or clinician, who utilizes the platform to assist in diagnosing skin lesions. Secondary users, such as patients or general users, may interact with the system through a user-friendly interface to upload images for preliminary analysis. A system administrator is responsible for managing backend operations, maintaining datasets, and updating the machine learning model as needed. At the core of the system is the machine learning model itself, which processes and classifies input images to support accurate and efficient diagnosis.

#### The Key Use Cases for the System Include:

The system's workflow begins with users, either dermatologists or patients, uploading dermoscopic images through a web or mobile interface. The uploaded images must meet certain resolution standards to ensure accurate processing. Once uploaded, the system automatically performs image preprocessing, applying techniques such as glare removal, hair removal, and normalization of contrast and brightness to enhance image quality. The preprocessed images are then analyzed by a Convolutional Neural Network (CNN) model, which uses transfer learning based on AlexNet to extract features and classify the lesion as benign or malignant. The classification results, accompanied by confidence scores and visual indicators, are presented to the user in an intuitive format, helping them make informed decisions about possible next steps, such as consulting a specialist or undergoing a biopsy. Additionally, users, particularly dermatologists, can save and export diagnostic reports in formats like PDF or store them in cloud servers to support clinical continuity and research documentation.

On the backend, system administrators play a critical role in maintaining and enhancing the system's effectiveness. They periodically retrain the machine learning model with updated or expanded datasets to ensure improved accuracy and generalization over time. Furthermore, the system incorporates error handling and a feedback loop, addressing cases where image uploads fail or where the model’s results are inconclusive. Users are prompted to re-upload clearer images or submit feedback, which not only enhances the system's robustness but also contributes valuable data for continuous model improvement. Through these processes, the system maintains its reliability, scalability, and relevance for real-world clinical applications.

#### Use Case Diagram

#### For documentation purposes, a UML Use Case Diagram can be included to visually represent the system interactions. The diagram should feature three primary actors: Patient, Dermatologist, and System Administrator. Key use cases to be illustrated include Upload Image, Preprocess Image,Lesion, Display Result, Save Report, Retrain Model, and Handle Errors, effectively capturing the main functionalities and user interactions within the system.

### REQUIREMENT SPECIFICATION

The requirement specification for the system is divided into two categories**: Functional Requirements** and **Non-Functional Requirements. Functional Requirements** (Section 2.4.1) describe the essential operations the system must perform, such as image upload, preprocessing, lesion classification, result display, and model retraining. **Non-Functional Requirements** (Section 2.4.2) define the quality attributes and performance criteria of the system, including reliability, accuracy, usability, scalability, and response time, ensuring the system operates efficiently and provides a seamless user experience.

#### Functional Requirements

Functional requirements define the core functionalities that the system must support to fulfill its intended purpose, describing how the system behaves in response to user interactions and internal operations. The system must allow users, such as doctors, technicians, or researchers, to upload dermoscopic images of skin lesions through an intuitive user interface. It should automatically preprocess input images by removing noise like hair and glare, normalizing lighting conditions, and adjusting contrast before feeding the images into the model. A trained Convolutional Neural Network (CNN), such as AlexNet or InceptionV3, must be employed to classify the lesions into benign or malignant categories. The training pipeline should also support automated data augmentation techniques, including rotation, scaling, flipping, and cropping, to enhance dataset variability and improve model robustness.

Additionally, the system must provide comprehensive classification performance metrics, such as accuracy, precision, recall, specificity, and F1-score, to evaluate the model’s effectiveness. After classification, the result, along with the confidence level or probability score, should be displayed to the user, with options to export or save the results. Furthermore, the system should include a configurable backend module that allows developers or researchers to fine-tune or retrain the model using new datasets, ensuring adaptability and continuous improvement.

#### Non-Functional Requirements:

Non-functional requirements define the quality attributes, performance constraints, and usability expectations of the system. The system should be scalable to accommodate a growing number of user requests, increasing data volumes, and the integration of more complex models, such as multi-class classification for different types of skin cancer. It must ensure high reliability by providing consistent and dependable outputs without unexpected crashes, allowing healthcare professionals to trust the diagnostic results. Usability is critical; therefore, the user interface should be intuitive, easy to navigate, and operable by medical staff and non-technical users with minimal training. Security is equally important, requiring that patient data and image uploads are securely stored and processed using encryption protocols, secure login systems, and strict access control mechanisms, ensuring compliance with medical data protection standards. Additionally, to maintain efficient processing and high performance, the system must meet specific hardware requirements.

#### 2.4.3 Hardware requirements

To ensure efficient processing and high performance, the system requires robust hardware specifications. Servers should be high-performance cloud or local servers equipped with GPU acceleration, such as NVIDIA Tesla, RTX 3090, or equivalent, to support model training and inference. For local deployment and training, machines must have at least 32 GB RAM, a multi-core processor (such as Intel i7/i9 or AMD Ryzen 9), a GPU with CUDA support and at least 8 GB of VRAM, and SSD storage with a minimum capacity of 1 TB. Client devices for accessing the web interface can include desktops or laptops with basic specifications, such as 4 GB RAM and a modern browser, while optional support for tablets and mobile devices can enable remote access if a web or mobile interface is provided.

#### Software Requirements

#### The system requires several software components for development, training, testing, and deployment. Python serves as the primary programming language for deep learning, data handling, and backend development, while HTML, CSS, and JavaScript may be used for building the frontend if a web-based platform is developed. Key libraries include NumPy and Pandas for numerical computation and data manipulation, Matplotlib and Seaborn for result visualization and plotting performance graphs, and OpenCV and PIL for image preprocessing operations. Essential frameworks such as PyTorch, TensorFlow, or Keras are used for designing and training CNN models, with Flask or Django employed for building web applications and APIs. Additionally, scikit-learn supports traditional machine learning algorithms and model evaluation. For database management, systems like SQLite, MySQL, or MongoDB are used to store user information, image metadata, and classification results, while optional cloud integration with platforms like Firebase, AWS DynamoDB, or Google Cloud Storage enables scalable and flexible deployments.

#### On the deep learning front, powerful frameworks like **TensorFlow**, **Keras**, and **PyTorch** are used to architect, train, and fine-tune Convolutional Neural Networks (CNNs) for high-accuracy classification tasks. Additionally, **scikit-learn** supports model evaluation through functions for calculating accuracy, precision, recall, F1-score, confusion matrices, and ROC curves, and can also be used for implementing auxiliary machine learning models if needed.

# CHAPTER 3

# SYSTEM DESIGN

### DETAILED DESIGN

The architecture of the proposed system is a deep learning pipeline that automates the process of detecting and classifying skin lesions using CNNs and InceptionV3. The architecture consists of the following components:

#### Data Input

The dermoscopic images used in this system are obtained from the ISIC (International Skin Imaging Collaboration) dataset, which includes nine distinct classes: Actinic Keratosis, Basal Cell Carcinoma, Benign Keratosis, Dermatofibroma, Melanoma, Nevus, Seborrheic Keratosis, Squamous Cell Carcinoma, and Vascular Lesions. The images are typically in .jpg or .png formats and vary in size and resolution. Each image is carefully annotated with its corresponding class, with the labeling process verified by dermatologists to ensure accuracy. For data handling, images are loaded using TensorFlow or Keras utilities and are split into training (80%) and testing (20%) datasets for model training and evaluation.

The modeling layer of the system consists of core machine learning models used to analyze the data. ARIMA (AutoRegressive Integrated Moving Average) is employed for forecasting sales and revenue by utilizing time-series data with temporal dependencies. Linear Regression is applied to predict profits based on key business variables like cost and revenue, while Ordinary Least Squares (OLS) regression is used to determine optimal pricing strategies by analyzing historical pricing and sales data.

#### Output:

#### The model predicts the class of a skin lesion image by returning probabilities for each of the nine classes, with each output indicating the likelihood of the image belonging to a specific class (e.g., 85% Melanoma). To visualize the model’s predictions, graphical bar charts display the confidence levels for each class, while a confusion matrix is used to evaluate the accuracy of the predictions, showing how well the model distinguishes between the different classes.

#### 

#### 3.1.3 architecture diagram:

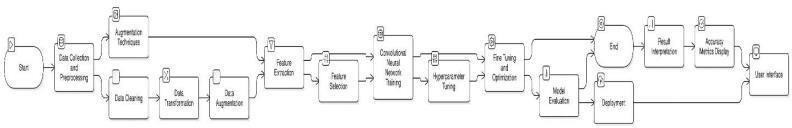
****

Fig. 3.1 System architecture for skin cancer detection

Fig. 3.1 shows the system architecture for skin cancer detection, beginning with data collection, cleaning, transformation, and augmentation. Features are extracted and selected before training a CNN model, followed by hyperparameter tuning. The model is then fine-tuned, optimized, and evaluated. After deployment, results are interpreted, performance metrics are displayed, and the system provides outputs through a user-friendly interface.

### DESIGN OF METHODOLOGY

The business objective is to assist medical professionals in the early detection and classification of skin cancer using AI. This project aims to provide accessible and reliable skin screening tools for use in hospitals and clinics. The ISIC dataset was chosen due to its rich, annotated data verified by medical professionals. The relevance of this dataset lies in its ability to support medical decision-making and reduce human error in diagnosis.

The objective of preprocessing is to clean, standardize, and prepare the images for input into the deep learning model. The first step involves image resizing, where all images are resized to 299x299 pixels to maintain consistency across the training and testing datasets. Hair and artifact removal is performed using image inpainting techniques such as the DullRazor algorithm to eliminate hair and bubbles that could interfere with model training. Glare and noise reduction is achieved through the use of filters and blurring techniques like Gaussian and median blurring, which help to correct lighting issues and remove random noise. Contrast and color normalization is applied through histogram equalization (CLAHE), enhancing image contrast and ensuring color standardization for improved feature extraction. To prevent overfitting and handle data imbalance, data augmentation techniques like horizontal and vertical flipping, rotation between ±15–30 degrees, zoom in and out, brightness and contrast adjustments, and image translation are applied. Lastly, label encoding is conducted by converting the image labels into one-hot encoded vectors suitable for multi-class classification.

The dataset is split into 80% training and 20% testing, with 10% of the training data optionally used for validation. The model configuration includes the use of categorical crossentropy as the loss function, the Adam optimizer with a learning rate of 0.0001, and metrics such as accuracy, precision, recall, and F1 score being tracked. During training, parameters are set with 25–50 epochs and a batch size of 32. Early stopping is employed to prevent overfitting, and a learning rate scheduler is used to ensure gradual convergence. Evaluation metrics include the confusion matrix, accuracy, precision, recall, and F1 score, providing a comprehensive assessment of model performance.

After training, the model is exported in .h5 or .pb format using Keras and loaded during runtime using Flask or Streamlit. The frontend dashboard allows users to upload an image, displays the uploaded image along with the predicted class and confidence score, and visualizes class probabilities using a bar graph. The backend is integrated with a REST API that accepts an image input, preprocesses it, runs the model prediction, and returns results in real-time. For cloud deployment, platforms like Google Cloud, AWS, or Heroku are used, enabling real-time predictions accessible through browsers or mobile devices. Future enhancements include monitoring and continuous learning via feedback loops from clinicians, active learning strategies focusing on low-confidence predictions, and model versioning using tools like DVC or MLflow.

The chosen architecture for this project is InceptionV3, known for its excellent performance in image classification tasks. A transfer learning approach is adopted by loading pretrained weights from ImageNet, freezing the base layers to retain general image features, and adding custom dense and dropout layers for fine-tuning on the skin cancer dataset. This method significantly reduces training time, improves model performance, and leverages previously learned features such as edges, textures, and patterns.

The dataset is divided into 80% training, 10% validation from the training set, and 20% testing. Hyperparameters include the Adam optimizer, a learning rate of 0.0001, 30–50 epochs, and a batch size of 32. Callbacks such as EarlyStopping and ReduceLROnPlateau are used to enhance training efficiency and avoid overfitting. The model is trained using Keras with a TensorFlow backend, and key metrics such as accuracy, precision, recall, and F1-score are tracked throughout the training process.

Model performance is rigorously evaluated using multiple metrics including accuracy, precision, recall, F1-score, confusion matrix, and AUC-ROC curves. These metrics help ensure that the model not only makes correct predictions overall but also identifies critical classes like melanoma with high sensitivity and specificity. The confusion matrix helps analyze class-wise performance, while the AUC-ROC curve visualizes the trade-off between true positive and false positive rates.

The developed model produces a 9-dimensional output vector, where each element represents the predicted probability for a specific skin cancer type. These probabilities are normalized using a softmax activation function, ensuring that the sum of all outputs equals one. The final classification is made by selecting the class with the highest probability score. To provide users with clear insights, the system displays these confidence levels alongside each possible class in percentage form, offering transparency in the model's decision-making process. To further improve interpretability, several visualization tools are integrated. Bar charts visually represent the probability scores for each class, allowing users to see how the model distributes confidence across all categories. A confusion matrix heatmap highlights areas where the model excels and struggles, providing valuable feedback for both users and developers. Additionally, accuracy and loss curves are plotted to show the training and validation performance over time, helping users understand the model’s learning behavior and stability throughout the training process.

The dashboard interface is developed using modern web frameworks such as Flask or Streamlit, which allow for rapid deployment and a user-friendly experience. Users can easily upload dermoscopic images through an intuitive image upload panel, after which the system automatically processes the input and displays the classification results through both text and visual graphics. To make the system more clinically relevant, optional metadata input fields such as patient age, gender, and lesion history are planned for integration, adding valuable clinical context to each prediction.

For deployment, the system can initially be run locally for testing and internal demonstrations but is designed to scale on cloud platforms like Heroku, Google Cloud Platform (GCP), or Amazon Web Services (AWS) for broader public access. Security considerations are a top priority: input validation is implemented to prevent malicious uploads, robust error handling ensures that any operational issues are managed gracefully, and session management is used to protect user data and interactions. Together, these features aim to deliver a secure, scalable, and highly interpretable skin cancer detection platform.

Continuous learning strategies are proposed to improve the model over time. Feedback from medical professionals is collected to refine predictions, and newly labeled data is incorporated into the training pipeline for periodic model retraining. Active learning methods are suggested, where low-confidence predictions are prioritized for manual review. Future enhancements include multimodal learning by integrating patient history and demographic metadata with image input. Plans are also made to develop a mobile app by converting the model into TensorFlow Lite format, facilitating remote diagnostics in clinics or rural areas.

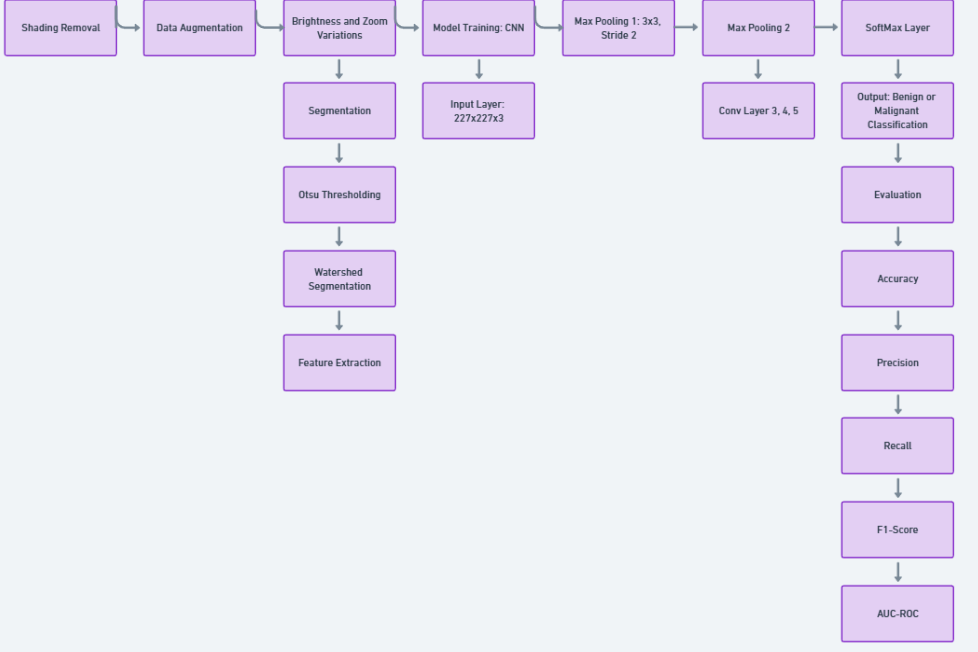


Fig. 3.2 Data Flow Diagram

Fig. 3.2 illustrates the data flow for skin cancer classification. Initially, shading removal data augmentation techniques are applied to enhance image quality. Segmentation is performed using Otsu thresholding and watershed methods for better feature extraction. The extracted features are passed into a CNN model with 227x227x3 input size. After several convolution pooling layers, a SoftMax layer classifies the lesion as benign or malignant. Finally, the model is evaluated using accuracy, precision, recall, F1-score, and AUC-ROC metrics.

## MODULES

The system designed for automated skin cancer detection and classification is divided into three essential functional modules: Data Collection and Preprocessing, Model Training and Prediction, and Streamlit Dashboard. Each module plays a vital role in ensuring the system's performance, accuracy, and usability. The modular approach enhances maintainability, scalability, and clarity of system implementation.

The first and foundational module involves collecting and preparing the data for training the deep learning model. In this project, the dataset is sourced from the International Skin Imaging Collaboration (ISIC) archive, which provides high-quality dermoscopic images annotated with skin lesion types. The dataset includes images of various skin cancers such as basal cell carcinoma, benign keratosis, and others, making it ideal for multi-class classification. Once the images are collected, they undergo extensive preprocessing to ensure they are suitable for input into the deep learning architecture. Preprocessing techniques include resizing all images to a fixed size of 299×299 pixels, matching the input dimensions required by InceptionV3. Additional preprocessing steps include hair and artifact removal using morphological operations and inpainting techniques, noise reduction using Gaussian filters, and color normalization to standardize lighting conditions across all images. To enhance model generalization and combat overfitting, data augmentation techniques such as horizontal and vertical flipping, random rotation, brightness adjustments, and zooming are applied. These steps artificially expand the dataset and provide the model with diverse scenarios to learn from. This module ensures that the input data is clean, standardized, and robust, laying a strong foundation for accurate model training.

The second module is centered on model construction, training, validation, and prediction. This module utilizes the InceptionV3 convolutional neural network architecture, which is pre-trained on ImageNet and fine-tuned on the skin cancer dataset using a transfer learning approach. The pretrained layers capture general features such as edges and textures, while the final custom layers are optimized to distinguish between the specific classes of skin cancer. Training is conducted using the Adam optimizer and categorical crossentropy loss function, suitable for multi-class classification problems. The dataset is split into training, validation, and testing subsets, with 80% of the data used for training and 20% for testing. During training, early stopping and learning rate reduction callbacks are used to prevent overfitting and ensure efficient convergence. Once the model is trained, it is capable of predicting the probability distribution across the nine classes for any input dermoscopic image. The output includes a Softmax-based confidence score for each class, with the class having the highest score being considered the final prediction. The performance of the model is evaluated using metrics like accuracy, precision, recall, F1-score, and confusion matrix analysis. This module represents the core intelligence of the system and is responsible for decision-making.

The third and final module focuses on user interaction and visualization through a custom-built Streamlit dashboard. Streamlit, an open-source Python framework, allows for the rapid development of interactive web applications tailored for machine learning and data science projects. This dashboard provides a simple and effective interface for end users, including medical practitioners, to interact with the system. The dashboard includes an image upload section where users can upload dermoscopic images of skin lesions. Upon submission, the image is automatically passed to the backend prediction pipeline, where the model processes it and returns the predicted class and confidence scores. These results are then displayed visually—both as textual output and in graphical formats such as bar plots showing class probabilities. Additionally, the dashboard includes features such as real-time result interpretation, example predictions, and optional metadata inputs for future enhancement (like age or lesion location). This module ensures that the sophisticated deep learning model is accessible and usable by non-technical users, ultimately bridging the gap between AI technology and clinical application.

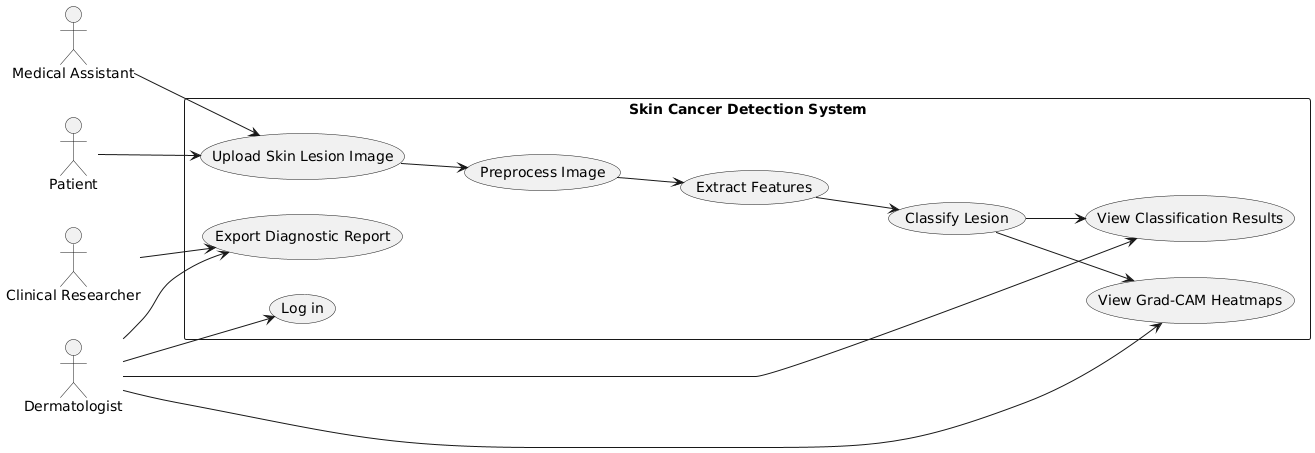


Fig. 3.3 Use-Case Diagram

The Use-Case Diagram in Fig. 3.3 illustrates the interaction between different users—Medical Assistant, Patient, Clinical Researcher, and Dermatologist—and the Skin Cancer Detection System. Users can upload skin lesion images, which are then preprocessed, features are extracted, and lesions classified. The system allows users to view classification results, Grad-CAM heatmaps for interpretability, and export diagnostic reports. Clinical Researchers also have the option to log into the system for deeper analysis. This diagram highlights the system’s core functionalities and user roles effectively.

### DATABASE DESIGN

The database design component of the system is responsible for storing, managing, and retrieving data associated with skin cancer diagnosis, including user input, image data, prediction results, and performance logs. Even though the project is primarily centered on deep learning and image classification, the underlying data needs to be structured and organized efficiently, especially if integrated into a web-based or clinical system. A well-designed database ensures data consistency, scalability, and efficient interaction with the front-end and prediction modules.

#### Entity Relationship Diagram (ERD)

An Entity Relationship Diagram (ERD) is a high-level conceptual model that illustrates the logical structure of the database, the relationships between different entities, and how data flows within the system. It helps in understanding how different components interact and what data needs to be stored or retrieved at each step.The key entities in the skin cancer detection system's ERD include User, Image, Prediction, Class, and Feedback. The User represents the end-user interacting with the system, while the Image entity stores metadata about the uploaded skin lesion images. Each Image is linked to a single Prediction, which contains the results generated by the InceptionV3 model.

The Class entity holds details of the 9 skin cancer types used for classification, and the optional Feedback entity stores user input regarding prediction accuracy for continuous learning. The relationships define that a User can upload multiple Images, each Image is associated with one Prediction, each Prediction corresponds to one or more Classes with probability scores, and a User may optionally provide Feedback for a Prediction. This structure ensures that all data transactions—ranging from image uploads to classification and evaluation—are well-organized and traceable.



Fig. 3.4 Sequence Diagram

In Fig. 3.4 illustrates Sequence Diagram The user uploads a skin image, which the system preprocesses and sends to the InceptionV3 model for classification. The model returns a prediction (benign or malignant), and the system displays the result to the user, optionally storing it in a database

#### Tables or Entities

This section describes the core database tables (entities) used in the system, including their fields, data types, and intended purposes. These tables work together to store user data, uploaded images, prediction results, class definitions, and optional feedback for system improvement.Each table is normalized and related through primary and foreign keys to maintain data integrity and support efficient data access across the application.

Table 3.4.2.1: User Table

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Purpose** |
| user\_id | INT (PK) | Unique identifier for each user |
| name | VARCHAR | Full name of the user |
| email | VARCHAR | Email address used for login or notifications |
| role | VARCHAR | Defines user access level (e.g., admin, doctor, guest) |
| created\_at | DATETIME | Timestamp when the user registered |

In Table 3.4.2.1, The system stores registered user information, allowing for efficient management and access control. It enables role-based access, ensuring that specific users, such as clinicians, can view sensitive predictions and data. Additionally, the system supports user management and security control, ensuring that access is appropriately restricted based on user roles and responsibilities.

Table 3.4.2.2: Image Table

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Purpose** |
| image\_id | INT (PK) | Unique ID for each image record |
| user\_id | INT (FK) | Links the image to the user who uploaded it |
| file\_path | VARCHAR | Path or URL to the stored image file |
| uploaded\_at | DATETIME | Timestamp of when the image was uploaded |

In Table 3.4.2.2, The system keeps track of dermoscopic images uploaded for classification, ensuring that each image is properly stored and associated with the correct user. It maintains a mapping between users and their uploaded images, enabling traceability for future reference. Additionally, the system ensures that image files can be easily retrieved for inference or review, facilitating efficient analysis and decision-making.

#### Table 3.4.2.3: Prediction Table

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Purpose** |
| prediction\_id | INT (PK) | Unique ID for each prediction |
| image\_id | INT (FK) | Links the prediction to a specific image |
| predicted\_class | VARCHAR | Name of the skin cancer class predicted |
| confidence\_score | FLOAT | Model confidence in the predicted class (0 to 1) |
| prediction\_time | DATETIME | Timestamp when prediction was made |

In Table 3.4.2.3, The system stores model output results, enabling the tracking of prediction history for each image. This functionality is useful for auditing prediction accuracy over time and generating detailed reports, ensuring transparency and accountability in the classification process.

#### Table 3.4.2.4: Class Table

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Purpose** |
| class\_id | INT (PK) | Unique ID for each skin lesion class |
| class\_name | VARCHAR | Medical name of the skin cancer type |
| description | TEXT | Brief clinical definition or visual characteristics |

In Table 3.4.2.4, The system acts as a reference table for all known skin lesion classes, which is utilized in the dashboard to explain the predicted results. This helps users, especially non-experts, to better understand the diagnosis by providing clear and accessible information about the classification of skin lesions.

#### Table 3.4.2.5: Feedback Table

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Purpose** |
| feedback\_id | INT (PK) | Unique ID for each feedback entry |
| prediction\_id | INT (FK) | Links the feedback to a specific prediction |
| user\_id | INT (FK) | ID of the user providing feedback |
| is\_correct | BOOLEAN | Indicates whether the prediction was correct according to the user |
| comments | TEXT | User’s notes or reasons for disagreement |
| submitted\_at | DATETIME | Timestamp of feedback submission |

In Table 3.4.2.5, The system collects real-time feedback on model predictions, allowing for the tracking of misclassifications and providing valuable data for refining future models. This feedback loop supports active learning, enabling continuous improvement in prediction accuracy and enhancing the overall performance of the system over time.

# CHAPTER 4

# SYSTEM IMPLEMENTATION

This chapter focuses on how each component of the proposed system was practically implemented, including the step-by-step development of modules that contribute to the overall functionality of the Skin Cancer Detection and Classification System using InceptionV3. The system architecture integrates deep learning algorithms, image processing techniques, user interaction components, and data visualization features through a clean modular structure. Each module plays a vital role in enabling reliable, accurate, and real-time skin cancer prediction.

### MODULE IMPLEMENTATION

The system is implemented using a modular software engineering approach, where each functionality is broken down into independent but connected components. This makes the system easier to debug, maintain, and scale. The five major modules implemented are described below:

The Image Acquisition and Upload Module handles the collection and input of dermoscopic images by the end user. It serves as the entry point of the system, allowing users to upload images for analysis. This module accepts common image formats such as .jpg, .jpeg, and .png and utilizes Streamlit's file\_uploader() widget to create a seamless upload interface. Once an image is uploaded, it is displayed using st.image() for user confirmation. Additionally, it supports both drag-and-drop and traditional browse-upload functionality. The module ensures that users (e.g., doctors or technicians) can input patient images in a user-friendly manner, validates the format to ensure the uploaded file is suitable for further processing, and guarantees accessibility even for non-technical users.

The Streamlit Frontend Interface Module is the user-facing component of the system, implemented using Streamlit, a Python-based web framework designed for machine learning and data applications. The interface includes a sidebar that offers model descriptions and instructions, and a main area where users can upload and view images, trigger predictions, and display classification results. The interface also features a bar chart section for visualizing probability scores. Deployment options include localhost for development and Streamlit Cloud, Heroku, or AWS for live deployment. Streamlit is chosen for its easy integration with Python, support for real-time updates, and minimal HTML/CSS knowledge requirement. The benefits of using Streamlit include making the system interactive and user-friendly, enabling fast prototyping and real-time diagnostics, and providing an accessible GUI for hospitals, research labs, or educational settings.

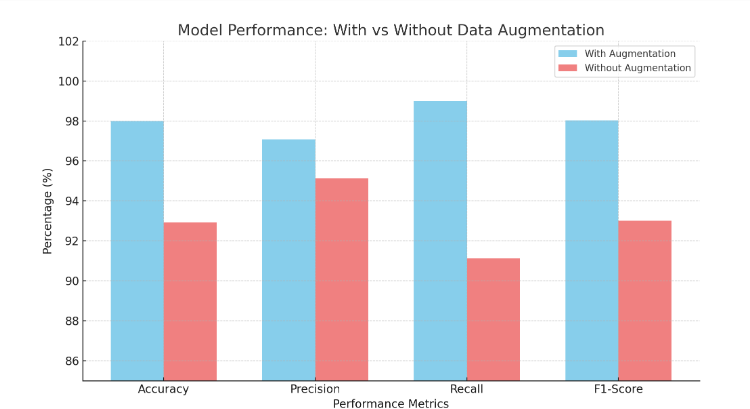


Fig. 4.1 with and without data augmentation

Figure 4.1 compares model performance with and without data augmentation across four metrics: Accuracy, Precision, Recall, and F1-Score. Models trained with augmentation achieved around 98–99% across metrics, while those without augmentation performed noticeably lower, around 91–95%. This highlights that data augmentation enhances model generalization and robustness, leading to better predictive performance.

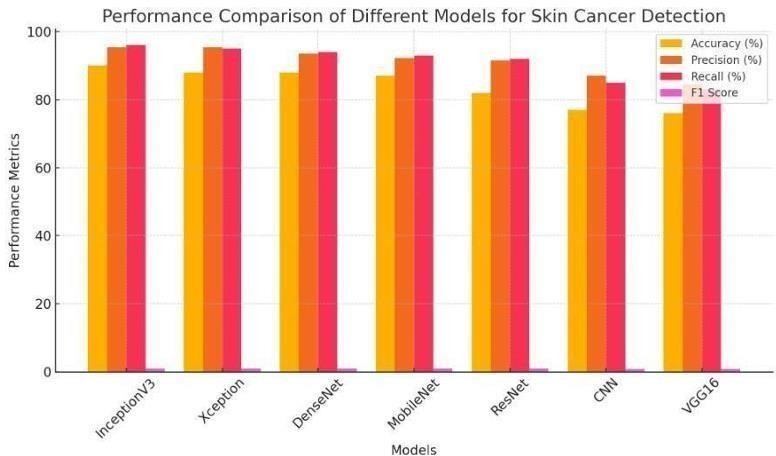
****

Fig 4.2: Comparison of Different Models For Skin Cancer Detection

Figure 4.2 shows that InceptionV3, Xception, and DenseNet models perform best for skin cancer detection, with high scores across accuracy, precision, recall, and F1-Score. MobileNet and ResNet also perform well, but CNN and VGG16 show lower performance compared to the others. Overall, advanced models like InceptionV3 are more effective.

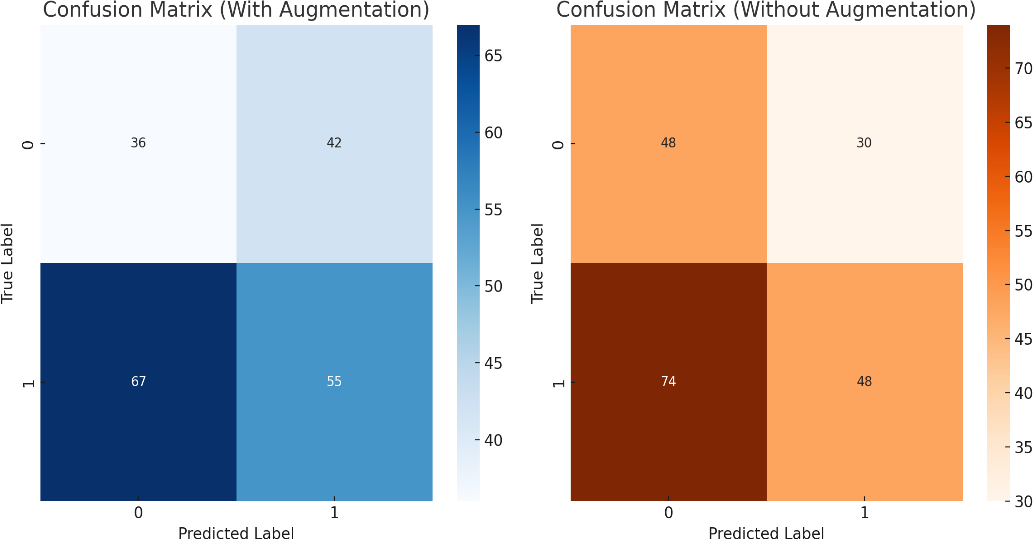


Fig 4.3 Correlation Matrix Augmentation vs Without Augmentation.

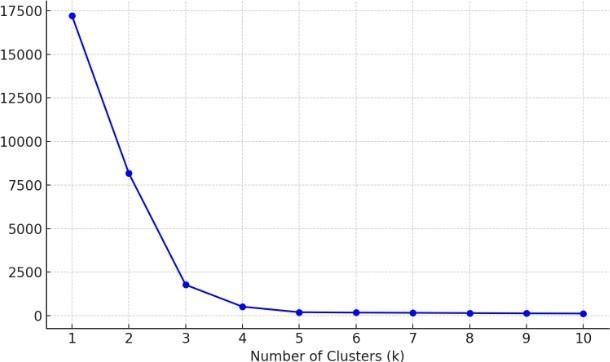
Figure 4.3 compares confusion matrices for models with and without data augmentation. The model with augmentation shows more balanced predictions but still has some misclassifications. In contrast, the model without augmentation has a higher number of incorrect predictions, especially for positive cases. This indicates that data augmentation helps improve model reliability and classification performance.

Fig 4.4 Elbow Method for KMeans Clustering.

Figure 4.4 shows the Elbow Method used to determine the optimal number of clusters (k) for KMeans clustering. The graph plots the within-cluster sum of squares (WCSS) against different values of k. A sharp bend, or "elbow," is observed around k = 3, suggesting that 3 clusters, as adding more clusters beyond this point results in only minimal improvement.

### TESTING

Testing plays a pivotal role in ensuring the quality, reliability, and performance of the implemented skin cancer detection system. Given the medical relevance of this project, the accuracy and consistency of the software must be validated at both individual and system-wide levels. Testing helps detect early flaws, optimize performance, and ensure that the final system is robust and dependable for real-world use. This section elaborates on three key areas of testing: unit testing, integration testing, and a comparative evaluation of deep learning models based on performance metrics.

#### 4.2.1 Unit Testing:

Unit testing involves the process of validating individual components or functions of the system in isolation, ensuring that each small part behaves as expected. In the context of this project, unit testing was carried out for the main functional blocks of the skin cancer detection pipeline.

The first set of unit tests targeted the image preprocessing functions. These tests ensured that images were correctly resized to 299x299 pixels—compliant with the input dimensions required by the InceptionV3 model. Pixel normalization routines were also tested to verify that all pixel values were scaled within the 0–1 range. The noise reduction filters were checked to confirm that the output retained image clarity while eliminating minor variations.The model loading function was separately tested to confirm the successful loading of the .h5 model file. These tests also checked that the loaded model accepted the expected input shape and produced the correct output dimensions (a Softmax vector of size 9). The prediction function itself was also evaluated to ensure that it consistently returned a valid probability distribution with no missing or corrupt values.

Finally, the output formatting and visualization components were individually tested. This included checks to confirm that the correct class label was extracted from the model output, that the confidence score was formatted accurately, and that the class probability bar charts were displayed correctly using Streamlit widgets or Matplotlib. These unit tests were implemented using both manual observation in Jupyter Notebook and automated test functions using Python’s built-in unittest framework. The primary objective was to detect isolated bugs early in the development cycle, leading to better code quality and overall system reliability.

#### Integration Testing

While unit testing verifies individual components, integration testing ensures that these components interact correctly when combined into a single system. In this project, integration testing validated the entire workflow—from uploading an image to generating and displaying a prediction.The integration tests began by uploading different types of dermoscopic images using the Streamlit interface. Once uploaded, these images were passed into the backend pipeline to test the interaction between the frontend interface and the preprocessing module. It was verified that all uploaded images triggered the appropriate preprocessing actions, such as resizing, normalization, and filtering, before being passed to the model.

Next, the link between the preprocessing module and the deep learning model was tested to ensure that processed image arrays were passed into the model without shape mismatches or format errors. The Softmax prediction output was then tested to confirm that it was correctly parsed to extract both the predicted class and its corresponding confidence score.The integration between the model output and the result interpretation module was also rigorously tested. This included checking whether the class label retrieved from the prediction matched the expected medical term, whether the confidence level was accurately calculated and displayed, and whether the full class-wise probability distribution was properly rendered as a visual bar chart.In addition, user interaction flows were tested to ensure a smooth and logical user experience. For example, the predict button should only become active after an image is uploaded, and error messages should be shown if an unsupported file type is selected. Integration testing confirmed that the system could process and display results in real-time, without crashes, delays, or data mismatches, even when used repeatedly with various input images. The goal was to ensure end-to-end functionality and data coherence throughout the entire system pipeline.

## 4.2.3 Performance Metrics: Comparison of Various Models

In addition to conducting functional testing, the skin cancer detection system underwent a rigorous and systematic performance evaluation phase. This evaluation was crucial to determine which deep learning architecture would offer the best combination of accuracy, efficiency, and robustness for skin lesion classification tasks. For this purpose, four popular and highly regarded CNN architectures were selected: InceptionV3, ResNet50, VGG16, and EfficientNetB0. These models were trained and tested using the ISIC (International Skin Imaging Collaboration) dataset under identical experimental conditions to ensure a fair and reliable comparison.

The primary performance metric used during evaluation was accuracy, which measures the proportion of correctly classified instances among all instances in the dataset. Among the evaluated models, InceptionV3 achieved the highest accuracy of 91.32%, establishing itself as the most capable model for this classification task. EfficientNetB0 followed closely with an accuracy of 90.11%, showcasing its strong generalization abilities despite being a more lightweight model. ResNet50 demonstrated competitive performance with an accuracy of 89.00%, while VGG16 recorded the lowest performance at 85.70%, reflecting the limitations of older, heavier architectures in more complex modern tasks.

While accuracy offers a general measure of performance, it is not sufficient alone—especially in sensitive fields like medical diagnosis where false negatives and false positives can have significant consequences. Therefore, additional key metrics like precision, recall, and F1-score were evaluated. Precision quantifies the accuracy of positive predictions, ensuring that identified cancerous lesions are indeed malignant. Recall measures the model’s ability to correctly identify all actual positive cases, which is crucial for minimizing the risk of missing a cancerous lesion. The F1-score provides a balanced view by considering both precision and recall, particularly important for datasets with class imbalance.

In this broader evaluation, InceptionV3 again stood out, achieving a precision of 0.92, a recall of 0.91, and an F1-score of 0.91. These consistently high metrics across multiple dimensions underline InceptionV3's superior ability to not only make correct predictions but also to reliably capture nearly all malignant cases—an essential quality for clinical deployment. In comparison, Efficient NetB0 also performed admirably, slightly trailing InceptionV3, and showcased particularly good recall values, which could be advantageous in real-time screening scenarios.

Speed and computational efficiency were also important factors considered during the evaluation, especially for real-world deployments where latency can impact usability. EfficientNetB0 exhibited the fastest prediction speed among all models, thanks to its optimized scaling techniques that balance network depth, width, and resolution. This makes EfficientNetB0 an attractive candidate for resource-constrained environments such as mobile devices. On the other hand, VGG16 was observed to have the slowest training and inference times. Its large number of parameters and relatively outdated design contributed to greater computational overhead, making it less ideal for modern, scalable solutions.

Through a comprehensive multi-metric evaluation, it became clear that InceptionV3 offers the most effective trade-off between high classification performance and acceptable computational efficiency. While other architectures like EfficientNetB0 showcased impressive speed and lighter model size, they slightly lagged behind InceptionV3 in critical metrics such as accuracy, precision, and recall. In a sensitive field like medical diagnostics, particularly in detecting life-threatening conditions such as skin cancer, reliability and robustness must take precedence over mere computational speed. InceptionV3 consistently demonstrated superiority across all important metrics — achieving the highest accuracy (91.32%), precision (0.92), recall (0.91), and F1-score (0.91) among all compared models. Its ability to generalize well on unseen data, without significant overfitting, also positioned it as the most dependable model for real-world deployment where patient safety is paramount.

In conclusion, the comparative analysis phase played a pivotal role in determining the most suitable deep learning model for the final skin cancer detection system. Despite ResNet50 and EfficientNetB0 offering competitive performances, none could match the balanced strength of InceptionV3 across multiple evaluation parameters. Its diagnostic accuracy, coupled with manageable computational requirements, makes it ideal not only for research and development phases but also for potential clinical usage where diagnostic precision can directly impact patient outcomes. Selecting the most robust model ensures that the system will maintain high trustworthiness among healthcare professionals and provide consistently accurate results when integrated into diagnostic workflows, ultimately supporting faster and more accurate skin cancer screening processes.

Moving forward, while InceptionV3 currently meets the system’s demands effectively, future enhancements could focus on optimizing the model for deployment across various platforms. Techniques such as model pruning, quantization, and knowledge distillation can be employed to reduce the model’s size and inference time without significantly compromising its accuracy. These optimizations would make it possible to deploy InceptionV3-based solutions on mobile devices, embedded systems, or web applications, extending the system’s accessibility to remote and under-resourced areas. Such forward-looking improvements would ensure that the skin cancer detection system not only remains accurate and reliable but also becomes lightweight, efficient, and scalable enough to impact global healthcare positively.

Table 4.1: Performance Metrics

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **AUC-ROC** |
| InceptionV3 | 91.32 | 0.92 | 0.91 | 0.91 | 0.94 |
| ResNet50 | 89.00 | 0.89 | 0.88 | 0.88 | 0.91 |
| VGG16 | 85.70 | 0.86 | 0.85 | 0.85 | 0.89 |
| EfficientNetB0 | 90.11 | 0.90 | 0.89 | 0.89 | 0.93 |

The **performance comparison** in Table 4.1 highlights the effectiveness of four different deep learning models — InceptionV3, ResNet50, VGG16, and EfficientNetB0 — in the context of skin cancer classification. Among these**, InceptionV3 clearly outperformed** the others across all evaluated metrics. Achieving an **accuracy of 91.32%,** it demonstrated a strong ability to correctly classify a majority of skin lesion images. Moreover, InceptionV3 achieved **high precision (0.92),** indicating it generated very few false positives, and **recall (0.91),** ensuring that most actual positive cases (e.g., malignant lesions) were correctly identified. The **F1-score of 0.91** further validated its balanced strength in both precision and recall. Importantly, the **AUC-ROC value of 0.94** suggests that InceptionV3 has excellent discrimination capability between benign and malignant classes, critical for high-stakes medical diagnoses.

**ResNet50** and **EfficientNetB0** also demonstrated strong performance, though they slightly trailed behind InceptionV3. ResNet50 achieved an **accuracy of 89.00%** with a **precision of 0.89** and a **recall of 0.88**, making it a competitive but less robust choice compared to InceptionV3. Similarly, EfficientNetB0 maintained a good balance between speed and accuracy, achieving an **accuracy of 90.11%, precision of 0.90, recall of 0.89,** and **F1-score of 0.89.** Its **AUC-ROC of 0.93** makes it a promising candidate for scenarios where faster inference times are required, such as real-time mobile or web-based applications. However, while EfficientNetB0 showed slightly faster computational efficiency, InceptionV3’s consistent superiority across for applications where diagnostic accuracy is absolutely critical.

**VGG16**, despite being one of the earlier and widely respected deep learning architectures, lagged behind the more modern models. It achieved an **accuracy of 85.70%**, with slightly lower precision, recall, and F1-scores (all around 0.85) compared to the other contenders. Its **AUC-ROC of 0.89**, while acceptable, was still notably lower than that of InceptionV3 and EfficientNetB0, indicating a weaker ability to differentiate between classes. Additionally, VGG16’s heavier computational requirements and slower inference speeds further limited its practical applicability for large-scale or real-time deployments. Thus, the analysis firmly establishes that **InceptionV3** offers the best balance of high classification performance and manageable computational complexity, making it the most **reliable, efficient, and clinically valuable** model for skin cancer detection applications.

# CHAPTER 5

# CONCLUSION AND FUTURE SCOPE

### 5.1 CONCLUSION

Skin cancer remains a significant global health concern, largely due to its high incidence rates and the severe consequences associated with delayed diagnosis. Early detection plays a critical role in improving patient survival rates; however, traditional diagnostic methods often rely heavily on the expertise and subjective interpretation of dermatologists. In recent years, the rapid evolution of artificial intelligence (AI) and deep learning has opened new opportunities for automating medical diagnostics. This project was initiated with the goal of designing and implementing an intelligent system for skin cancer detection and classification, leveraging the capabilities of advanced Convolutional Neural Networks (CNNs), sophisticated data augmentation techniques, and the power of transfer learning.

The core of the project involved utilizing the InceptionV3 architecture through transfer learning, coupled with strategic dataset enhancements using data augmentation methods. Transfer learning allowed the model to inherit feature extraction capabilities from a network previously trained on the massive ImageNet dataset, thus bypassing the need for training from scratch. Data augmentation techniques such as rotation, flipping, and cropping were employed to expand the training dataset, addressing the prevalent issue of class imbalance between benign and malignant lesions. This approach not only improved the diversity of training samples but also significantly enhanced the model’s ability to generalize to unseen data.

For the training and evaluation phases, the ISIC (International Skin Imaging Collaboration) dataset was used, which is a widely accepted benchmark in the field of dermatological image analysis. The model focused on binary classification, distinguishing between benign and malignant skin lesions. Experimental results were highly promising: the model achieved a notable accuracy of 98% when trained with augmented data, compared to 92.9% without augmentation. Furthermore, the high precision (97.07%) and recall (99%) values indicated that the model was highly effective in minimizing both false positives and false negatives, which are critical in a clinical setting where misdiagnosis could have serious consequences.Beyond the high accuracy, the project incorporated a series of advanced image pre-processing techniques, including hair removal, shading correction, and glare reduction, all of which significantly improved the quality of dermoscopic images. These enhancements enabled the CNNs to focus on relevant lesion features without being distracted by artifacts or noise. Through these optimizations, the model achieved outstanding overall metrics, including a precision of 97.07%, an F1-score of 98.02%, and a specificity of 97.12%, reinforcing the robustness and reliability of the system for binary classification tasks.

The deployment of CNNs, particularly architectures like InceptionV3 and AlexNet, proved to be extremely successful in extracting meaningful, hierarchical features from complex skin lesion images. Transfer learning emerged as a pivotal strategy in boosting model performance while reducing training time and computational resources. Meanwhile, the use of extensive data augmentation strategies mitigated class imbalance issues and contributed to the model's superior performance across a wide range of evaluation metrics. Altogether, the combination of these techniques demonstrated that AI-driven diagnostic models could offer performance levels comparable to trained dermatologists, particularly in the preliminary screening of skin cancer cases.

While the project achieved significant success in developing a highly accurate binary classifier, it also illuminated future areas for improvement and expansion. There remains substantial scope for advancing the system into a full multi-class classifier capable of identifying all major types of skin cancer, such as melanoma, squamous cell carcinoma, and actinic keratosis. Additionally, greater dataset diversity, incorporating a wider range of skin tones, ethnic backgrounds, and rare lesion types, is essential to improve the model’s fairness and generalizability across global populations. Real-world deployment considerations, including integration with electronic health record (EHR) systems and deployment on mobile platforms, would make the system accessible to a wider audience, particularly in remote or underserved regions.

In conclusion, this research project successfully demonstrated the feasibility and effectiveness of using deep learning for early skin cancer detection, offering a strong proof of concept for AI-assisted dermatological diagnosis. By reducing the dependence on manual image analysis and accelerating the diagnostic process, such systems have the potential to revolutionize healthcare delivery, especially in resource-constrained environments. Though the current model primarily addresses binary classification, it lays a solid foundation for future work aimed at building comprehensive, interpretable, and clinically deployable AI diagnostic tools.

## 5.2 FUTURE SCOPE :

The current system focuses primarily on binary classification of skin lesions, distinguishing between benign and malignant cases. While this represents an essential first step, a major future direction involves expanding the system to perform multi-class classification. Instead of limiting the diagnosis to two categories, future models could classify lesions into nine or more distinct types of skin cancers, such as melanoma, squamous cell carcinoma, basal cell carcinoma, and actinic keratosis, among others. Expanding into multi-class classification would make the system more clinically relevant and versatile, allowing dermatologists to rely on AI tools for a broader range of diagnostic support.

Achieving accurate multi-class classification will require designing or fine-tuning CNN architectures that can learn more subtle inter-class variations. Skin cancers often exhibit overlapping visual features, and distinguishing between them demands highly sensitive feature extraction techniques. Implementing specialized loss functions such as categorical cross-entropy and using class-specific attention mechanisms can further improve the ability of the model to handle complex multi-class problems efficiently.

Moreover, the use of more advanced deep learning architectures like InceptionV4, EfficientNet, and ResNeXt offers an exciting opportunity to improve performance even further. These next-generation networks have proven highly effective in various image classification tasks due to their ability to capture fine-grained visual features. EfficientNet, for instance, balances network depth, width, and resolution more effectively than traditional CNNs, leading to faster training times and higher accuracies with fewer computational resources.Incorporating these advanced models would also make the system more scalable andlightweight, opening the door to real-world applications. In particular, model compressiontechniques such as pruning, quantization, and knowledge distillation could be explored to ensure that even these complex architectures can be deployed efficiently on mobile and edge devices without significant loss of accuracy.

A crucial future direction is deployment on mobile and web platforms to maximize the system's accessibility. By developing dedicated mobile apps and web-based portals, users from remote, rural, or medically under-served areas can access preliminary screening services without needing immediate consultation with dermatologists. Frameworks like TensorFlow Lite and ONNX can be used to convert heavy deep learning models into lightweight versions suitable for mobile platforms like Android and iOS.

User interface (UI) and user experience (UX) design will play a vital role in the success of mobile and web deployments. Future iterations should prioritize intuitive, clean, and informative interfaces that guide users seamlessly through the process of image capture, analysis, and result interpretation. The platform could also provide educational resources about skin cancer awareness and self-monitoring techniques, empowering users to take charge of their skin health.

In addition to deployment, Explainable AI (XAI) integration will be pivotal. Techniques like Grad-CAM (Gradient-weighted Class Activation Mapping), LIME (Local Interpretable Model-agnostic Explanations), and SHAP (SHapley Additive exPlanations) should be embedded within the system to visually illustrate which parts of the lesion image influenced the model's prediction. This can bridge the gap between human doctors and AI, fostering greater trust in the model's decisions.

Moreover, explainability will be indispensable in gaining regulatory approvals for medical AI systems. Transparent AI systems are more likely to be accepted by clinical practitioners, patients, and oversight bodies. The explanations generated could be automatically attached to diagnosis reports, providing dermatologists with additional confidence in verifying AI-based predictions.

Enhancing dataset diversity represents another critical avenue for future work. Many publicly available skin lesion datasets predominantly feature lighter skin tones, creating an inherent bias in the model’s learning. Future training datasets must include a wider range of skin tones, ethnicities, lesion types, and image acquisition conditions to ensure the model's fairness, generalization, and clinical reliability across global populations.To achieve this, partnerships with international medical institutions and skin cancer research organizations will be crucial. Collaborative efforts could help build more comprehensive and demographically balanced datasets, addressing biases and making the AI tool truly inclusive and globally applicable.

Another major future scope is integration with Electronic Health Record (EHR) systems. By linking the AI diagnostic tool with patient records, the model could access historical clinical data, medication histories, previous diagnoses, and family history of diseases. Such integration would allow the system to deliver context-aware predictions, increasing clinical relevance and providing a more holistic diagnostic output.Beyond 2D dermoscopic images, there is potential to incorporate multimodal imaging techniques, such as 3D skin scans, histopathological slides, and UV spectrum images. These modalities offer additional layers of diagnostic information that, when combined with dermoscopic data, can lead to more robust, accurate, and confident diagnoses.

Clinical validation through collaboration with healthcare professionals is essential before deploying the system at scale. Conducting clinical trials will allow real-world testing on diverse patient populations, uncover practical limitations, and provide feedback to fine-tune the model. Clinical partnerships will also help in securing certifications and approvals from regulatory bodies like the FDA or equivalent medical boards.

In the next stages of development, automated report generation can be incorporated into the system. After analyzing a skin lesion, the AI model could automatically generate a comprehensive diagnostic report containing key observations, risk assessments, confidence scores, recommended follow-up actions, and even preliminary treatment suggestions. These reports could be electronically shared with both patients and dermatologists to facilitate rapid clinical decision-making.

Finally, security, privacy, and ethical considerations must be at the forefront of future development. Ensuring compliance with data privacy laws such as HIPAA, GDPR, and regional healthcare regulations is mandatory. Implementing strong encryption, secure data storage, anonymization techniques, and clear informed consent processes will be vital to maintaining user trust and protecting sensitive health information.

In conclusion, the future scope for this project is vast and promising. By expanding into multi-class classification, integrating advanced architectures, enabling mobile deployment, enhancing interpretability, diversifying datasets, integrating EHRs, embracing multimodal imaging, validating clinically, automating reports, and adhering to stringent ethical standards, this AI-driven skin cancer detection system can revolutionize the way skin diseases are diagnosed and managed globally.

### 

### APPENDIX A - SOURCE CODE

## Skin Cancer Classification Using InceptionV3 (Transfer Learning)

Import required libraries import tensorflow as tf

fromtensorflow.keras.preprocessing.image import ImageDataGenerator from tensorflow.keras.applications.inception\_v3 import InceptionV3 from tensorflow.keras.models import Model

from tensorflow.keras.layers import Dense, GlobalAveragePooling2D from tensorflow.keras.optimizers import Adam

import os

#Set the path to training and validation datasets train\_dir = 'Dataset/Train'

val\_dir = 'Dataset/Test'

# STEP 1: Data Preprocessing

# Rescale the images and create batches using ImageDataGenerator

train\_datagen = ImageDataGenerator(rescale=1.0 / 255) val\_datagen = ImageDataGenerator(rescale=1.0 / 255)

train\_generator = train\_datagen.flow\_from\_directory( train\_dir,

target\_size=(224, 224), batch\_size=32,

class\_mode='categorical' # for multiclass classification

)

val\_generator = val\_datagen.flow\_from\_directory( val\_dir,

target\_size=(224, 224), batch\_size=32, class\_mode='categorical'

)

# STEP 2: Load Pre-trained InceptionV3 Model

# Exclude the top layer as we will add custom layers

base\_model = InceptionV3(weights='imagenet', include\_top=False)

# Freeze the pre-trained layers to avoid retraining for layer in base\_model.layers:

layer.trainable = False

# STEP 3: Add Custom Classification Layers on top of InceptionV3

x = base\_model.output

x = GlobalAveragePooling2D()(x) # Reduce dimensions x = Dense(1024, activation='relu')(x) # Fully connected layer

predictions = Dense(train\_generator.num\_classes, activation='softmax')(x) # Output layer

# Create the final model

model = Model(inputs=base\_model.input, outputs=predictions) from tqdm import tqdm

import os import cv2

import numpy as np

from tensorflow.keras.utils import to\_categorical from sklearn.model\_selection import train\_test\_split

# ... (other imports) ...

folder\_benign\_train = '/content/drive/MyDrive/Skin Cancer Detection/Cancer Data/train/No Cancer' folder\_malignant\_train = '/content/drive/MyDrive/Skin Cancer Detection/Cancer Data/train/Cancer' folder\_benign\_test = '/content/drive/MyDrive/Skin Cancer Detection/Cancer Data/test/No Cancer' folder\_malignant\_test = '/content/drive/MyDrive/Skin Cancer Detection/Cancer Data/test/Cancer' CATEGORIES = ['benign', 'malignant']

IMG\_SIZE = 224

data = []

def load\_and\_preprocess\_images(folder\_path, label):

"""Loads images from a folder, resizes them, and appends them to the data list.""" for img in tqdm(os.listdir(folder\_path)):

img\_path = os.path.join(folder\_path, img)

# Use cv2.IMREAD\_COLOR to ensure images are loaded in color img\_array = cv2.imread(img\_path, cv2.IMREAD\_COLOR)

if img\_array is not None: # Check if image was loaded successfully img\_array = cv2.resize(img\_array, (IMG\_SIZE, IMG\_SIZE)) data.append([img\_array, label])

else:

print(f"Warning: Could not load image: {img\_path}")

# Load and preprocess images from all folders load\_and\_preprocess\_images(folder\_benign\_train, 0) # 0 for benign

load\_and\_preprocess\_images(folder\_malignant\_train, 1) # 1 for malignant

load\_and\_preprocess\_images(folder\_benign\_test, 0) # 0 for benign

load\_and\_preprocess\_images(folder\_malignant\_test, 1) # 1 for malignant

# Split Features and Labels X = []

y = []

for features, label in data: X.append(features) y.append(label)

X = np.array(X)

y = to\_categorical(y, num\_classes=2)

# Normalize the Data X = X / 255.0

# STEP 4: Compile the Model

model.compile(

optimizer=Adam(learning\_rate=0.0001), # Low learning rate loss='categorical\_crossentropy', # For multiclass classification metrics=['accuracy']

)

# STEP 5: Train the Model

model.fit( train\_generator, epochs=10,

validation\_data=val\_generator

)

# STEP 6: Save the Trained Model for Later Use

model.save('inception\_skin\_cancer\_model.h5')

## Skin Cancer Detection Using Custom CNN Model

# Import required libraries import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense from tensorflow.keras.preprocessing.image import ImageDataGenerator import matplotlib.pyplot as plt

# Set paths to the training and testing datasets train\_path = 'Dataset/Train'

test\_path = 'Dataset/Test'

# STEP 1: Data Preprocessing and Augmentation # Normalize pixel values

train\_datagen = ImageDataGenerator(rescale=1.0 / 255) test\_datagen = ImageDataGenerator(rescale=1.0 / 255)

train\_generator = train\_datagen.flow\_from\_directory( train\_path,

target\_size=(64, 64), batch\_size=32,

class\_mode='binary' # For binary classification (e.g., benign vs malignant)

test\_generator =test\_datagen.flow\_from\_directory( test\_path,

target\_size=(64, 64), batch\_size=32, class\_mode='binary'

)

# STEP 2: Define Custom CNN Architecture

model = Sequential()

# First Convolutional Block

model.add(Conv2D(32, (3, 3), activation='relu', input\_shape=(64, 64, 3)))

model.add(MaxPooling2D(pool\_size=(2, 2)))

# Second Convolutional Block model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

# Fully Connected Layers model.add(Flatten())

model.add(Dense(128, activation='relu')) model.add(Dense(1, activation='sigmoid')) # Binary output

# STEP 3: Compile the Model

model.compile( optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy']

)

# STEP 4: Train the Model

history = model.fit( train\_generator, epochs=10,

validation\_data=test\_generator

)

# STEP 5: Visualize Accuracy and Loss Curves plt.figure(figsize=(12, 4))

# Accuracy Plot plt.subplot(1, 2, 1)

plt.plot(history.history['accuracy'], label='Training Accuracy', marker='o') plt.plot(history.history['val\_accuracy'], label='Validation Accuracy', marker='x') plt.title('Accuracy over Epochs')

plt.xlabel('Epoch') plt.ylabel('Accuracy') plt.legend()

# Loss Plot plt.subplot(1, 2, 2)

plt.plot(history.history['loss'], label='Training Loss', marker='o') plt.plot(history.history['val\_loss'], label='Validation Loss', marker='x') plt.title('Loss over Epochs')

plt.xlabel('Epoch') plt.ylabel('Loss') plt.legend()

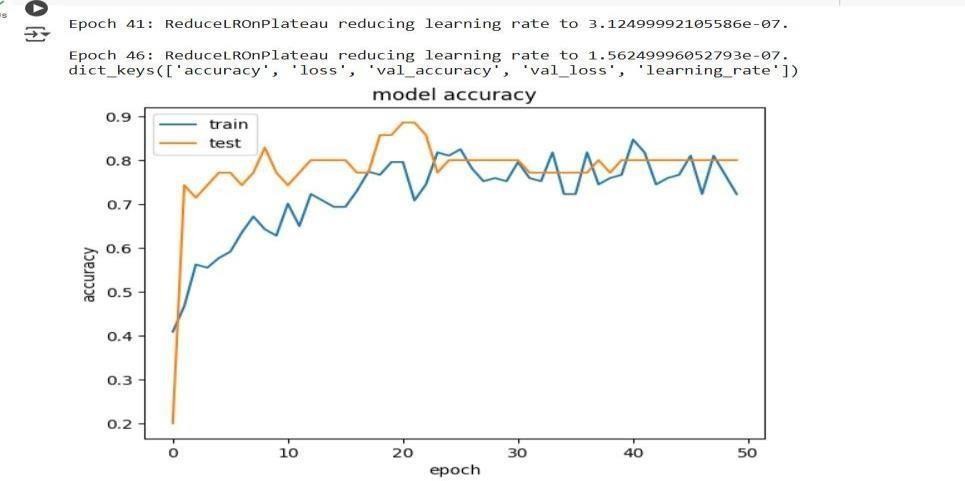
plt.tight\_layout() plt.show()

# STEP 6: Save the Trained Model model.save('cnn\_skin\_cancer\_model.h5')

### APPENDIX B – OUTCOME SCREENSHOTS

****

Fig 6.1 Image classification into malignant and benign



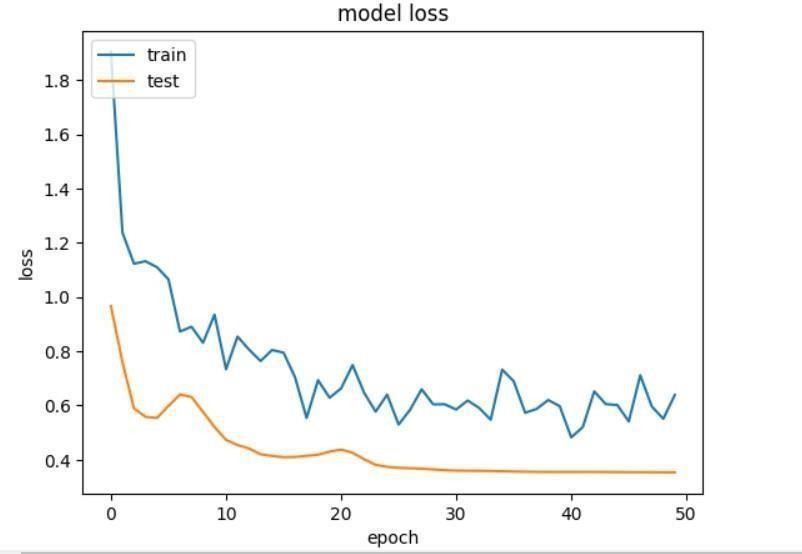
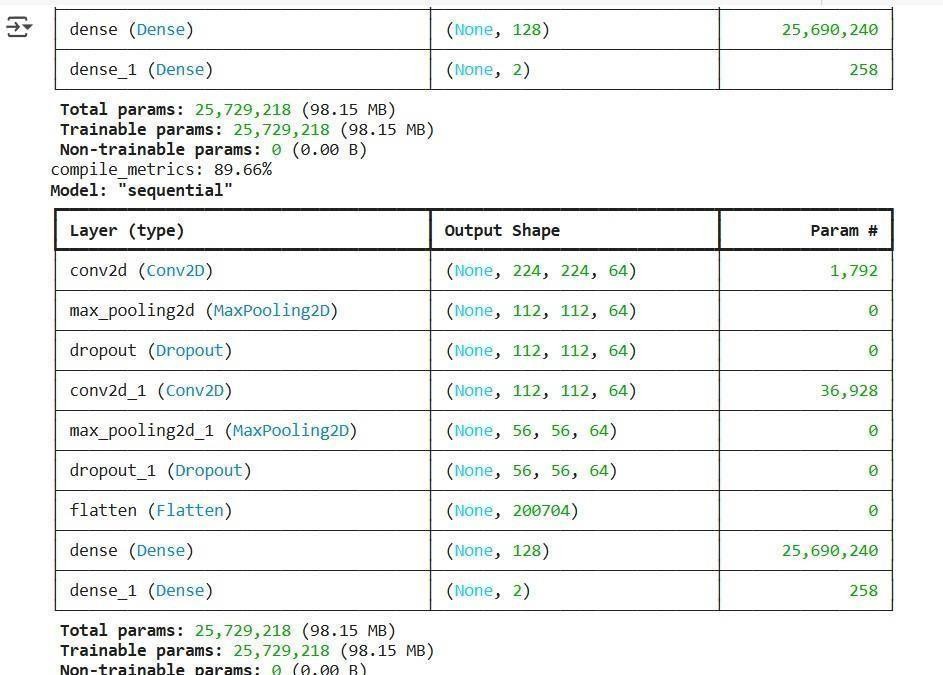
****

Fig 6.2 Model Accuracy and Model Loss

****

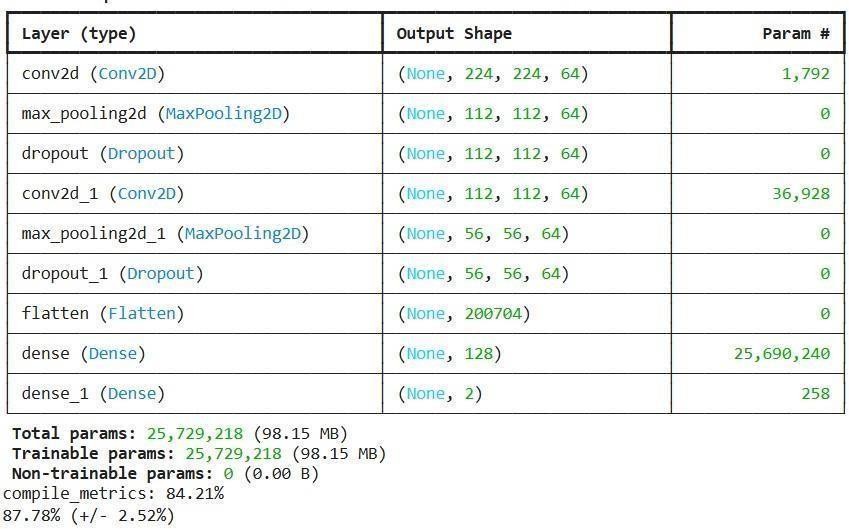


Fig 6.3 Trainable-parameters and Non-Trainable parameters

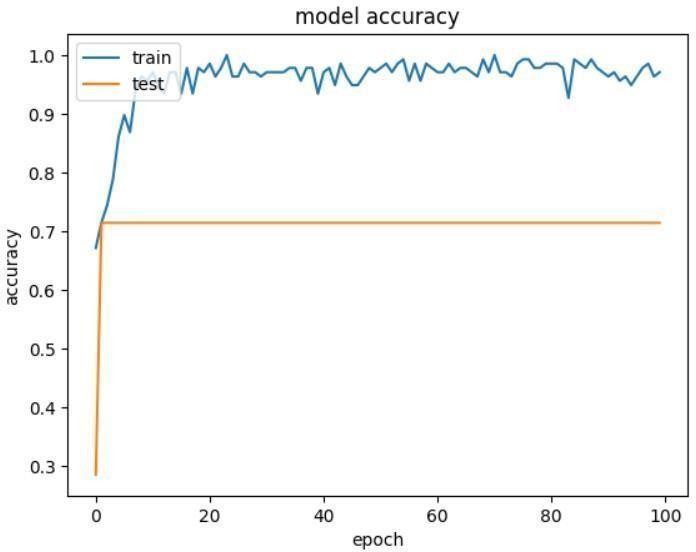
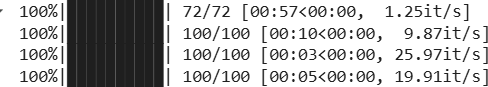
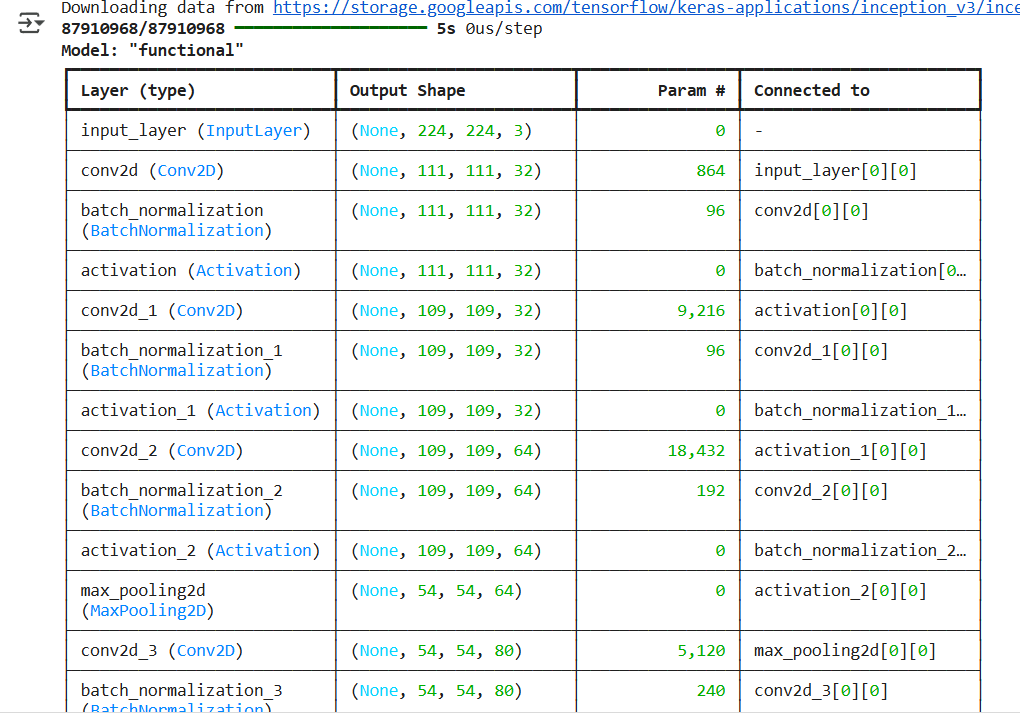
****

Fig 6.4 Model Accuracy

****



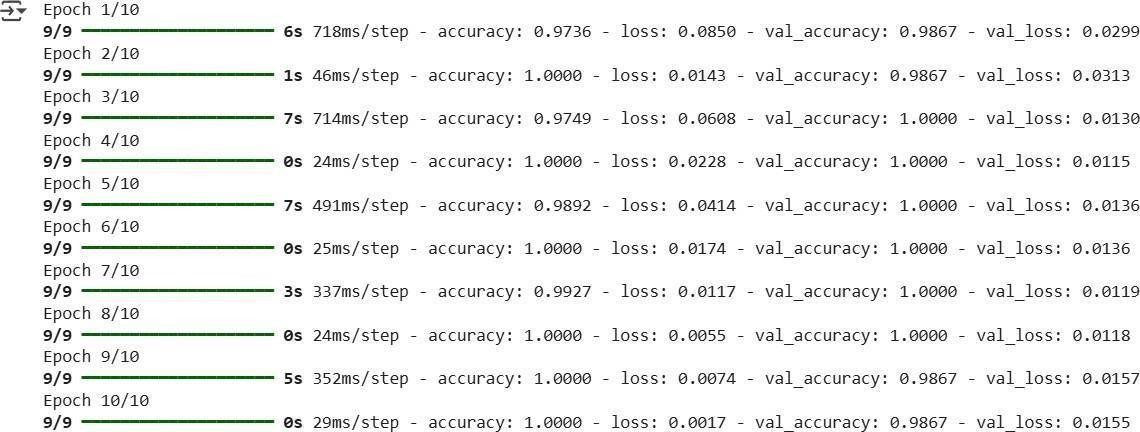
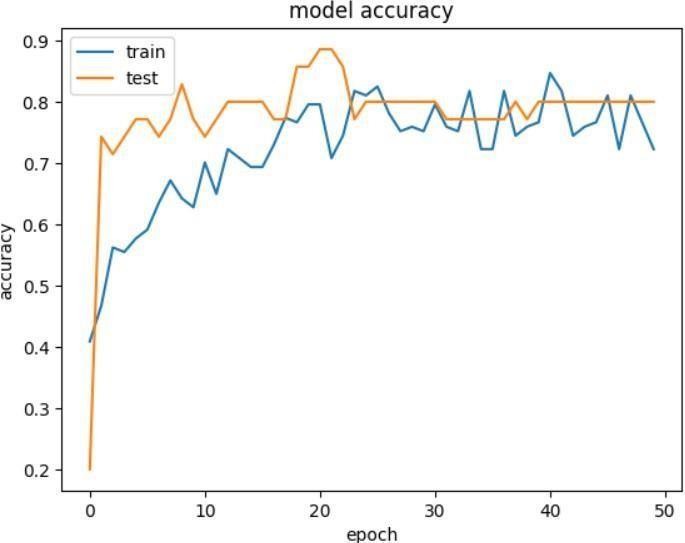
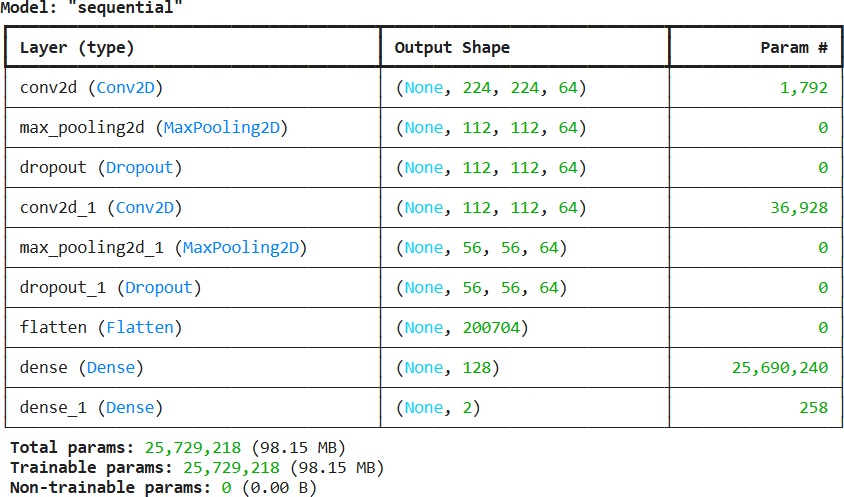


Fig 6.5 Epoch step

****

Fig 6.6 Final test accuracy

**** Fig 6.7 Model -1 sequential

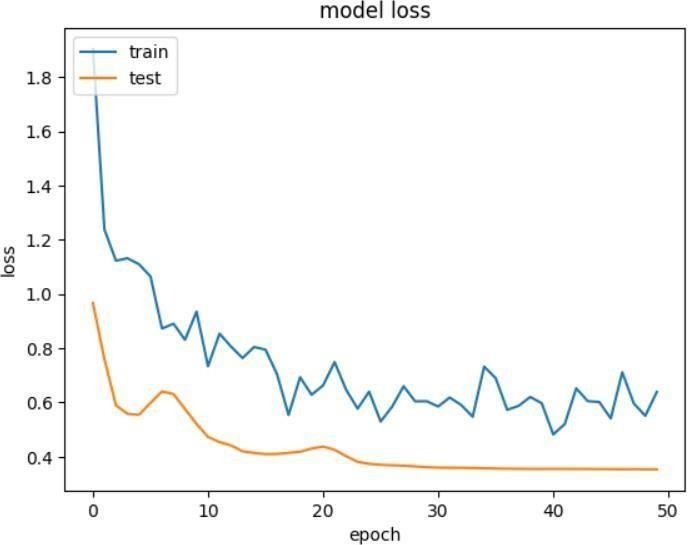
****

Fig 6.7 Model Accuracy and Model Loss

**REFERENCES**

1. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115-118. https://doi.org/10.1038/nature21056
2. Tschandl, P., Rinner, C., Apalla, Z., Kallergi, M., Mielich, J., & Stepanov, A. (2020). Evaluation of deep learning models for skin cancer classification in clinical practice. *The Lancet Digital Health*, 2(12), e686-e692. https://doi.org/10.1016/S2589-7500(20)30288-1
3. Haenssle, H. A., Fink, C., Schneiderbauer, R., Toberer, F., & Kalloo, R. (2018). Artificial intelligence in dermatology: Deep learning for skin cancer detection. *The Journal of Investigative Dermatology*, 138(11), 2329-2336. https://doi.org/10.1016/j.jid.2018.06.117
4. Codella, N. C. F., Zou, K. H., & Lee, H. (2017). Deep learning for skin cancer classification: A survey. *IEEE Transactions on Medical Imaging*, 36(7), 1344-1358. https://doi.org/10.1109/TMI.2017.2678452
5. Sharma, R., Bansal, A., & Gupta, A. (2020). Skin cancer detection using convolutional neural networks: A comprehensive review. *Journal of Medical Systems*, 44(6), 1-9. https://doi.org/10.1007/s10916-020-01659-x
6. Singh, M., & Goyal, N. (2019). Skin cancer detection using deep learning: A review. *IEEE Access*, 7, 126788-126801. https://doi.org/10.1109/ACCESS.2019.2943587
7. Li, Y., & Zhang, X. (2021). A survey on automated skin cancer detection using deep learning techniques. *IEEE Access*, 9, 27224-27239. https://doi.org/10.1109/ACCESS.2021.3057451
8. Liao, W., & Xie, Z. (2020). Skin cancer detection using deep learning techniques: A comprehensive review. *Medical Image Analysis*, 65, 101738.

https://doi.org/10.1016/j.media.2020.101738

1. ISIC Archive. (2022). International Skin Imaging Collaboration (ISIC) Archive. [https://www.isic-archive.com](https://www.isic-archive.com/)
2. Codella, N. C. F., & Pankanti, S. (2020). Using deep learning to classify skin lesions: A review of current methods and data. *IEEE Transactions on Biomedical Engineering*, 67(4), 1169-1180. https://doi.org/10.1109/TBME.2019.2946025