RSET
RAJAGIRI SCHOOL OF
ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

Project Report on

Skin Cancer Detection using ML and Explainable AI

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Renu Lijo (u2103172)
Rhea John Kandathil (u2103174)
Rohan Jojo (u2103179)
Tom Rajeev Thomas (u2103211)

Under the guidance of

Ms. Liya Joseph

Department of Computer Science and Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
April 2025

CERTIFICATE

This is to certify that the project report entitled ***"Skin Cancer Detection using ML and EXPLAINABLE AI "*** *is a bonafide record of the work done by* ***Renu Lijo (u2103172) Rhea John Kandathil (u2103174) Rohan Jojo (u2103179) Tom Rajeev Thomas (u2103211)*** *, submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in "Computer Science and Engineering" during the academic year 2021-2025.*

Ms.Liya Joseph
Project Guide
Assistant Professor
Dept. of CSE
RSET

Ms.Sangeetha Jamal
Project Coordinator
Assistant Professor
Dept. of CSE
RSET

Dr.Preetha KG
Professor & HOD
Dept. of CSE
RSET

ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Rev.Dr.Jaison Paul Mulerikkal CMI,** Principal of RSET, and **Dr.Preetha K.G**, Head of the Department of Computer Science and Engineering for providing us with the opportunity to undertake our project, Skin Cancer Detection using ML and Explainable AI.

We are highly indebted to our project coordinators, **Dr.Preetha K.G**, HOD & Professor,Computer Science and Engineering,**Ms.Sangeetha Jamal**,Assistant Professor,Computer Science and Engineering for their valuable support.

It is indeed my pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Ms.Liya Joseph**,Assistant Professor,Computer Science and Engineering for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, we would like to express my sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Renu Lijo

Rhea John Kandathil

Rohan Jojo

Tom Rajeev Thomas

Abstract

This project focuses on the development of an intricate user interface for skin cancer detection that leverages machine learning (ML) algorithms and Explainable AI (XAI) to offer accurate and interpretable predictions. The system aims to improve early skin cancer detection by analyzing skin lesion images and providing users with reliable insights into potential cancerous patterns.Key features of the skin cancer detection system include: CNN-Based Prediction: The system employs the CNN (Convolutional Neural Networks) deep learning model to assess the likelihood of skin cancer based on analyzed images. CNNs are particularly well-suited for image analysis due to their ability to automatically detect and learn spatial hierarchies and features such as edges, textures, and patterns. The model is trained on extensive datasets, allowing it to capture variations in skin lesions, making it highly effective at differentiating between benign and malignant cases. By leveraging these capabilities, the system delivers accurate and robust predictions, significantly enhancing the effectiveness of early detection and timely intervention in skin cancer diagnosis. Explainable AI (XAI): To build user trust and ensure transparency, the system integrates Explainable AI techniques such as LIME, SHAP and GradCAM. These tools help break down the complex decision-making process of deep learning models, offering clear and understandable explanations for each prediction. By visualizing the areas of the image or features that most influenced the model's decision, users can better comprehend the reasoning behind the assessments. This not only increases confidence in the system's predictions but also ensures accountability, which is crucial for sensitive applications like medical diagnosis. Real-Time Analysis and Feedback: The application supports real-time image analysis, offering immediate feedback and predictions to users. This feature is crucial for timely detection and intervention, allowing users to take prompt action based on the system's recommendations. Overall, the skin cancer detection system offers a powerful tool for early diagnosis by combining cutting-edge image analysis,

predictive algorithms, and transparent AI explanations. This integrated approach aims to enhance the reliability and user-friendliness of skin cancer detection, supporting better health outcomes and informed decision-making.

Contents

List of Abbreviations

- **ML** - Machine Learning
- **XAI** - Explainable AI
- **LIME** - Local Interpretable Model-agnostic Explanations
- **SHAP** - SHapley Additive exPlanations
- **CNN** - Convolutional Neural Network

List of Tables

Chapter 1

Introduction

1.1 Background

Skin cancer, a leading cancer globally, has continued to increase over the years. Early diagnosis is vital to enhance survival because the disease is easy to treat as long as the condition is recognized early. Skin cancer, however, poses a challenge in terms of diagnosis since some skin lesions present themselves as non-malignant conditions. Traditional approaches typically rely on clinicians to visually diagnose skin lesions prior to when they will biopsies them to confirm their diagnosis. As much as the practices are good, they would be time-wasting and unaffordable in remote or less developed areas. Explainable artificial intelligence (XAI) and machine learning (ML) are a cost-effective solution to such limitations. Machine learning algorithms, with advanced computational models, have the capability of detecting potentially cancerous image patterns in images of skin lesions. Convolutional Neural Networks (CNNs), one of the deep technology learns, has proved to be extremely promising in picture recognition, especially in medical images. CNNs are capable of detecting subtle differences between benign and malignant skin lesions through hierarchical picture pattern examination systematically. While CNN-based models are accurate, sometimes they can also be "black boxes" and not readily enable end-users to comprehend how exactly they arrive at their conclusions. Such a transparency issue can become vexing, especially in crucial areas like medical diagnosis that necessitate responsibility and trustworthiness. To assist in the resolution of this problem, Explainable AI (XAI) methods such as GradCAM, SHAP, and LIME can be incorporated into the design. These methods provide transparent, understandable explanations why models make decisions between features from areas of an image highlighted or features used in making predictions. This helps users like healthcare professionals comprehend the reasons why certain diagnoses are being made, hence verifying the system and how it is being applied in actual situations. Real-time analysis of skin image data is also possible with the technology, and this immediately gives feedback and predictions, which is critical for early action so that users can react if necessary. Combination of real-time skin cancer diagnostic technology, high-performance machine learning algorithms, and explainable explanation is crucial to early diagnosis and decision-making. It will drastically improve diagnostic precision, reduce healthcare professionals' workload, and end up benefiting the patients.

1.2 Problem Definition

The objective of this project is to build an enhanced system for diagnosing skin cancer that encompasses utilizing ML algorithms, i.e., CNNs, in systematically examining skin lesion images and assessing the likelihood of skin cancer. Through the use of XAI methods, the approach provides explainability and transparency such that people can comprehend the justification for each prediction, which in the long run allows for the early detection and accurate diagnosis of skin cancer.

1.3 Scope and Motivation

1.3.1 Scope:

This project seeks to create a platform for skin cancer diagnosis by improving the validity and transparency of skin cancer assessment through Explainable AI and machine learning algorithms. Skin lesions captured on this platform are expected to offer real-time predictions for the identification of possible cancers that can be used towards early diagnosis. The goal is to provide the diagnosis with crisp, easy-to-understand descriptions, combined with correct predictions using XAI methods on CNN models like LIME, SHAP, and GradCAM. The device has been developed for doctors who will first hear it; all who need a user-friendly accessible device to identify skin cancer can use the device. Retrieving transparency is a critical component of AI-based medical technology, and increasing detection rates.

1.3.2 Motivation

There are several strong motivations for such projects; increasing incidence of skin cancer worldwide, to be in a position to offer cost-effective and effective detection using inexpensive solutions, and visual inspections that involve biopsies and take huge amounts of professional service time away from the patient, taking away the advantage of early diagnosis. Explainable AI is solving the fundamental trust question in AI systems, especially in the healthcare field, while machine learning is used to develop potential for automation and accuracy in the diagnosis of skin cancer. The primary objective of this project is to integrate the two technologies in order to develop a system that will provide potential for early diagnosis, which will eventually improve treatment outcomes and save more lives.

1.4 Objectives

1. To create a system that reliably identifies skin cancer by consistently analyzing images of skin lesions through machine learning techniques, particularly Convolutional Neural Networks (CNNs).

2. To integrate Explainable AI (XAI) techniques such as LIME, SHAP, and GradCAM to provide users with clear and comprehensible predictions.

3. To perform immediate analysis and deliver feedback for rapid skin cancer detection, enabling prompt actions and decisions.

4. To improve the system's ability to distinguish between benign and malignant cases by training the model with a substantial dataset of skin lesion images.

5. To ensure the system's usability and accessibility for both healthcare professionals and patients, offering a user-friendly interface for effortless interaction.

1.5 Challenges

This project faces a major obstacle in acquiring a varied, high-quality assortment of skin lesion images necessary for effectively training the machine learning models. Keeping clarity and comprehension of deep learning models, which are inherently complex, poses challenges with Explainable AI techniques. An additional critical challenge is to achieve real-time processing capability for accurate predictions while maintaining the dependability and effectiveness of the systems.

1.6 Assumptions

1. The system assumes that users provide high-quality, clear images of skin lesions for accurate analysis.
2. The model assumes that the training dataset contains a adequately diverse set of skin lesion images to generalize well across various skin types and conditions.
3. It is assumed that users will engage with the system for early detection instead of using it as a replacement for professional medical diagnosis.
4. The system assumes that the explanations provided by the Explainable AI techniques (LIME, SHAP, GradCAM) will be understandable to users who do not possess a strong technical background.

1.7 Societal / Industrial Relevance

Both the community and the healthcare industry will gain significant benefits from this initiative. It supplies society with an important resource for the early identification of skin cancer, which can improve health outcomes and potentially save lives through timely intervention. Individuals residing in underserved areas with limited access to dermatological expertise will especially reap the rewards of this system. In the healthcare domain, the initiative responds to the growing demand for AI-powered diagnostic tools that support healthcare professionals in delivering quicker and more precise diagnoses. It also corresponds with the rising trend of employing Explainable AI (XAI) to bolster confidence in automated systems, ensuring transparency and responsibility in medical decision-making. By merging machine learning with skin cancer detection, the initiative contributes to the wider field of medical imaging and AI-driven diagnostic solutions, encouraging innovation within the industry.

1.8 Organization of the Report

The report is organized as follows:

- **Chapter 1: Introduction** - Provides a brief overview of the project, including its context, objectives, and significance in the field of skin cancer detection through machine learning and Explainable AI.

- **Chapter 2: Literature Review** - Reviews the existing research and technologies related to skin cancer detection, machine learning, and Explainable AI, highlighting their relevance to the current study.

- **Chapter 3: Methodology** - Describes the approach, methods, and tools used in the project, featuring the CNN model, Explainable AI techniques, and information about the dataset.

- **Chapter 4: Implementation** - Details the practical aspects of the project, including the system design, development workflow, and the integration of machine learning with Explainable AI.

- **Chapter 5: Results and Evaluation** - Presents the results of the system's effectiveness, including evaluation metrics such as accuracy, precision, recall, and F1-score, along with a discussion of the findings.

- **Chapter 6: Conclusion and Future Work** - Summarizes the key findings of the project, its limitations, and suggests potential directions for future improvements and investigations.

1.9 Conclusion

This chapter outlines the primary findings and outcomes of the project. The skin cancer detection system developed using machine learning and Explainable AI has demonstrated significant potential in accurately identifying skin cancer from lesion images. By employing Convolutional Neural Networks (CNNs) for image analysis and integrating Explainable AI approaches such as LIME, SHAP, and GradCAM, the system not only provides accurate predictions but also promotes transparency and trust in the decision-making process. The project effectively tackles the challenge of early detection and clarity in medical diagnoses by offering a tool that supports healthcare professionals and individuals in identifying possible cases of skin cancer. There are still obstacles to address in the current results, including expanding the dataset and refining the real-time processing abilities. Future efforts will concentrate on enhancing the model's generalization, integrating more advanced Explainable AI methods, and conducting further research in real-time diagnostics to improve health outcomes and support evidence-based decision-making.

Chapter 2

Literature Survey

This chapter offers an overview of different research papers related to the categorization of skin cancer using machine learning and explainable AI methods. The objectives of the research concentrate on improving the accuracy and comprehension of skin lesion and cancer diagnoses.

2.1 Skin Cancer Classification Using XAI

This section investigates articles that employ explainable artificial intelligence (XAI) techniques for skin cancer classification, highlighting the enhancement of accuracy and clarity.

2.1.1 T. Khater et al. [1] - Skin Cancer Classification Using Explainable Artificial Intelligence on Pre-extracted Image Features (2023)

This research improves the detection of skin cancer by using highly precise models that integrate XAI methods applied to pre-obtained image attributes. The goal is to increase both classification effectiveness and clarity. [1]

2.1.2 S. Wang et al. [2] - Interpretability-Based Multimodal Convolutional Neural Networks for Skin Lesion Diagnosis (2023)

This study suggests a multimodal convolutional neural network (CNN) that merges images and metadata, integrating interpretable components like GradCAM and SHAP, to enhance the precision and clarity of skin lesion diagnoses.. [2]

2.1.3 Sarthak Gupta et al. [3] - Skin Lesion Classification Based on Various Machine Learning Models Explained by Explainable Artificial Intelligence (2023)

This study employs a combination of CNN and XGBoost models, enhanced by XAI techniques (LIME and SHAP), to ensure that the model's predictions are both accurate and interpretable. [3]

2.1.4 Bhuvaneshwari Shetty et al. [4] - Skin Lesion Classification of Dermoscopic Images Using Machine Learning and Convolutional Neural Networks (2022)

This study employs CNNs along with data augmentation and k-fold cross-validation on the HAM10000 dataset to create a scalable and automated approach for classifying skin lesions.[4]

2.1.5 Iván Matas et al. [5] - AI-Driven Skin Cancer Diagnosis: Grad-CAM and Expert Annotations for Enhanced Interpretability (2024)

This research combines MobileNet-V2 with Grad-CAM and specialist annotations to enhance the interpretability and precision of diagnosing basal cell carcinoma (BCC).[5]

2.2 Summary and Gaps Identified

This section highlights the main findings from the literature reviewed and points out the gaps that must be tackled in upcoming research. The studies analyzed show significant advancements in the detection and comprehension of skin cancer; however, there are several areas that require further enhancement.

2.2.1 Summary

Below is a table that outlines the pros and cons of each study examined:

Study Title	Advantages	Disadvantages
T. Khater [1]	High classification accuracy, Aligns with clinical knowledge, Multiclass classification	Limited dataset, Lack of real-world evaluation
S. Wang [2]	Combines metadata and images, GradCAM for visual interpretability, SHAP for feature importance	Computationally demanding, Extensive data preparation required
Sarthak Gupta [3]	Uses data augmentation, High accuracy	Limited dataset variation, Potential overfitting
Bhuvaneshwari Shetty [4]	Scalable approach, Automation features	High dependency on image quality, Resource intensive
Iván Matas [5]	Resource efficient, Focus on doctor-relevant features, GradCAM for visual explainability	Struggles with identifying some BCC regions, Limited to BCC regions, Risk of false positives/negatives

Table 2.1: Summary of Key Studies on Skin Cancer Classification

2.2.2 Gaps Identified

The subsequent gaps have been recognized in the existing studies regarding skin cancer classification and interpretation:

1. **Limited Dataset Variability:**Numerous studies utilize small or relatively uniform datasets, which restricts the models' generalization to a range of real-world

situations.

2. **Lack of Real-World Validation:**Several proposed models do not have genuine clinical validation in practical environments, which reduces their effectiveness in practice.

3. **High Computational Demands:**The models frequently need significant computational resources, which may restrict their application in clinical contexts where resources are limited.

4. **Handling False Positives and Negatives:**In spite of progress, the models find it challenging to accurately detect in specific areas (e. g. , basal cell carcinoma), resulting in an increased risk of false positives and negatives.

5. **Interpretability in Complex Cases:**Although XAI techniques such as Grad-CAM and SHAP enhance interpretability, there are still challenges in offering clear, actionable insights for complex or unclear situations that clinicians may encounter.

Chapter 3

Requirements

3.1 Hardware Requirements

Processor: A high-performance processor, such as Intel Core i5/i7 or AMD Ryzen 5/7, ensures efficient data processing and model training. Faster processors with more cores improve parallel computation, speeding up deep learning tasks. **RAM:** At least 16GB RAM is required to handle large datasets and deep learning models without memory bottlenecks. Higher RAM (32GB or more) enhances multitasking and improves performance during model training. **GPU:** A CUDA-enabled GPU, like NVIDIA GTX 1650 (minimum) or RTX 3090 (recommended), accelerates matrix computations and deep learning tasks. More VRAM (8GB or higher) is crucial for training complex models efficiently. **Storage:**An SSD (minimum 512GB, recommended 1TB) ensures fast read/write speeds, improving dataset loading and model checkpoint saving. NVMe SSDs further reduce data transfer latency, enhancing training speed. **Other:**High-speed internet is necessary for cloud-based training on platforms like Google Colab . Proper cooling systems prevent overheating during prolonged GPU-intensive training sessions.

3.2 Software Requirements

- **Programming Languages:** Python is the primary language for deep learning model development and backend processing, while HTML and CSS are used for the frontend interface, ensuring a user-friendly web application.
- **Deep Learning Frameworks:** Keras, built on TensorFlow, is used for designing, training, and fine-tuning models like CNN, ResNet, Inception, and Xception.
- **Web Framework:** Flask is used to deploy the trained models as a web application, enabling real-time skin cancer detection through API-based interactions between the

frontend and backend.

- **Other Dependencies**: Essential libraries such as NumPy (numerical computations), Pandas (data handling), OpenCV (image processing) and explainability tools like Grad-CAM, LIME, and SHAP enhance model interpretability and usability

Chapter 4

System Architecture

This chapter offers an overview of the system architecture, describes each component of the system, outlines the algorithms applied at different phases along with the data flow diagram, the tools and technologies employed, the dataset selected for training the detection models, and the primary deliverables. Ultimately, this chapter specifies the module divisions and work breakdown together with the timeline required to finalize the project.

4.1 System Overview

The design of the system is illustrated below:

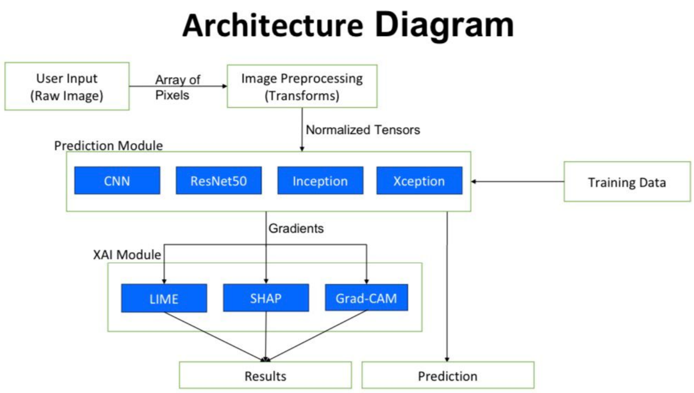


Figure 4.1: System Architecture

1. **User Input (Raw Image)** The system starts with a user-provided raw image in common formats such as JPEG or PNG.

2. **Array of Pixels → Image Preprocessing (Transforms):** The raw image is converted into an array of pixel values. This array undergoes preprocessing steps

such as resizing, normalization, or data augmentation. The output is a normalized tensor, suitable for input into deep learning models.

3. **Prediction Module:**

   (a) **CNN (Convolutional Neural Network):** A CNN examines the raw image to recognize significant features, including textures, shapes, and edges, which are crucial for making predictions.

   (b) **ResNet50**: A residual network with 50 layers designed to combat the vanishing gradient problem.

   (c) **Inception**: A model that uses different filter sizes in parallel to extract features.

   (d) **Xception**: A deep CNN that uses depthwise separable convolutions to improve efficiency.

XAI Module:

1. **Metadata:**Additional details, like the patient's age and the region impacted, are provided to assist the explanation in becoming more context-sensitive.

2. **LIME:**Modifies the depiction and estimates a more straightforward model to clarify the prediction.

3. **SHAP:** Assigns specific input attributes to the model's output to recognize important elements in the prediction.

4. **Grad-CAM:**Highlights the areas of the image that are most significant to the CNN's recognition.

Training Data:A categorized collection of medical images is utilized to train the CNN for predicting conditions.

Results: The XAI module provides the following:

1. **Feature Importance:**Recognizing the most impactful features of the prediction.

2. **Visual Explanations:**Highlighting key areas of the image to enhance clarity for medical professionals.

4.2 Component Design

The creation of the skin cancer detection system includes four primary elements, which are described below.

4.2.1 Input from the Image Source

The procedure starts by obtaining images of skin lesions through the interface of the system. Users either upload the input images or they are taken using attached imaging devices. The system guarantees compatibility with conventional image formats (e. g. , JPEG, PNG) and upholds adequate resolution for precise analysis.

4.2.2 Input Pre-processing

To maintain the quality and uniformity of the input data, the unprocessed images go through pre-processing stages, consisting of:

- **Image resizing:**Standardizes each image to a uniform resolution to permit efficient analysis by the machine learning model.
- **Noise reduction:** Applies filters to eliminate artifacts and improve the clarity of lesion patterns.
- **Contrast enhancement:** Improves the visibility of subtle details by adjusting image brightness and contrast settings.
- **Segmentation:** Identifies and distinguishes the lesion from the surrounding skin to focus the examination on the region of interest.

4.2.3 Feature Extraction and Enhancement

To get the images ready for ML analysis, the system gathers and improves pertinent features. The steps encompass:

- **Feature extraction via CNN:** Employs convolutional layers to identify spatial hierarchies within the images, recognizing patterns like edges, textures, and anomalies.

- **Normalization:** Scales pixel intensity values to a uniform range, enhancing the model's performance under different lighting conditions.

4.2.4 Machine Learning Models

Convolutional Neural Network (CNN):

- Analyzes and recognizes intricate spatial patterns in lesion images.
- Differentiates between benign and malignant cases based on the features obtained.
- Trained on various datasets to ensure accurate and broadly applicable predictions.

Residual Neural Network (ResNet):

Improves feature extraction by implementing deep residual learning to address vanishing gradient problems.

Enhances classification accuracy by effectively identifying hierarchical patterns in lesion images.

Trained on a range of datasets to guarantee strong generalization and dependable skin cancer detection.

Inception Model:

- Employs multiple convolutional filters of varying sizes to capture both intricate and broad details in lesion images.
- Effectively minimizes computational expenses while retaining high accuracy through factorized convolutions.
- Improves feature extraction by simultaneously processing spatial and channel-wise information, enhancing skin cancer classification.

Explainable AI (XAI) Techniques:

- Employs methods such as LIME (Local Interpretable Model-Agnostic Explanations), SHAP (SHapley Additive exPlanations), and Grad-CAM (Gradient-weighted Class Activation Mapping).

- Provides visual representations that highlight the parts of the image that had the most significant influence on the model's decision, promoting user trust and accountability.

4.2.5 Feedback and User Interaction

- **Real-time predictions:** Delivers instant analysis and feedback, promoting swift intervention and decision-making.
- **Interactive explanations:** Offers visual overlays and textual descriptions, helping users grasp the reasoning behind predictions.
- **Report generation:** Provides detailed summaries of analysis results, including identified patterns, risk evaluations, and recommended next steps.

4.3 Data Flow Diagram

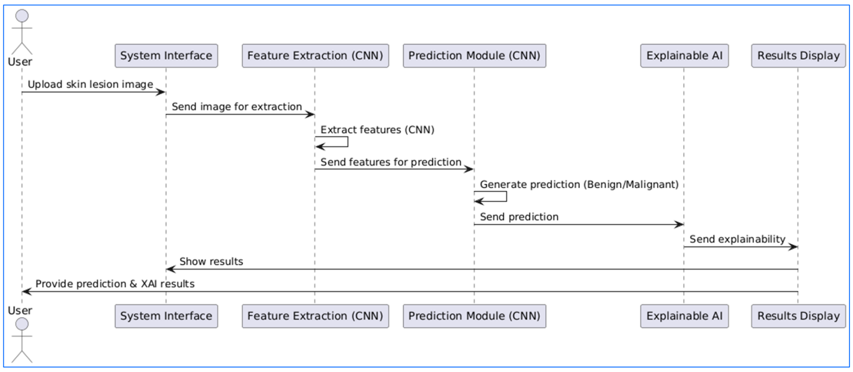


Figure 4.2: Data Flow Diagram

4.4 Tools and Technologies

4.4.1 Hardware Requirements

The subsequent hardware specifications are advised to guarantee effective handling of extensive datasets, immediate analysis, and precise predictions for skin cancer identification utilizing machine learning models:

- **Intel Core i7/i9 CPU**: The Intel Core i7 or i9 CPU will provide the necessary computational power for real-time image processing, executing machine learning inference, and handling multiple tasks simultaneously. These processors are highly efficient at managing complex data processing tasks, like running Convolutional Neural Networks (CNNs) and generating predictions from high-resolution skin lesion images.

- **NVIDIA GeForce RTX GPU**: An NVIDIA GeForce RTX series GPU (such as the RTX 3060, 3070, 3080) is essential for enhancing the inference of deep learning models. The GPU will manage parallel processing functions, including the convolutional processes in CNNs, which facilitates quicker processing of skin lesion images for prediction and explanation generation (like GradCAM visualizations). This hardware is required for minimizing the duration needed for image analysis.
- **16GB/32GB RAM**: With 16GB to 32GB of RAM, the system will possess the ability to manage large image datasets and ensure smooth functionality of machine learning models.
- **Windows 11, 64-bit OS**:A 64-bit edition of Windows 11 is suggested for this project. The 64-bit framework permits improved memory handling and alignment with contemporary development tools and libraries, including React (for frontend work), Node. js (for backend operations), and PyTorch (for deep learning). Windows 11 provides superior performance and compatibility with the newest tools vital for machine learning and web development.
- **Storage Requirements (SSD, 512GB - 1TB)**:A Solid State Drive (SSD) with a storage size between a minimum of 512GB and 1TB is suggested for rapid data access and effective management of extensive image datasets and model files. SSDs provide considerably quicker data access speeds in comparison to conventional Hard Disk Drives (HDDs), guaranteeing fast loading times throughout training, inference, and real-time processing.

4.4.2 Software Requirements

Frontend Development:

- React: A JavaScript library intended for developing dynamic and responsive user interfaces. React will be used to construct the frontend of the skin cancer detection system, enabling image uploads, immediate feedback, and visual explanations.

- Bootstrap / Material-UI: Frontend frameworks used for styling and creating responsive layouts. These will help in developing an interface that is user-friendly for uploading images and displaying predictions and explanations..

Backend Development:

- Python: Python will act as the coding language for the machine learning model. PyTorch, a widely recognized deep learning framework, will be employed for the classification of skin cancer using pre-trained Convolutional Neural Networks (CNNs).

- Flask / FastAPI (for Python backend): Either Flask or FastAPI will function as the framework to develop a RESTful API for interaction between the Node. js backend and the machine learning model. This is the point at which both image processing and model inference will take place..

- PyTorch: A deep learning framework designed for creating and deploying the skin cancer detection model. PyTorch supplies the resources necessary for training the CNN and performing inference on the images that have been uploaded..

4.5 Dataset Identified

The suggested manuscript centers on a deep learning approach for the automated diagnosis and classification of various types of skin cancer utilizing the ISIC archive and SIIM-ISIC dataset, which contains a wide array of annotated dermoscopic images. The foundation of this study is a CNN model, where the idea of categorizing skin lesions primarily hinges on their determination as benign or malignant, with a key focus on detecting melanoma; the dataset includes over 10,000 images.

The model will undergo training and assessment using a large volume of images that showcase different categories of skin lesions. The preprocessing techniques, which encompass image normalization and augmentation, will also be analyzed to enhance the model's accuracy and strength. Moreover, Explainable AI (XAI) methods like GradCAM will be used to provide interpretable visual explanations of the predictions generated by the model, thereby making the system more transparent and dependable for clinical uses.

4.6 Module Division and Work Breakdown

4.6.1 Module Division

Dataset Creation

- Functionality: Gather a variety of skin lesion images (including both benign and malignant) from different sources, making sure to include diversity in skin types, lesion characteristics, and environmental conditions.
- Tasks: Gather and label images with metadata, making sure to include a variety of demographics, lighting conditions, and lesion types.

Image Preprocessing

- Functionality: Standardize and improve raw images for consistent input.
- Tasks: Transform images to a consistent format, implement noise and silence elimination, adjust size, normalize, and segment lesions.

Feature Extraction and Enhancement

- Functionality: Extract and enhance features from images to aid in accurate classification.
- Tasks: Extract texture, color, and shape features; apply feature normalization; enhance deep features using CNNs.

SkinCancer Detection Model

- Functionality: Classify skin lesions as benign or malignant using machine learning models (CNNs).
- Tasks: Train CNN models, apply transfer learning, and evaluate model performance.

Explainability and Transparency (XAI)

- Functionality: Utilize XAI techniques to deliver clear model predictions.

- Tasks: Integrate LIME, SHAP, and GradCAM to explain model decisions and highlight important image features.

Real-Time Prediction and Alert System

- Functionality: Enable real-time skin lesion analysis and alert generation.
- Tasks: Process images instantly, trigger alerts for suspicious lesions, and enable user feedback integration.

User Interface (UI) and Visualization

- Functionality: Offer an engaging interface for users to submit images and observe outcomes..
- Tasks: Provide prediction results with confidence scores and visual explanations, allowing user interaction.

4.6.2 Work Breakdown

Task	**Team Members**
Project Planning	Common
Data Collection	Renu and Rhea
Set Up Environment	Rohan Jojo
Initial CNN Model Design	Rohan Jojo
CNN Training	Common
ResNet Training	Rhea John
Inception Training	Rohan Jojo
Xception Training	Renu Lijo
Model Testing and Validation	Common
Frontend	Rhea and Renu
Backend	Tom
Research paper	Renu and Rhea

4.7 Key Deliverables

The expected system to be delivered is a detection system that:

1. Integration Ability: The skin cancer detection system can be incorporated into existing healthcare frameworks for easy use in clinical settings..

2. Classification Model:A robust model that classifies skin lesions as benign or malignant through image analysis, utilizing Convolutional Neural Networks (CNNs).

3. Real-time Evaluation: The system offers real-time evaluation and feedback regarding skin lesion images to enable prompt action.

4. Clarification Features: To ensure clear diagnostics, Explainable AI (XAI) methods are incorporated to offer brief explanations of the model's decision-making process.

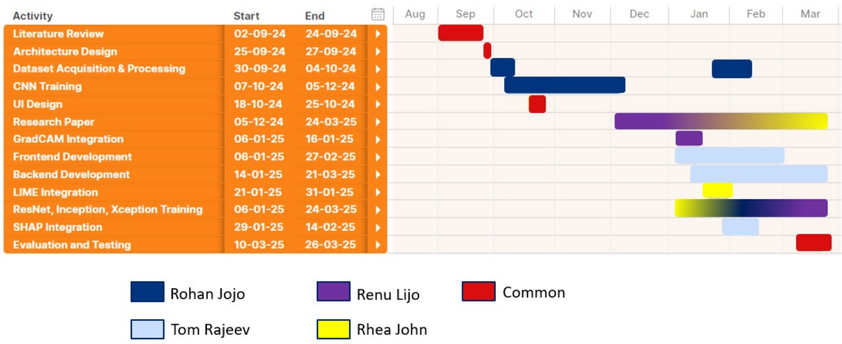


Figure 4.3: Project Timeline

This chapter describes the structure of the system, specifying the various modules and the technologies used in their creation, along with the expected results of the system.

Chapter 5

System Implementation

This chapter thoroughly details the methodology employed in this project and how it was carried out. It describes the operation of the prototype developed as part of this project.

5.1 Datasets Identified

We developed a robust and accurate deep learning model for identifying skin cancer using a number of reliable dermatology datasets. These datasets were carefully selected and merged to ensure a diverse and high-quality training set. By integrating datasets that highlight various lesion types and imaging settings, we aimed to increase model generalization, decrease bias, and improve classification performance. A list of the primary datasets utilized in this study can be seen below.

- **ISIC Archive** – A popular source of labeled images of skin lesions for deep learning and dermatological research is the ISIC Archive.

- **BCV20000 –** By showing a range of skin lesions, this large dataset aids in model generalization.

- **Challenge 2024 –** A prestigious dataset from the most recent ISIC Challenge that includes state-of-the-art dermoscopic images.

- **HAM10000** – 10,000 images of skin lesions divided into seven categories; commonly used to train algorithms for categorization.

- **SIIM-ISIC Melanoma Classification (Kaggle)** –SIIM-ISIC Melanoma Classification (Kaggle) is a dataset of pre-labeled melanoma pictures that improves the identification of malignant patients.

- **Dataset Integration** – To preserve integrity, all datasets were combined while ensuring that only high-quality, annotated images were contributed.

5.2 Proposed Methodology

Data Preprocessing

Data preprocessing is a crucial stage to ensure the quality and consistency of input images used for training. The dataset goes through multiple stages of cleaning, which consist of resizing images to a uniform size, normalizing to standardize pixel values, and applying augmentation techniques like rotation, flipping, and contrast adjustments to improve model generalization. In addition, irrelevant or duplicated images are removed to maintain data integrity. The dataset is then divided into training, validation, and test sets in a proportional manner to prevent model bias.

Feature Extraction

Deep learning models derive spatial and textural characteristics from lesion images. CNN, ResNet, Inception, and Xception frameworks analyze images through convolution, pooling, and activation functions. These layers recognize complex patterns like shape, color variations, and texture differences, enabling models to differentiate between benign and malignant lesions. ResNet and Inception improve feature learning using skip connections and multi-scale analysis.

Classification

After features are extracted, deep learning models categorize skin lesions. Hyperparameter optimization, including adjustments to the learning rate, dropout regularization, and batch normalization, enhances accuracy. A softmax activation function in the last layer calculates probability scores for various lesion classes. A comparative assessment of CNN, ResNet, Inception, and Xception identifies the most effective model.

Explainability and Interpretability

Explainable AI (XAI) methods such as Grad-CAM, LIME, and SHAP promote trust and transparency in model predictions.

Grad-CAM (Gradient-weighted Class Activation Mapping) Grad-CAM emphasizes significant regions of the image by calculating the gradient of the output class score in relation to feature maps:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{5.1}$$

$$L_{Grad-CAM}^c = ReLU \left(\sum_k \alpha_k^c A^k \right) \tag{5.2}$$

where α_k^c indicates the significance of feature map k for class c.

LIME (Local Interpretable Model-agnostic Explanations) LIME approximates the model's decision in a local setting using an interpretable linear model:

$$\hat{f}(x) = \sum_{i=1}^{n} w_i x_i + b \tag{5.3}$$

where w_i denotes the weight attributed to feature i.

SHAP (SHapley Additive Explanations) SHAP values fairly allocate feature contributions based on cooperative game theory:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f(S \cup \{i\}) - f(S)] \tag{5.4}$$

where ϕ_i signifies the SHAP value for feature i.

Web Application Interface

A user-friendly web interface is created using Flask, HTML, CSS, and JavaScript for immediate skin cancer classification. The interface enables users to upload lesion images, view classification outcomes, and access explainability visualizations. The backend guarantees efficient model inference and secure data management.

Performance Evaluation

Models are assessed using accuracy, precision, recall, F1-score, and confusion matrices. Cross-validation provides stability, while ROC-AUC curves identify sensitivity-specificity trade-offs. CNN, ResNet, Inception, and Xception are compared to evaluate generalization and robustness.

5.3 Description of Implementation Strategies

Libraries Used

- **NumPy** NumPy is used for efficient numerical computations, handling multi-dimensional arrays and performing mathematical operations essential for preprocessing and model training.
- **PyTorch** PyTorch provides a flexible deep learning framework with dynamic computation graphs, enabling model development, training, and evaluation with GPU acceleration.
- **SHAP** SHAP (SHapley Additive Explanations) is employed to interpret model predictions by quantifying feature contributions, enhancing model transparency and decision-making.
- **LIME** LIME (Local Interpretable Model-agnostic Explanations) generates locally interpretable models to explain deep learning predictions by approximating decision boundaries.
- **Grad-CAM** Grad-CAM is used for visualizing important regions in input images by computing activation map gradients, improving explainability in CNN-based classification models.
- **Flask** Flask is a lightweight web framework used to develop the application's back-end, enabling image uploads, model inference, and API integrations for real-time predictions.

5.4 Code Snippet

```python
class ConvNet(nn.Module):
    def __init__(self, num_classes=2, dropout=0.5):
        super(ConvNet, self).__init__()

        self.conv1 = nn.Conv2d(in_channels=3, out_channels=16, kernel_size=3, stride=1, padding=1)
        self.bn1 = nn.BatchNorm2d(num_features=16)
        self.relu1 = nn.ReLU()
        self.pool1 = nn.MaxPool2d(kernel_size=2, stride=2)

        self.conv2 = nn.Conv2d(in_channels=16, out_channels=32, kernel_size=3, stride=1, padding=1)
        self.bn2 = nn.BatchNorm2d(num_features=32)
        self.relu2 = nn.ReLU()
        self.pool2 = nn.MaxPool2d(kernel_size=2, stride=2)

        self.conv3 = nn.Conv2d(in_channels=32, out_channels=64, kernel_size=3, stride=1, padding=1)
        self.bn3 = nn.BatchNorm2d(num_features=64)
        self.relu3 = nn.ReLU()
        self.pool3 = nn.MaxPool2d(kernel_size=2, stride=2)

        self.conv4 = nn.Conv2d(in_channels=64, out_channels=128, kernel_size=3, stride=1, padding=1)
        self.bn4 = nn.BatchNorm2d(num_features=128)
        self.relu4 = nn.ReLU()
        self.pool4 = nn.MaxPool2d(kernel_size=2, stride=2)

        self.conv5 = nn.Conv2d(in_channels=128, out_channels=256, kernel_size=3, stride=1, padding=1)
        self.bn5 = nn.BatchNorm2d(num_features=256)
        self.relu5 = nn.ReLU()
        self.pool5 = nn.MaxPool2d(kernel_size=2, stride=2)

        self.conv6 = nn.Conv2d(in_channels=256, out_channels=512, kernel_size=3, stride=1, padding=1)
        self.bn6 = nn.BatchNorm2d(num_features=512)
        self.relu6 = nn.ReLU()
```

Figure 5.1: CNN Model Architecture

```python
self.pool6 = nn.MaxPool2d(kernel_size=2, stride=2)

self.conv7 = nn.Conv2d(in_channels=512, out_channels=512, kernel_size=3, stride=1, padding=1)
self.bn7 = nn.BatchNorm2d(num_features=512)
self.relu7 = nn.ReLU()
self.pool7 = nn.MaxPool2d(kernel_size=2, stride=2)
self.dropout = nn.Dropout(p=dropout)
self.global_average_pooling = nn.AdaptiveAvgPool2d((1, 1))
self.fc = nn.Linear(in_features=512, out_features=num_classes)
```

Figure 5.2: CNN Model Architecture

```python
# Training and evaluation
for epoch in range(start_epoch, num_epochs):
    model.train()
    running_loss = 0.0
    correct_train, total_train = 0, 0

    for inputs, labels in train_loader:
        inputs, labels = inputs.to(device), labels.to(device)
        optimizer.zero_grad()
        outputs = model(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        running_loss += loss.item()
        _, predicted = torch.max(outputs, 1)
        total_train += labels.size(0)
        correct_train += (predicted == labels).sum().item()

    train_accuracy = 100 * correct_train / total_train
    train_accuracies.append(train_accuracy)
```

Figure 5.3: ResNet Model Training

```python
for epoch in range(start_epoch, num_epochs):
    model.train()
    running_loss = 0.0
    correct_train, total_train = 0, 0

    for images, labels in train_loader:
        images, labels = images.to(device), labels.to(device)

        optimizer.zero_grad()
        outputs = model(images)

        if hasattr(model, 'AuxLogits'):
            loss1 = criterion(outputs.logits, labels)
            loss2 = criterion(outputs.aux_logits, labels)
            loss = loss1 + 0.4 * loss2
        else:
            loss = criterion(outputs, labels)

        loss.backward()
        optimizer.step()
        running_loss += loss.item()

        _, predicted = torch.max(outputs.logits if hasattr(model, 'AuxLogits') else outputs, 1)
        total_train += labels.size(0)
        correct_train += (predicted == labels).sum().item()
```

Figure 5.4: Inception Model Training

```python
for epoch in range(start_epoch, num_epochs):
    model.train()
    correct_train, total_train = 0, 0

    for images, labels in train_loader:
        images, labels = images.to(device), labels.to(device)
        optimizer.zero_grad()
        outputs = model(images)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        _, predicted = torch.max(outputs, 1)
        total_train += labels.size(0)
        correct_train += (predicted == labels).sum().item()
```

Figure 5.5: Xception Model Training

```python
# Define a prediction function for LIME
def predict_fn(images):
    batch = torch.stack([transformer(Image.fromarray(image)).to(device) for image in images])
    with torch.no_grad():
        outputs = model(batch)
        probabilities = torch.nn.functional.softmax(outputs, dim=1)
    return probabilities.cpu().numpy()
```

Figure 5.6: LIME

```python
def explain_image(image_path, model):
    # Load and preprocess the image
    image = Image.open(image_path).convert('RGB')
    input_tensor = transformer(image).unsqueeze(0).to(device)

    # Define a SHAP explainer
    background = torch.randn((10, 3, 150, 150)).to(device)  # Random background data
    explainer = shap.DeepExplainer(model, background)

    # Generate SHAP values
    shap_values = explainer.shap_values(input_tensor)

    # Convert SHAP values to numpy arrays
    shap_values = np.array(shap_values[0])  # Use the SHAP values for the first class
    input_image = input_tensor.cpu().numpy().transpose(0, 2, 3, 1)[0]  # Convert to NHWC format
    #print("SHAP values shape:", shap_values.shape)
    shap_image=shap_values[0]
    # Sum across channels to match image dimensions
    shap_sum =np.sum(shap_image, axis=-1)

    # Filter out low SHAP values below threshold
    threshold = 0.0005
    shap_sum[np.abs(shap_sum) < threshold] = 0

    # Lighten the input image for better visibility
    lightened_image = np.clip(input_image * 0.7 + 0.3, 0, 1)  # Increase brightness

    # Plot fused image
    plt.figure(figsize=(8, 6))

    # Show lightened input image
    plt.imshow(lightened_image)
```

Figure 5.7: SHAP

```python
# Global average pooling of gradients
weights = gradients.mean(dim=(2, 3), keepdim=True)

# Weighted sum of activations
gradcam = (weights * activations).sum(dim=1, keepdim=True)
gradcam = F.relu(gradcam)  # Apply ReLU
gradcam = gradcam.squeeze().cpu().numpy()
```

Figure 5.8: Grad-CAM

Chapter 6

Result and Discussion

The performance of CNN, ResNet, Inception, and Xception models was systematically evaluated for skin cancer detection using explainable AI (XAI) techniques. The CNN model achieved a training accuracy of 92

To enhance interpretability and model transparency, we employed XAI techniques, including Grad-CAM, LIME, and SHAP. Grad-CAM provided visual heatmaps to highlight key regions influencing model predictions, aiding in understanding decision boundaries. LIME facilitated localized interpretability by analyzing perturbed input data, offering insights into model sensitivity to feature variations. SHAP quantified feature importance, enabling a comprehensive understanding of how different input factors contribute to classification outcomes.

Additionally, the evaluation included an in-depth analysis of loss convergence trends, learning rate adjustments, and regularization techniques to optimize performance. Hyperparameter tuning, such as batch size modification and dropout implementation, was conducted to mitigate overfitting and enhance generalization. The models were further assessed based on precision, recall, and F1-score to ensure a balanced evaluation of predictive capabilities.

The integration of XAI techniques strengthened the reliability of the models by providing transparent decision-making pathways, ensuring that critical clinical applications maintain high interpretability and trust. The findings underscore the potential of deep learning in automated skin cancer detection while emphasizing the necessity for model explainability to facilitate real-world deployment in medical diagnostics.

Chapter 7

Conclusions & Future Scope

In conclusion, the proposed skin cancer detection system effectively integrates the advantages of machine learning and Explainable AI (XAI) to tackle the difficulties of early diagnosis. Utilizing Convolutional Neural Networks (CNNs), the system achieves high precision in differentiating between benign and malignant skin lesions, while XAI methods ensure clarity and confidence in the decision-making process. It also becomes an extremely practical tool for both healthcare professionals as well as individuals living in underserved regions with the added real-time analysis capability. This solution truly brings into reality the potential of AI for transforming healthcare diagnostics and reducing the burdens that the medical professionals have to face, thereby improving patient outcomes through timely intervention.

The future scope of this project involves expanding the dataset to include more diverse skin types, lesion categories, and demographic diversity to make the model more generalizable and equitable across populations. The system can also be made more robust against low-quality input images through advanced preprocessing methods and data augmentation techniques. This can further increase accessibility through integration with mobile applications and telemedicine platforms in remote areas. The system will further consider the scope of future development with the incorporation of multi-class classification that identifies multiple types of skin diseases besides cancer for comprehensive diagnostic ability. Continuous progress with improvements of this project could help to advance the application of AI in medical images and the detection of diseases in early stages.

References

[1] T. Khater *et al.*, "Skin cancer classification using explainable artificial intelligence on pre-extracted image features," *Journal of Medical Imaging and Health Informatics*, 2023.

[2] S. Wang *et al.*, "Interpretability-based multimodal convolutional neural networks for skin lesion diagnosis," *IEEE Transactions on Medical Imaging*, 2023.

[3] S. Gupta *et al.*, "Skin lesion classification based on various machine learning models explained by explainable artificial intelligence," *IEEE Access*, 2023.

[4] B. Shetty *et al.*, "Skin lesion classification of dermoscopic images using machine learning and convolutional neural networks," *Journal of Healthcare Engineering*, 2022.

[5] I. Matas *et al.*, "Ai-driven skin cancer diagnosis: Grad-cam and expert annotations for enhanced interpretability," *IEEE Transactions on Medical Imaging*, 2024.

Appendix A: Presentation

Skin Cancer Detection using AI/ML Techniques

Final Project Presentation

Group No. 14:
Renu Lijo (U2103172)
Rhea John (U2103174)
Rohan Jojo (U2103179)
Tom Rajeev (U2103211)

Guide:
Ms. Liya Joseph
Assistant Professor
Dept. of CSE
Rajagiri School of
Engineering & Technology

Contents

Problem Statement

Detecting skin cancer early is challenging due to its subtle and complex symptoms, necessitating an automated tool that leverages Machine Learning for accurate diagnosis and Explainable AI to ensure transparency and trust in the decision-making process.

Purpose and Need

- The project aims to create an automated tool to help detect skin cancer early and accurately using Machine Learning. It also uses Explainable AI to make the tool's decisions clear and understandable.

- Diagnosing skin cancer can be difficult because the symptoms are subtle and need expert analysis. There's a need for a tool that can make diagnosis more accurate and easier to understand for doctors.

Objectives

- Create an automated tool for the early and accurate detection of skin cancer using Machine Learning.
- The system will utilize Convolutional Neural Networks (CNNs) to analyze skin images and identify potential cancerous lesions.
- Incorporate Explainable AI (XAI) to ensure transparency in the decision-making process, helping healthcare professionals trust the results and improve patient outcomes.

Literature Review

Paper	Summary	Advantages	Disadvantages
A. Panthakkan [1] **Concatenated Xception-ResNet50 — A novel hybrid approach for accurate skin cancer prediction** (2022)	The study introduces a hybrid deep learning model combining Xception and ResNet50 to improve the accuracy of skin cancer detection. Inception is used as a reference.	+ 97.8% accuracy + Utilizes a simpler architecture with fewer deep layers	- Limited by availability of Dataset - Currently works for 7 types of cancers.
S. Wang [2] **Interpretability-based multimodal convolutional neural networks for skin lesion diagnosis** (2023)	This research proposes a multimodal CNN (EfficientNetB5 and CatBoost) combining images and metadata with interpretable modules, achieving high performance for skin lesion diagnosis.	+ 95.1% accuracy + GradCAM provides visual interpretability + SHAP provides feature importance	- Computationally demanding - Requires extensive data preparation

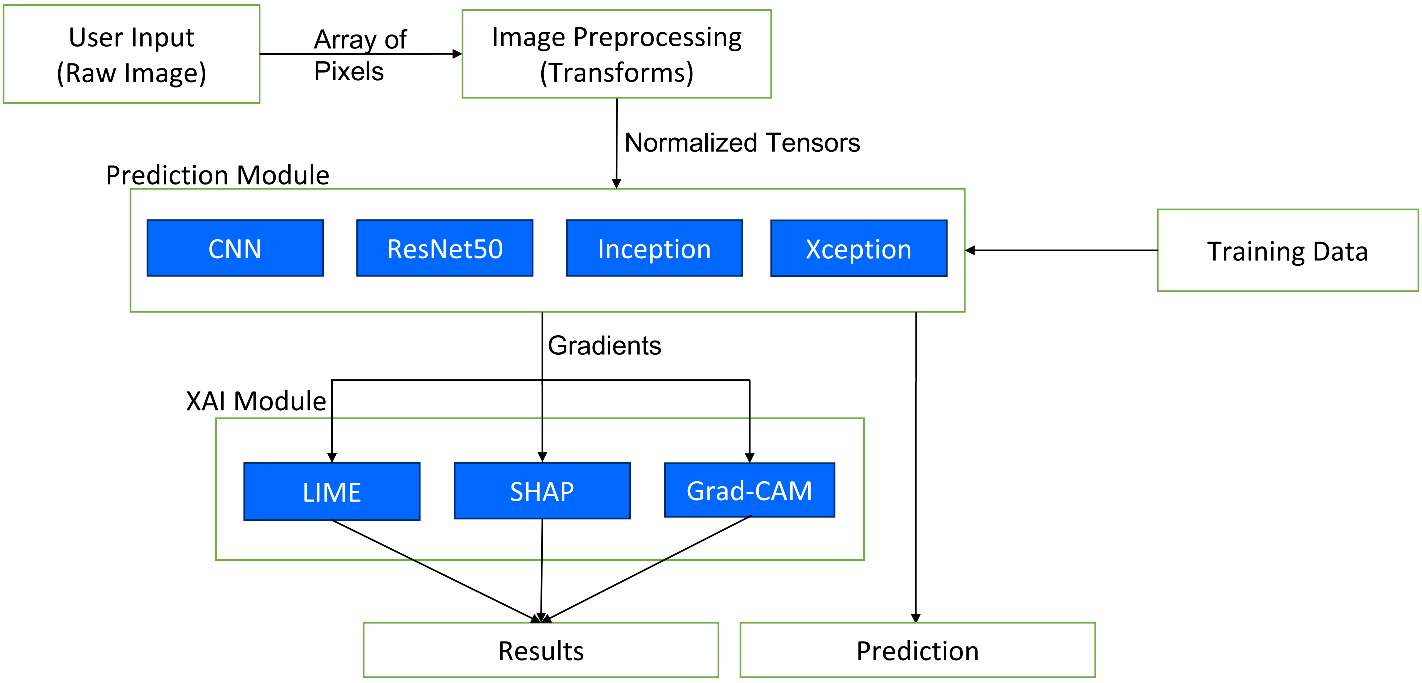
Literature Review

Sarthak Gupta [3] **Skin Lesion Classification Based on Various Machine Learning Models Explained by Explainable Artificial Intelligence** (2023)	This study develops a skin lesion classifier using CNN and XGBoost, enhanced with XAI methods LIME and SHAP to ensure interpretability and transparency.	+ Employs data augmentation + LIME and SHAP provide interpretability	- 86.3% accuracy - Limited dataset variation - Potential overfitting
Bhuvaneshwari Shetty [4] **Skin Lesion Classification of Dermoscopic Images using Machine Learning and Convolutional Neural Networks** (2022)	This study uses CNNs with data augmentation and k-fold cross-validation on the HAM10000 dataset.	+ 95.18% accuracy + Scalable + Provides automation features	- High dependency on image quality - Resource intensive

Literature Review

Iván Matas [5] **AI-driven Skin Cancer Diagnosis: Grad-cam and Expert Annotations for Enhanced Interpretability** (2024)	This study integrates MobileNet-V2 with Grad-CAM and expert annotations to enhance interpretability and accuracy in diagnosing basal cell carcinoma (BCC).	+ 90% accuracy + Resource efficient + GradCAM provides a visual interpretation on explainability	- Struggles with identifying some BCC regions - Limited to BCC regions

Architecture Diagram



Methodology

- **Data Collection**
  - Gather a diverse dataset of skin lesion images, including labeled examples of various skin cancers.
  - Ensure that the dataset represents different skin types and lighting conditions to enhance the model's robustness.
  - Sourced from kaggle.com and ISIC archives.
  - Total image count : 27,305 (~11,000 training, ~2000 testing)
  - Size : 24 GB

- **Data Preprocessing**
  - Split the dataset into training and test sets to evaluate model performance accurately.
  - Resize images to a consistent format for input into the model.
  - Perform horizontal and vertical flips.
  - Normalize the images to improve model performance.

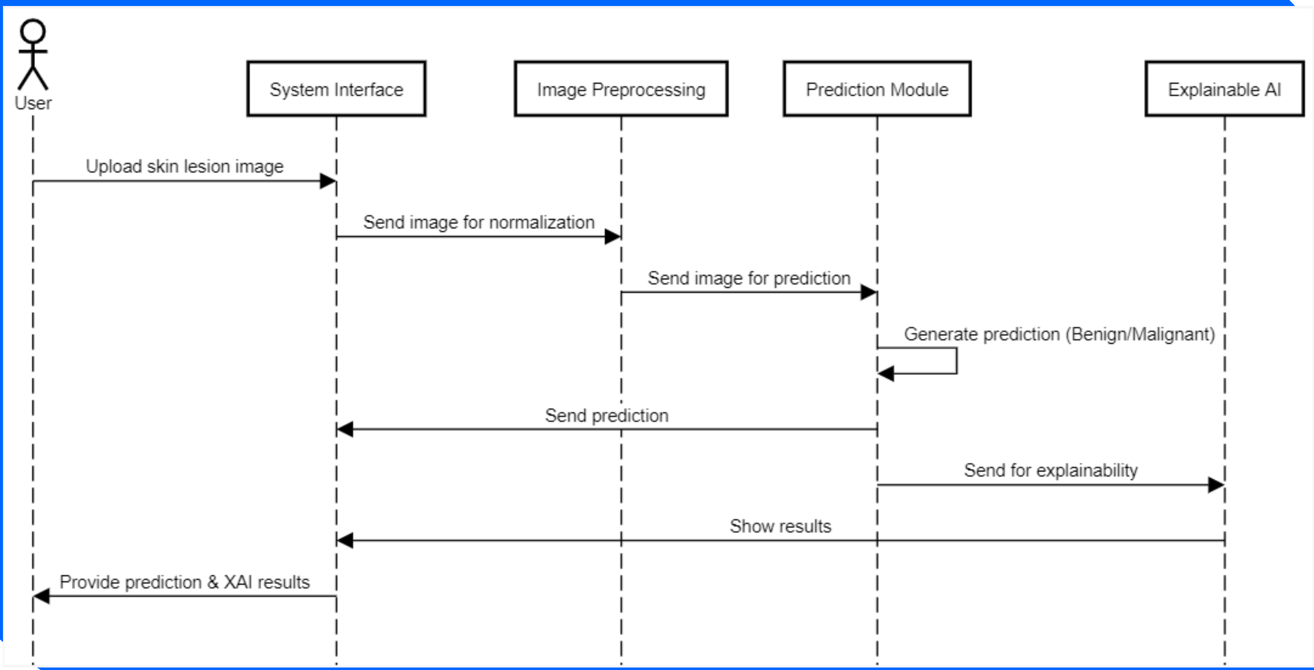
- **Model Development**
  - Design a CNN architecture specifically tailored for image classification tasks.
  - The model will learn to extract and analyze features from skin images to identify potential cancerous lesions.
  - Integrate pre-trained models such as ResNet50, Inception & Xception for skin cancer detection.

- **Model Training and Evaluation**
  - Train the models using the training dataset and evaluate its performance on the validation set, adjusting hyperparameters as needed.
  - Once optimized, assess the final models on the test dataset.

- **Integration of Explainable AI**
  - Implement XAI techniques such as SHAP, LIME and Grad-CAM to provide interpretable insights into the model's predictions.
  - Use these methods to generate visual explanations of which parts of the images influenced the model’s decisions.

- **User Interface Development**
  - Create a user-friendly interface that allows healthcare professionals to upload images and receive diagnostic results, along with explanations generated by the XAI methods.

Sequence Diagram



Modules

- Input
- Image Preprocessing
- Prediction Module
- XAI Module
- Output

Input

- The raw input provided to the system is typically a clinical image of a skin lesion.
- Images are further separated into skin cancer and skin lesions.
- Images are captured under standardized conditions to ensure consistency in analysis.
- The input image is preprocessed to enhance quality.

Image Preprocessing

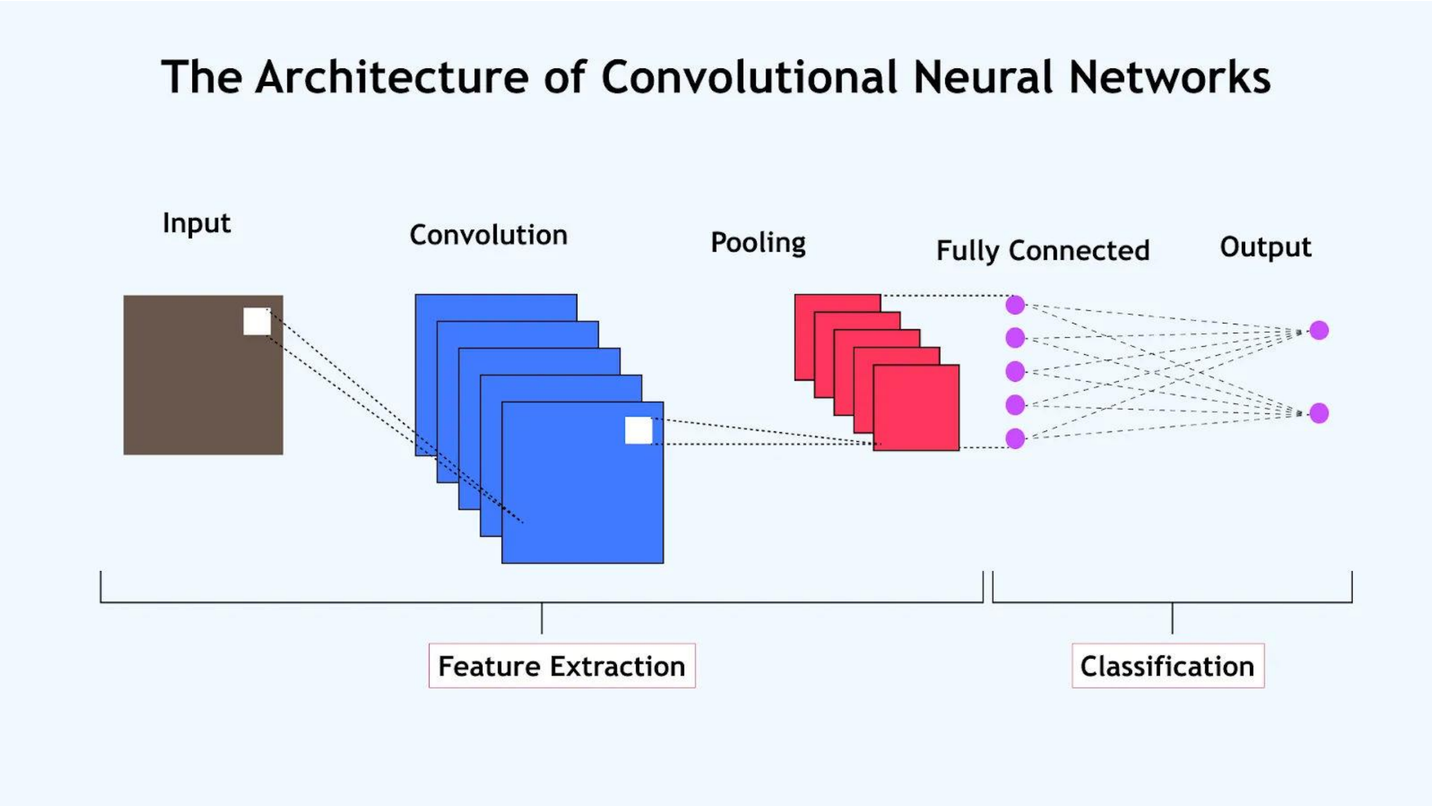
- Image preprocessing can help in improving model accuracy by reducing noise and irrelevant variations.

- **Resizing**: Ensures all images have the same dimensions (224, 224).
- **Flipping**: Horizontally and vertically flip the images randomly.
- **Tensor Conversion**: Converts the image into a PyTorch tensor with pixel values in [0,1].
- **Normalization**: Scales pixel values to (-1,1) to ensure fair comparison.

Convolution Neural Network

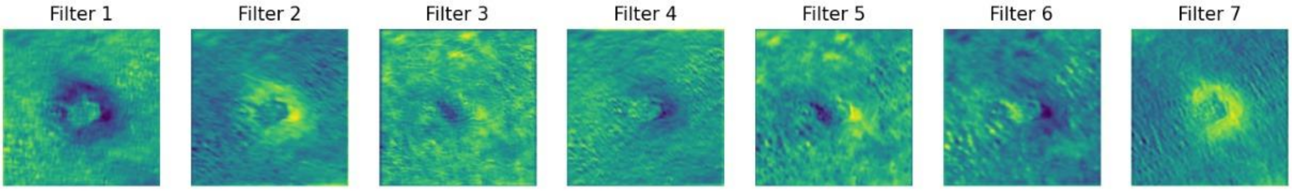
The extracted features are passed through multiple convolutional, pooling, and fully connected layers. Each layer refines the understanding of the image:

- Convolutional Layers detect patterns and textures.
- Pooling Layers reduce spatial dimensions and retain important features.
- Fully Connected Layers combine the learned features to make a final decision.
- CNN predicts probabilities for classes Benign and Malignant



Layer-wise Description

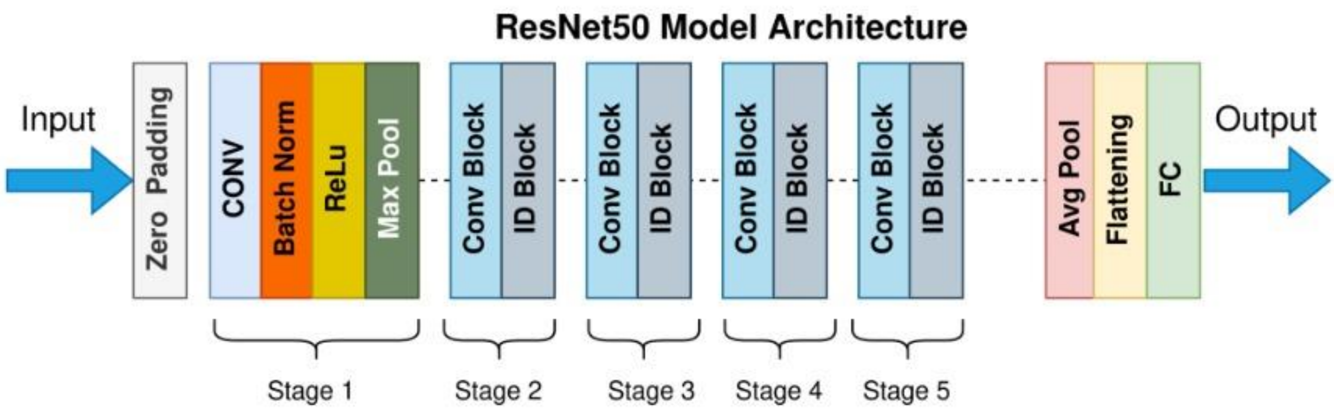
Layer	Input Channels	Output Channels	Purpose
Conv1	3 (RGB)	16	Detects basic edges
Conv2	16	32	Detects textures
Conv3	32	64	Detects patterns
Conv4	64	128	Detects object parts
Conv5	128	256	Detects complex structures
Conv6	256	512	Detects high-level features
Conv7	512	512	Final feature extraction



ResNet (Residual Network)

ResNet50 introduces skip connections to allow gradients to flow directly through the network, preventing vanishing gradient issues in deep architectures.

- **Residual Blocks**: Enable learning deeper features efficiently by skipping certain connections to avoid vanishing gradients.
- **Convolutional Layers**: Extract low-level and high-level features.
- **Pooling Layers**: Reduce spatial dimensions while keeping important features.
- **Fully Connected Layers**: Aggregate learned features for classification.

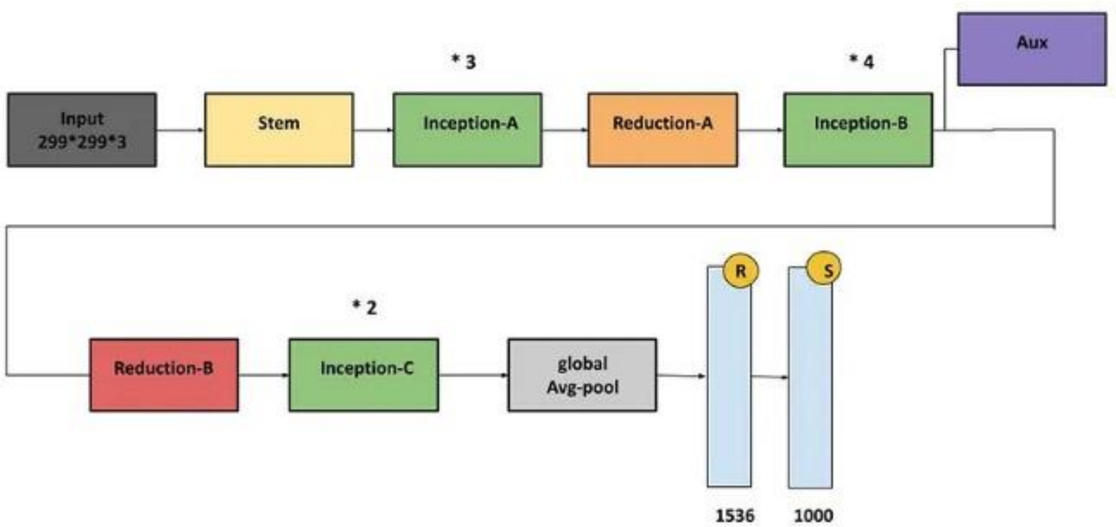


Resnet-50 Model architecture

Inception Model

Inception uses multiple convolutional filters at different scales in parallel, enabling the network to capture both fine and coarse features.

- **Inception Blocks**: Apply 1x1, 3x3, and 5x5 convolutions in parallel.
- **Pooling Layers**: Reduce spatial size while retaining essential information.
- **Fully Connected Layers**: Combine extracted features for decision-making.
- **Output Layer**: Predicts lesion class probabilities (**benign/malignant**).

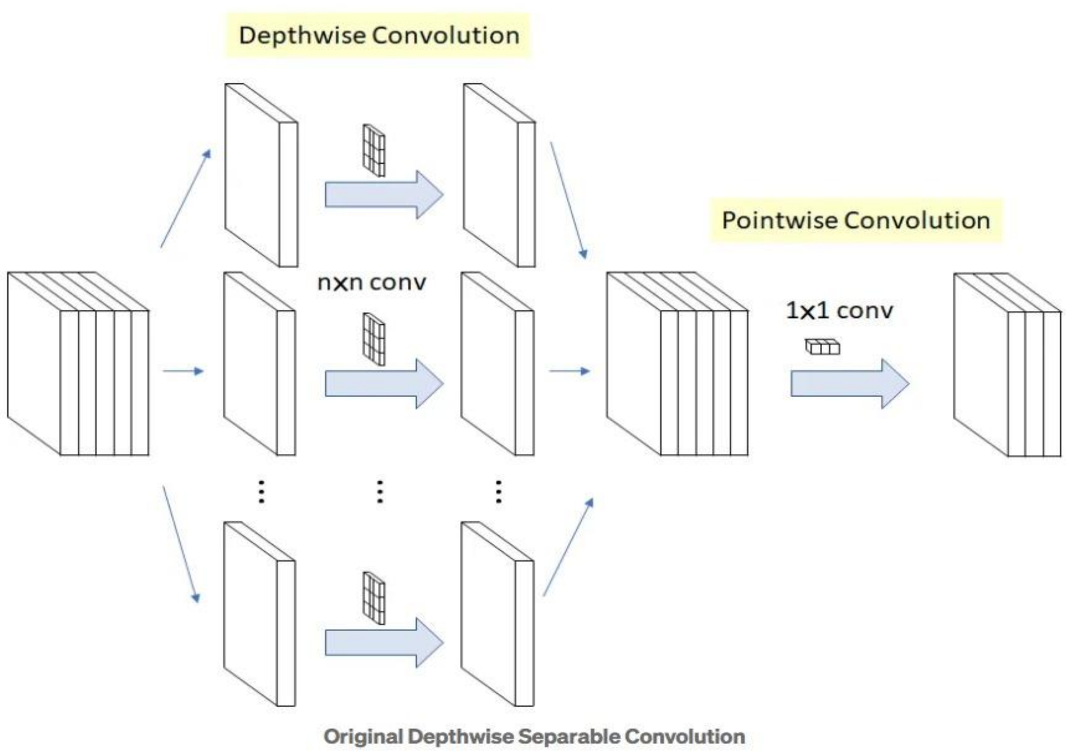


Inception Model Architecture

Xception (Extreme Inception)

Xception makes use of Depthwise Separable Convolutions where standard convolutions are broken into two steps:

- **Depthwise convolution**: Instead of applying a single filter across all input channels, depthwise convolution applies a separate filter to each input channel.
- **Pointwise Convolution**: After depthwise convolution, pointwise convolution (1×1 convolution) is applied to combine the outputs from depthwise convolution across channels.



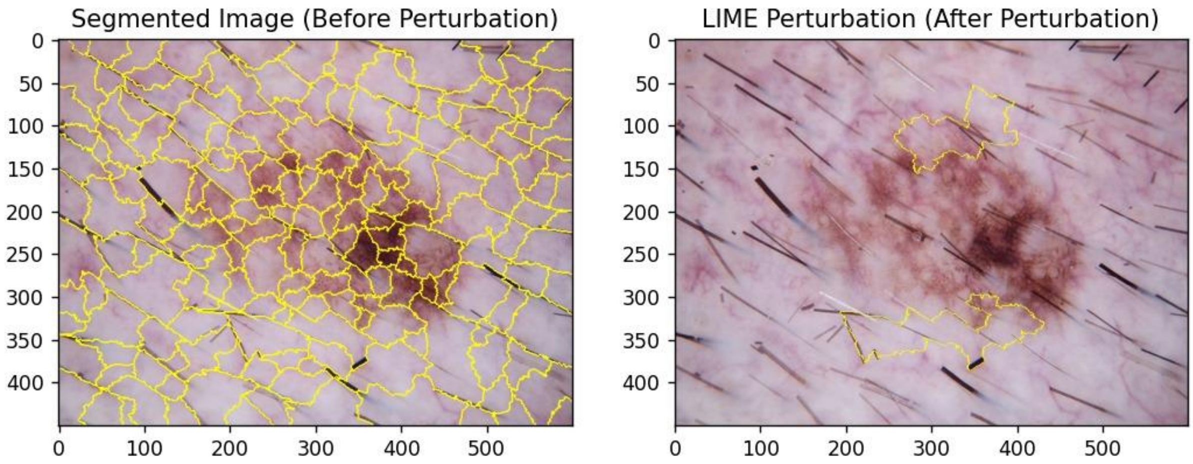
Xception Model Architecture

Explainable AI Module

- The XAI (Explainable AI) module is designed to provide interpretability to the model's predictions.
- Doctors need to trust and understand the CNN model's predictions to make informed decisions.
- XAI helps by making these predictions interpretable and transparent.

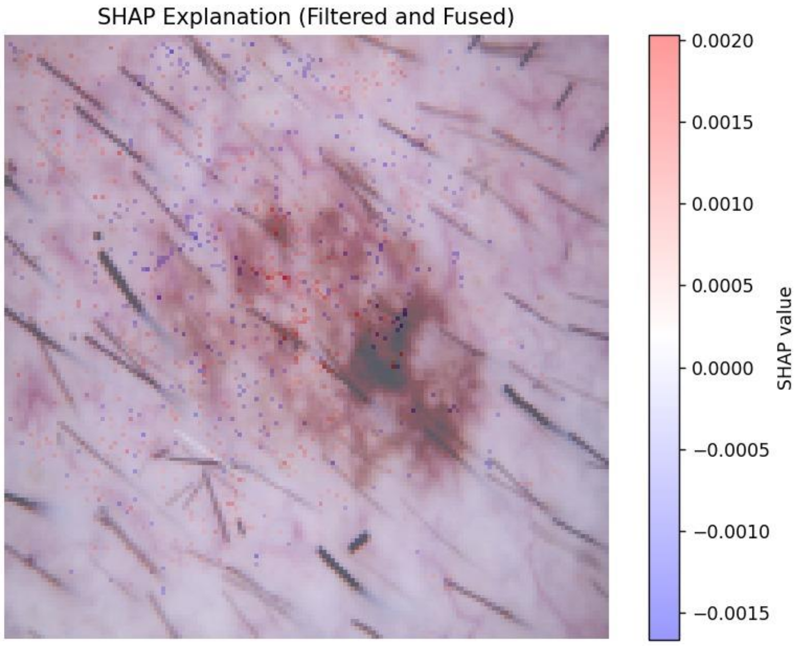
Local Interpretable Model-Agnostic Explanations (LIME)

- LIME approximates the behavior of the CNN model around a specific prediction by building a simpler, interpretable model for that specific prediction.
- It works by creating perturbed versions of the input image and seeing how the CNN's predictions change.
- If the CNN classifies an image as cancerous, LIME explains which features contributed the most to that specific prediction.



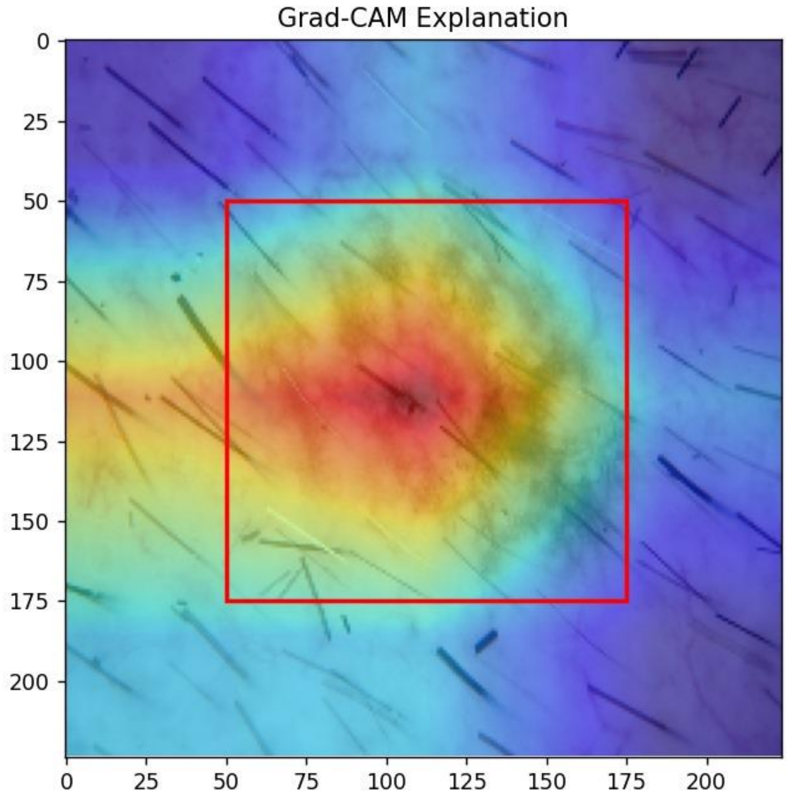
SHapley Additive exPlanations (SHAP)

- SHAP assigns each feature in the input an importance value and explains the contribution of each input feature towards the final prediction.
- SHAP can quantify the impact of individual features toward the model's prediction in the form of Shapely Values.
- This can be explained through plots such as bee-swarm plots, force plots, etc.



Gradient-weighted Class Activation Mapping (GradCAM)

- GradCAM creates visual heatmaps that highlight the regions in the input image that had the greatest influence on the prediction.
- It works by analyzing the gradients of the CNN model's last convolutional layers and showing where the model focused its attention.
- For an image classified as cancerous, GradCAM will highlight the regions of the skin that were most influential in the model's decision.



Assumptions

- **Image quality**
  The input image has sufficient resolution to capture fine details like edges, texture and color patterns.

- **Presence of Lesion**
  Each input image contains a skin lesion clearly centered and visible without significant occlusion.

Work Breakdown

Renu Lijo
- Xception training
- Grad-CAM

Rhea John
- ResNet50 training
- LIME

Rohan Jojo
- CNN training
- Inception training

Tom Rajeev
- SHAP
- Frontend & Backend

Hardware & software requirements

Hardware

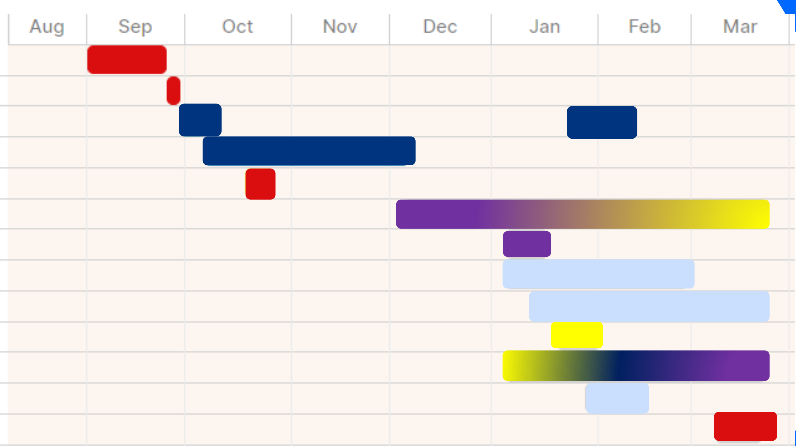
- CPU : Intel Core i5 / AMD Ryzen 5
- GPU : NVIDIA GeForce GTX 1650 (4BG RAM)
- OS : Windows 11

Software

- Development Environment : MobaXtreme (Sunya Lab), Visual Studio Code
- Libraries : PyTorch, LIME, SHAP
- Framework : Flask, HTML/CSS

Gantt Chart

Activity	Start	End
Literature Review	02-09-24	24-09-24
Architecture Design	25-09-24	27-09-24
Dataset Acquisition & Processing	30-09-24	04-10-24
CNN Training	07-10-24	05-12-24
UI Design	18-10-24	25-10-24
Research Paper	05-12-24	24-03-25
GradCAM Integration	06-01-25	16-01-25
Frontend Development	06-01-25	27-02-25
Backend Development	14-01-25	21-03-25
LIME Integration	21-01-25	31-01-25
ResNet, Inception, Xception Training	06-01-25	24-03-25
SHAP Integration	29-01-25	14-02-25
Evaluation and Testing	10-03-25	26-03-25



Rohan Jojo
Renu Lijo
Common
Tom Rajeev
Rhea John

Risk & challenges

- **Privacy and Ethical Issues:**
  Medical images contain sensitive patient information, raising concerns about data privacy and compliance with HIPAA or GDPR regulations.

- **Overfitting:**
  CNNs are prone to overfitting, especially with small datasets, where the model performs well on the training data but poorly on new, unseen data.

- **Trust and Transparency Issues:**
  Doctors may hesitate to trust AI predictions without clear explanations, particularly for life-critical decisions like cancer diagnosis.

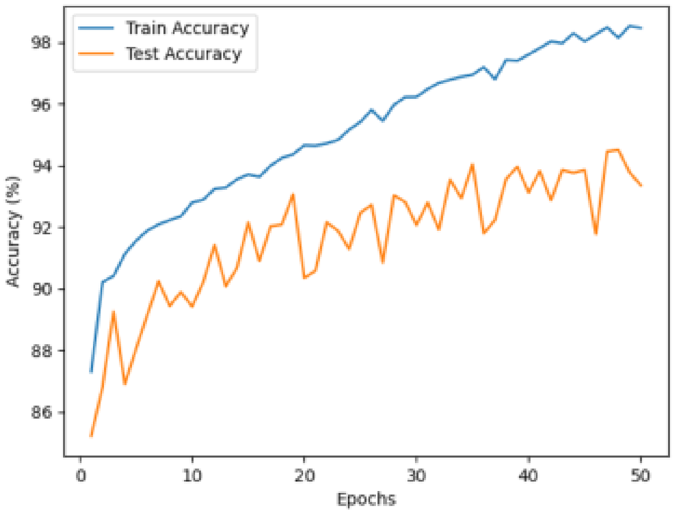
Results

Testing out with CNN

The graph above shows the training and testing accuracy of your custom CNN model over the final epochs (44 to 50).

- **Train Accuracy (Blue Line)**: The training accuracy remains relatively high and stable, suggesting that the model is well-fitted to the training data by the end of training.
- **Test Accuracy (Orange Line)**: The test accuracy fluctuates significantly and does not show consistent improvement, indicating that the model may not be generalizing well to unseen data, potentially pointing to overfitting.

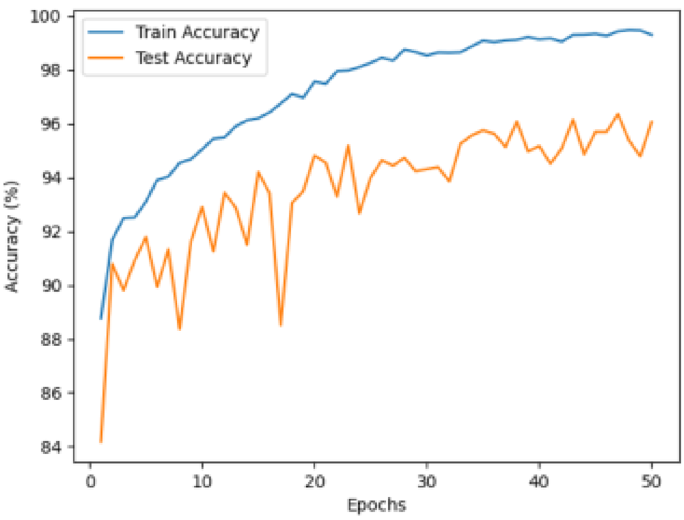
Testing out with Resnet Model



The graph above displays the training and testing accuracy of the ResNet50 model over 50 epochs.

- **Train Accuracy (Blue Line)**: The training accuracy consistently increases, indicating the model is learning and improving its predictions on the training data as epochs progress.
- **Test Accuracy (Orange Line)**: The test accuracy follows a similar upward trend, demonstrating that the model is generalizing well to unseen data. However, there is some fluctuation, which may suggest slight overfitting at certain points.

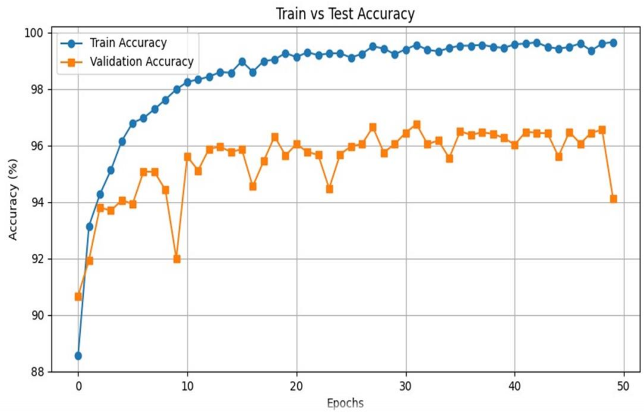
Testing out with Inception



The graph above illustrates the training and validation accuracy of the Inception model over 50 epochs.

- **Train Accuracy (Blue Line)**: The training accuracy steadily increases, showing that the model is learning and improving its predictions on the training dataset.
- **Validation Accuracy (Orange Line)**: The validation accuracy follows a similar upward trend but with some fluctuations. Despite these fluctuations, it stays relatively high and demonstrates that the model is generalizing well to the unseen validation data.

Testing out with Xception



The graph above illustrates the training and validation accuracy of the Xception model over 50 epochs.

- **Train Accuracy (Blue Line)**: The training accuracy steadily increases, showing that the model is learning and improving its predictions on the training dataset.
- **Validation Accuracy (Orange Line)**: The validation accuracy follows a similar upward trend but with some fluctuations. Despite these fluctuations, it stays relatively high and demonstrates that the model is generalizing well to the unseen validation data.

Comparison Table

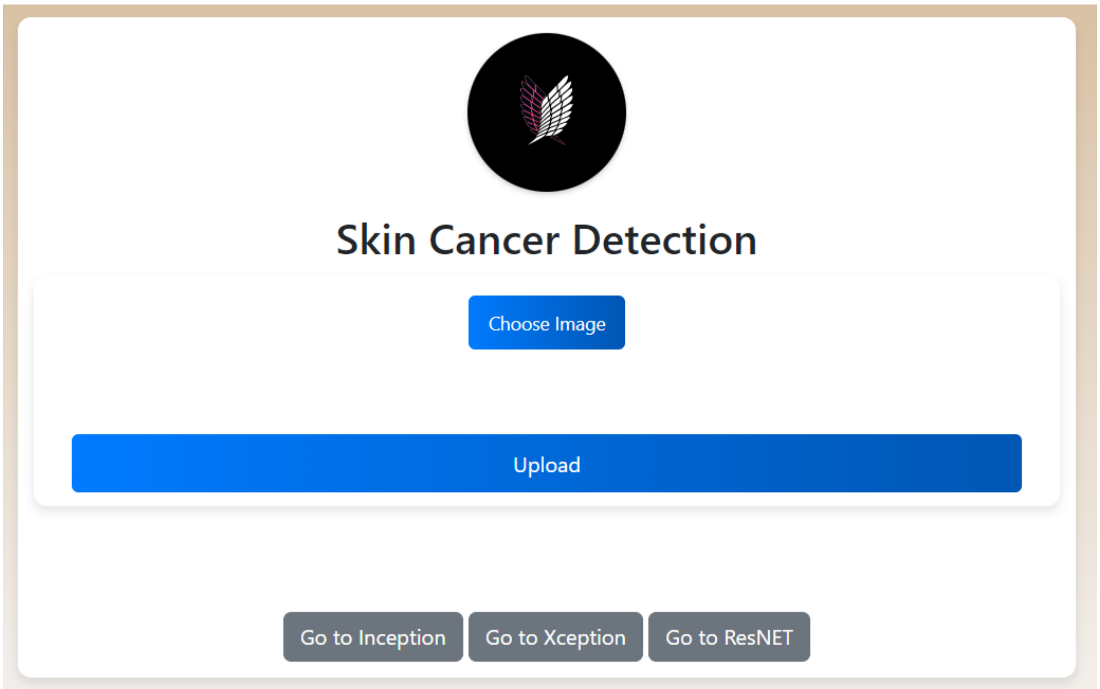
Features	CNN	ResNet	Inception	Xception
Number of Layers	7	50	48	71
Accuracy (%)	93.19	94.51	96.15	96.75
Speed (s)	24	166	124	110 (No SHAP)
Advantages	Easy to Train	Solves Vanishing Gradient problem	Multi-Scale Feature Extraction	Reduces required parameters
Disadvantages	Struggles with Depth	Large Model Size	Complex Architecture	High Computational Cost

UI/UX

UI Components:

- **File Upload**: Users can select an image of a skin lesion for analysis.
- **Prediction Button**: After uploading, users can click **“Upload”** to analyze the skin lesion. Prediction result (e.g., Melanoma/Benign).
- **Model switch**: Three buttons to switch between models.

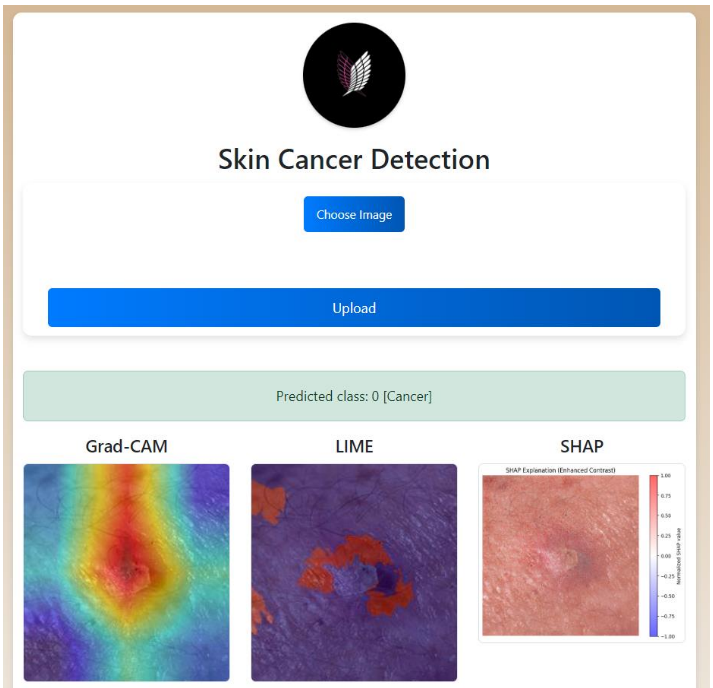
UI/UX



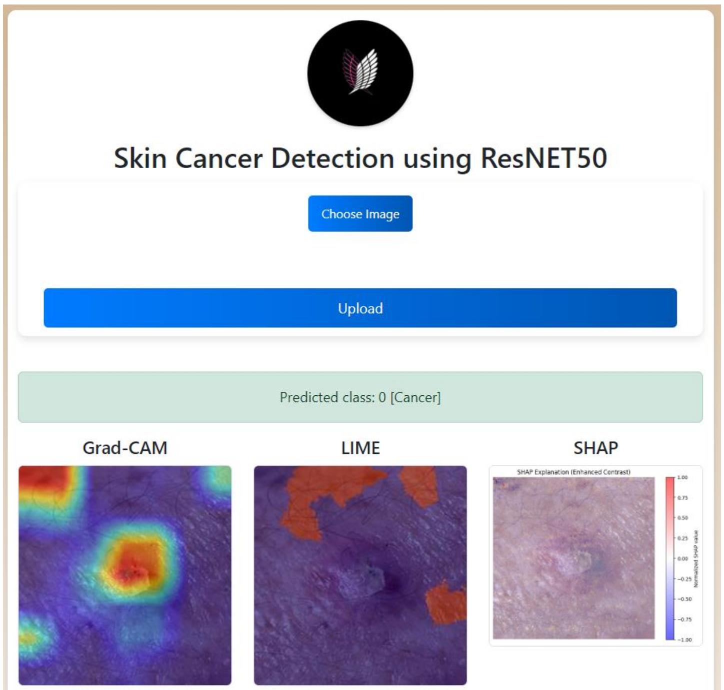
XAI Integration

- Integrated **Grad-CAM, LIME, and SHAP** for a more comprehensive explanation of model predictions.
- **Grad-CAM** highlights key regions influencing the prediction.
- **LIME** provides local interpretability by approximating feature importance for individual predictions.
- **SHAP** offers a global understanding of feature contributions across the dataset.
- **Purpose** – Enhances transparency, helping users trust AI-driven skin cancer detection.

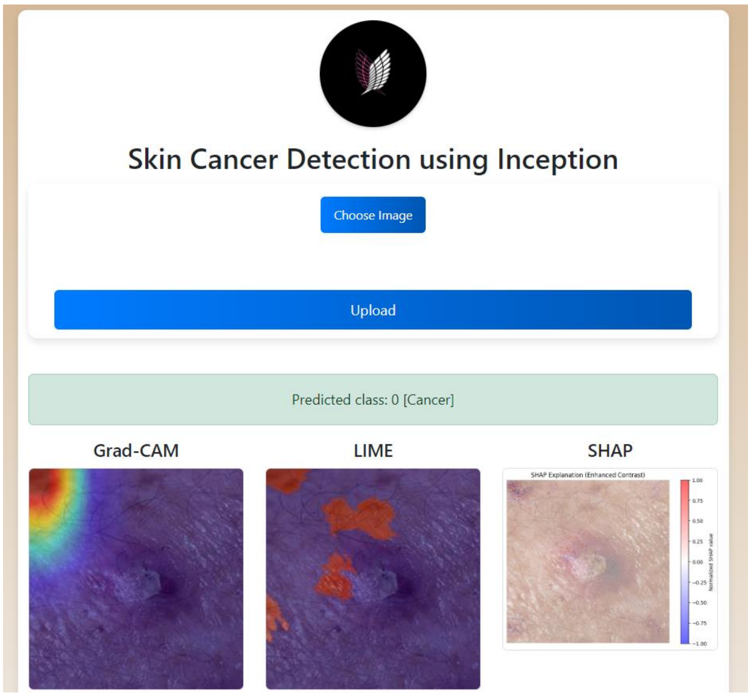
Results



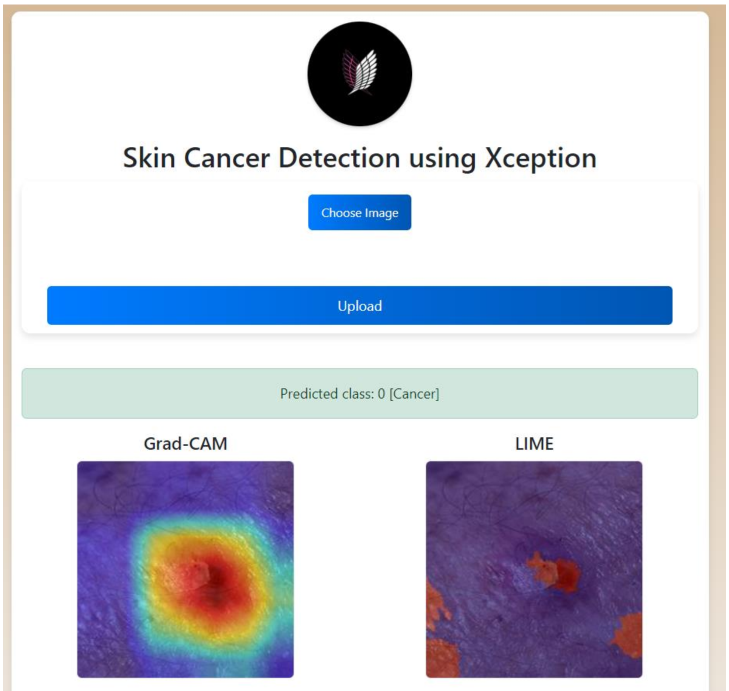
Results



Results



Results



Research Paper Publication

- **Title:** *Skin Cancer Detection Using Explainable AI (XAI)*
- **Objective:** Enhance skin cancer detection accuracy with transparent AI explanations.
- **Status**: Pending
- **Conference**: IEEE 4th International Conference on Advances in Computing, Communication, Embedded and Secure Systems

Future Work

- Extend the current binary classification to detect multiple types of skin cancers including melanoma, seborrheic keratosis, and benign nevi.
- Fuse image-based models with patient metadata (e.g., age, lesion location, skin type) to enhance diagnostic accuracy through multimodal learning.
- Implement an active learning pipeline where dermatologists can iteratively provide feedback to improve model performance and refine interpretability.

Conclusion

- This skin cancer detection system integrates multiple deep learning models with Explainable AI (Grad-CAM, LIME, SHAP) to enhance accuracy and transparency.
- With an improved UI and expanded model training, users receive more precise predictions along with clear visual explanations.
- By refining interpretability and accuracy, this project strengthens trust in AI-driven diagnosis, supporting early detection and better health outcomes.

References

[1] A. Panthakkan, S.M. Anzar, "Concatenated Xception-ResNet50 — A novel hybrid approach for accurate skin cancer prediction" (2022).

[2] S. Wang and Y. Yin, "Interpretability-based multimodal convolutional neural networks for skin lesion diagnosis" (2023).

[3] Sarthak Gupta, " Skin Lesion Classification Based on Various Machine Learning Models Explained by Explainable Artificial Intelligence" (2023).

[4] Bhuvaneshwari Shetty, "Skin Lesion Classification of Dermoscopic Images using Machine Learning and Convolutional Neural Networks" (2022).

References

[5] Iván Matas, "AI-driven Skin Cancer Diagnosis: Grad-cam and Expert Annotations for Enhanced Interpretability" (2024).

[6] Hung Truong, "Evaluation of Explainable Artificial Intelligence: SHAP, LIME, and CAM" (2022).

[7] SIIM & ISIC, "SIIM-ISIC Melanoma Classification" [Online]. Available: https://www.kaggle.com/competitions/siim-isic-melanoma-classification

[8] ISIC Archive, "ISIC Archive Collections " [Online]. Available: https://api.isic-archive.com/collections

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision
To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission
To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision
To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission
To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)
Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills
The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills
The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills
The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes (CO)

After the completion of the course the student will be able to:

Course Outcome 1: Model and solve real world problems by applying knowledge across domains (Cognitive knowledge level: Apply).

Course Outcome 2: Develop products, processes or technologies for sustainable and socially relevant applications (Cognitive knowledge level: Apply).

Course Outcome 3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: Apply).

Course Outcome 4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: Apply).

Course Outcome 5: Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: Analyze).

Course Outcome 6: Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: Apply).

JUSTIFICATIONS FOR CO-PO MAPPING

Mapping	Level	Justification
101003/CS822U.1-PO1	H	Application of core concepts in machine learning and medical imaging to detect skin cancer using explainable AI.
101003/CS822U.1-PO2	H	Ability to analyze dermatological image data, apply Grad-CAM, LIME, and SHAP to interpret model decisions.
101003/CS822U.1-PO3	H	Design of a complete skin cancer detection system using PyTorch integrated with explainability tools.
101003/CS822U.1-PO4	M	Evaluation of model performance and effectiveness of Grad-CAM, LIME, and SHAP explanations.
101003/CS822U.1-PO5	H	Use of PyTorch and visualization libraries for implementing and interpreting the model's predictions.
101003/CS822U.1-PO6	M	Understanding the impact of explainable AI in healthcare and its benefits for patient trust and diagnosis.
101003/CS822U.1-PO7	L	Optimization of the training process to reduce computational costs while maintaining interpretability.
101003/CS822U.1-PO8	M	Ethical considerations in handling sensitive patient data and generating transparent AI-based diagnoses.
101003/CS822U.1-PO9	M	Team collaboration in model training, applying Grad-CAM/LIME/SHAP, and validating outputs.
101003/CS822U.1-PO10	H	Clear presentation of project methodology and explainable outputs using visual and verbal communication.
101003/CS822U.1-PO11	H	Systematic execution of tasks including dataset preparation, model development, and interpretation using XAI tools.
101003/CS822U.1-PO12	H	Learning and application of emerging explainable AI tools such as LIME, SHAP, and Grad-CAM in healthcare.

101003/CS822U.1-PSO1	H	Applying core computer science skills in PyTorch and XAI to solve real-world medical problems.
101003/CS822U.2-PSO2	M	Contribution to society through development of transparent and accessible AI tools for skin cancer diagnosis.
101003/CS822U.3-PSO3	H	Use of LIME, SHAP, and Grad-CAM to build trust in AI-based diagnosis by making results interpretable.
101003/CS822U.4-PSO3	H	Effective planning and execution of explainable AI workflows using Grad-CAM, LIME, and SHAP.
101003/CS822U.5-PSO1	H	Building an innovative solution using PyTorch integrated with XAI tools for better medical decision-making.
101003/CS822U.6-PSO3	H	Documentation and presentation of findings with a focus on visual explanations and their significance.