A Project Report on

DERMATOLOGICAL MALIGNANCY DETECTION USING CNN

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

In

Computer Science and Engineering

Submitted by

M.Deepak Reddy
(20H51A05H7)

S.Jashwitha
(20H51A05L7)

M.V.Devendranathreddy
(20H51A05P6)

Under the esteemed guidance of

Ms. Komal Parashar
(Assistant Professor)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A^+ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020 - 2024

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project report entitled **"DERMATOLOGICAL MALIGNANCY DETECTION USING CNN"** being submitted by M. Deepak reddy (20H51A05H7), S.Jashwitha (20H51A05L7), M.V.Devendranathreddy (20H51A05P6) in partial fulfillment for the award of Bachelor of Technology in **Computer Science And Engineering** is a record of bonafide work carried out under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Komal Parashar
Assistant Professor
Dept. Of CSE

Dr. Siva Skandha Sanagala
Associate Professor and HOD
Dept. Of CSE

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express our heartfelt gratitude to all the people who helped in making this project a grand success.

We are grateful to **Ms. Komal Parashar Assistant Professor** , Department of Computer Science and Engineering for her valuable technical suggestions and guidance during the execution of this project work.

We would like to thank**, Dr. Siva Skandha Sanagala,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete our project work successfully.

We are very grateful to **Dr. Ghanta Devadasu**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary& Correspondent, CMR Group of Institutions, and Shri Ch Abhinav Reddy, CEO, CMR Group of Institutions for their continuous care and support.

Finally, we extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly or indirectly in completion of this project work.

M. Deepak Reddy 20H51A05H7
S.Jashwitha 20H51A05L7
M.V.Devendranathreddy 20H51A05P6

TABLE OF CONTENTS

List of Figures

List of Tables

ABSTRACT

Skin cancer is a prevalent and potentially life-threatening disease that continues to pose a significant public health concern. Starting stage detection Plays a crucial part in enhancing treatment efficacy. Furthermore, the advancement of deep learning has enabled the creation of automated frameworks for detecting maligant cancer. This proposed system utilizes a varied dataset encompassing various types of skin cancer, such as malignant melanomas and benign nevi, to educate a learning model. Utilizing CNN, pertinent features are automatically extracted from skin cancer images, facilitating precise classification. Assessment of the suggested system showcases encouraging outcomes concerning sensitivity, specificity, and overall precision. Comparative assessments against conventional techniques underscore the superior efficacy and effectiveness of the devised model in identifying potential skin cancer instances. Moreover, the system's interpretive capacity is heightened through the integration integration of attention mechanisms offering insights into the areas of focus within the skin cancer images

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1 Problem Statement

Skin cancer is a significant public health concern, with early detection playing a crucial role of a crucial role in improving patient outcomes. The problem statement for skin cancer detection using CNNs involves developing an accurate and efficient deep learning model capable of automatically classifying skin lesions as benign or malignant from images.Design a CNN architecture tailored for skin cancer detection. Experiment with different network architectures, including variations of convolutional, pooling, and fully connected layers, to optimize performance.By addressing these components, the goal is to develop a CNN-based skin cancer detection system that can accurately and efficiently identify malignant lesions, facilitating early diagnosis and improving patient outcomes

1.2 Research Objective

The primary objective of research in skin cancer detection is to develop accurate, reliable, and efficient methods for early detection and classification of skin lesions. This involves leveraging advancements in technology, particularly in the field of machine learning and computer vision, to enhance the diagnosis process. The specific research objectives include By addressing these research objectives, the aim is to advance the field of skin cancer detection, ultimately leading to improved patient outcomes, reduced healthcare costs, and increased accessibility to early diagnosis and treatment.

1.3 Project Scope and Limitations

Project Scope

Acquire a diverse dataset of skin lesion images, ensuring it covers various skin types, lesion types, and imaging conditions. Preprocess the images to standardize size, resolution, and lighting conditions.

Limitations

Limitations in Skin Cancer Detection:

1. **Data Availability and Quality:** Limited availability of large, diverse datasets with annotated skin lesion images can hinder the development and evaluation of skin cancer detection algorithms. Additionally, variability in image quality, lighting conditions, and imaging techniques may affect model performance.
2. **Data Imbalance:** Imbalanced datasets, where the number of malignant lesions is significantly smaller than benign lesions, can lead to biased model predictions and re duced performance in detecting malignant cases.
3. **Generalization Across Populations:** Skin cancer detection algorithms may not generalize well across diverse populations due to variations in skin types, ethnicities, and genetic backgrounds. Models trained on data from one population may not perform optimally when applied to another population.
4. **Interpretability:** Deep learning models, particularly convolutional neural networks (CNNs), are often considered black-box models, making it challenging to interpret the reasoning behind their predictions. Lack of interpretability may limit trust and acceptance among dermatologists and healthcare professionals.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1 Skin cancer classification using Resnet

2.1.1 Introduction:

Skin cancer can be dangerous, and its severity relies on the hinge of skin cancer, its stage at the time of diagnosis, and how quickly it is treated. basal cell, carcinoma , squamous cell carcinoma , and melanoma[3] etc are top skin diseases.personal Detecting skin maligant using CNN is a common and effective approach with in the realm of randiomics. CNNs are well-suited in the objective like classification, making them suitable for identifying patterns and features in medical images

Skin cancer detection using ResNet involves employing a learning model, specifically ResNet (Residual Neural Network), to analyze and classify skin lesions as either benign or malignant network Evaluate the data model on the test the dataset set to assess its generalization performance. Metrics like exactitude, exactness, retrieval rate , and F1 score for the proficiency the model's It's significant name of the success of dermatological malignancy detection using ResNet reliable on quality and diversity of the dataset, appropriate hyperparameter tuning, and magnitude of the model. Continuous monitoring and updates may be necessary to adapt to new data and enhance the model's performance over time[1].

Remember that while deep learning models like ResNet can achieve high accuracy in skin cancer detection, they should be used as decision support tools rather than replacements for professional medical diagnosis. Always consult with a healthcare professional for accurate diagnosis and treatment

while deep learning models like ResNet can achieve impressive results, they are not a substitute for medical expertise. Always consult with medical professionals for accurate diagnosis and treatment recommendations

2.1.2 Merits ,Demerits and challenges :

Merits :

1. **Deep Architecture**: ResNet is a deep neural network architecture that is highly effective in learning complex patterns and features from images. Its deep layers allow for the extraction of intricate features from skin lesion images, enabling more accurate classification of benign and malignant lesions.
2. **Residual Learning**: ResNet introduces the concept of residual learning, which helps address the problem of vanishing gradients in very deep networks. This allows for the training of deeper networks with hundreds of layers while maintaining performance and avoiding degradation issues.
3. **Transfer Learning**: Pre-trained ResNet models, trained on large datasets such as ImageNet, can be fine-tuned for skin cancer detection tasks. Transfer learning allows leveraging knowledge gained from general image recognition tasks to improve performance on specific medical imaging tasks, even with limited training data.
4. **Robustness to Overfitting**: ResNet's residual connections help mitigate overfitting, a common challenge in deep learning models, by facilitating better gradient flow during training. This can lead to more robust models that generalize well to unseen data, improving the reliability of skin cancer detection systems.
5. **Efficient Training**: Despite its depth, ResNet can be trained efficiently using techniques like batch normalization and parallel processing. This allows for faster convergence during training and reduces the computational resources required, making it practical for real-world applications in skin cancer detection.

Demerits:

1. **Complexity and Overfitting**: ResNet's deep architecture, while advantageous for learning complex features, can also lead to overfitting, especially when dealing with limited training data or noisy datasets. Training deep networks like ResNet requires careful regularization techniques to prevent overfitting and ensure generalization to unseen data.
2. **Computational Resources**: The deep architecture of ResNet requires significant computational resources for training and inference, including high-performance GPUs or TPUs. This can pose challenges for deployment in resource-constrained environments, especially in healthcare settings with limited access to specialized hardware.
3. **Training Data Requirements**: ResNet's effectiveness relies on large and diverse datasets for training. Obtaining labeled medical imaging data, especially for rare skin conditions or specific demographics, can be challenging and time-consuming. Biases in the training data can also impact the model's performance and generalization ability.
4. **Interpretability Challenges**: Despite its interpretability advantages compared to some other deep learning architectures, understanding the decisions made by ResNet models can still be challenging. The complex interactions between layers and features may not always align with human intuition, limiting the interpretability of model predictions, which is crucial for clinical acceptance.
5. **Fine-tuning and Transfer Learning**: While transfer learning with pre-trained ResNet models can accelerate training and improve performance, fine-tuning for specific medical imaging tasks may still require expertise and careful parameter tuning. Inadequate fine-tuning or inappropriate transfer learning strategies can lead to suboptimal performance or biases in the model.

Challenges:

1. **Limited Data Availability**: Obtaining large and diverse datasets of labeled skin lesion images for training ResNet models can be challenging. Data scarcity, especially for rare or specific types of skin cancer, can hinder model performance and generalization to new cases.
2. **Data Quality and Annotation**: Ensuring the quality and accuracy of labeled data is crucial for training effective ResNet models. Variations in image quality, inconsistent labeling, and inter-observer variability in diagnosis can introduce noise and biases into the training data, impacting model performance.
3. **Model Complexity and Overfitting**: ResNet's deep architecture makes it prone to overfitting, especially when dealing with limited training data. Balancing model complexity with generalization ability is essential to prevent overfitting and ensure robust performance on unseen data.
4. **Interpretability**: Understanding the decisions made by ResNet models can be challenging due to their complex architecture and hierarchical feature representations. Ensuring the interpretability of model predictions is crucial for clinical acceptance and trust among healthcare professionals.

2.1.3 Implementation:

Implementing a skin cancer detection system using ResNet involves several steps, including data collection, preprocessing, model training, evaluation, and deployment. Here's a high-level overview of the implementation process:

1. **Data Collection and Preparation**:
    - Gather a large dataset of skin lesion images with corresponding labels (e.g., benign or malignant).
    - Ensure that the dataset is diverse, representative of different skin types, ages, and lesion types.
    - Preprocess the images, including resizing, normalization, and augmentation to improve model generalization

1. **Model Selection**
    - Choose a ResNet architecture suitable for the skin cancer detection task, such as ResNet-50, ResNet-101, or ResNet-152, based on computational resources and performance requirements.
    - Optionally, utilize transfer learning by initializing the ResNet model with pre-trained weights on large-scale image datasets like ImageNet.
2. **Model Training**
    - Split the dataset into training, validation, and test sets to assess model performance.
    - Fine-tune the ResNet model on the training data using techniques like stochastic gradient descent (SGD) with momentum or adaptive optimization algorithms like Adam.
    - Monitor training progress using metrics such as loss and accuracy on the validation set and employ techniques like learning rate scheduling or early stopping to prevent overfitting.
3. **Model Evaluation**
    - Evaluate the trained ResNet model on the test set to assess its performance in detecting skin cancer lesions.
    - Calculate metrics such as accuracy, precision, recall, F1-score, and ROC curve to quantify the model's performance.
    - Conduct error analysis to identify common misclassifications and areas for improvement.
4. **Deployment**
    - Integrate the trained ResNet model into a software application or web service for deployment.
    - Ensure compliance with regulatory requirements and ethical guidelines for medical software.
    - Implement features for user interaction, such as uploading images, viewing predictions, and accessing diagnostic recommendations.

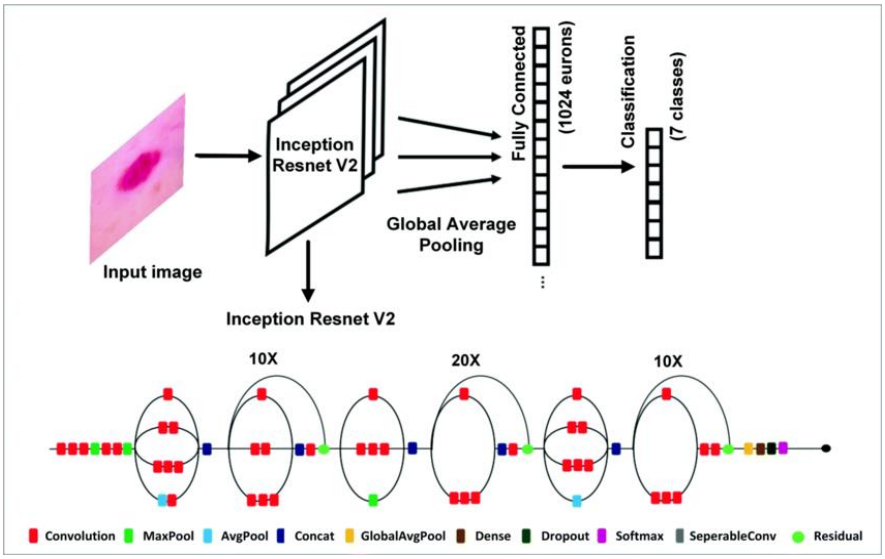


Fig* 2.1.3.1 *: Resnet implementation

2.2 Skin cancer classification using Vggnet

2.2.1 Introduction :

A. Obtain a dataset of dermatological cancer images. A ubicutious dataset for this task is reducing the structure of system. and fine-tuning it on a dataset of skin cancer images Resize images to the input size required by the VGGNet model (typically 224x224 pixels). Load the pre-trained VGGNet model. You can use a pre-trained VGG16 or VGG19 model from a learning library like Keras or PyTorch. Train the modified VGGNet on your skin cancer dataset. Use the training for instruction and the validation set to monitor the model's performance and avoid overfitting. Fine-tune the model by adjusting turningparameters like lsteprate,grouping magnitude, and the number of epochs., Utilize metrics such as exactitude, exactness, retrieval rate to evaluate the model's effectiveness in skin cancer classification.[2]

Skin cancer is one of the most common type of cancer in humans. This type of cancer is produced by skin cells called melanocytes and occurs as a result of division and multiplication of the mentioned cells. The most important symptom of skin cancer is the formation of spots on the skin or the observation of changes in the shape, color, or size of the existing spot. It is necessary to consult a specialist to distinguish the difference between a normal spot and skin cancer. Expert physicians examine and follow up the spots on the skin using skin surface microscopy, called dermatoscopy, or take a sample from the suspicious area and request it to be examined in laboratory environment. This situation increases the cost of the procedure for the diagnosis of skin cancer and also causes it to be treated at a later stage. Therefore, there is a need for a metod that can detect skin cancer early. Thanks to machine learning, become popular in recent years, many diseases can be diagnosed with software that helps expert physicians. In this study, VGGNet model structures (VGG-11, VGG-13, VGG-16, VGG-19) that quickly classify skin cancer and become a traditional convolutional neural network architecture using deep learning method, a subfield of machine learning, were used. It has been observed that the VGG-11 architecture, which is one of the VGGNet model structures, detects skin cancer with superior success accuracy (83%) compared to other model structures[1].

.2.2.2 Merits and Demerits :

Merits:

1. **Availability**: VGGNet is a well-established architecture and is readily available in deep learning libraries such as TensorFlow and PyTorch. This availability simplifies implementation, as researchers and developers can easily access pre-implemented versions of VGGNet and adapt them to their specific needs.
2. **Interpretability**: While deeper architectures like VGGNet may not be as interpretable as shallower ones, they still offer some degree of interpretability. Visualization techniques such as class activation maps can provide insights into which regions of the input images are most influential in the model's decision-making process, aiding clinicians in understanding and trusting the model's predictions.

3.Community Support: Due to its popularity and widespread use, there is a wealth of resources, tutorials, and research papers available on VGGNet and its applications. This extensive community support can be invaluable for developers and researchers seeking guidance or troubleshooting assistance during the implementation process.

4.Robustness: VGGNet's depth and architecture make it relatively robust to variations and distortions in input images, such as changes in lighting conditions, image orientation, or minor occlusions. This robustness is crucial for real-world applications where input images may vary in quality and appearance.

Demerits :

1. **Computational Complexity**: VGGNet is a deep convolutional neural network with a large number of parameters, making it computationally expensive to train and deploy, especially on resource-constrained devices. Training deep models like VGGNet may require significant computational resources, including high-performance GPUs or TPUs, and lengthy training times.
2. **Overfitting**: Deep neural networks like VGGNet are prone to overfitting, especially when trained on small or imbalanced datasets. Overfitting occurs when the model learns to memorize the training data rather than generalize to unseen examples, leading to poor performance on new data. Regularization techniques such as dropout and weight decay may be necessary to mitigate overfitting.
3. **Limited Interpretability**: While VGGNet can achieve high accuracy in image classification tasks, its deep architecture makes it challenging to interpret how the model arrives at its predictions. Understanding which features or patterns the network is using for classification can be difficult, limiting its interpretability and potentially hindering trust and adoption by healthcare professionals.
4. **Data Requirements**: Training deep neural networks like VGGNet requires large amounts of annotated data. Obtaining high-quality labeled data

Challenges:

1. **Data Quality and Quantity**: Obtaining high-quality labeled data for skin cancer detection can be challenging. Annotating medical images requires expertise, and datasets may suffer from inter-observer variability in annotations. Additionally, collecting a sufficiently large and diverse dataset to train a deep learning model effectively can be time-consuming and resource-intensive.
2. **Class Imbalance**: Skin cancer datasets often suffer from class imbalance, with benign lesions outnumbering malignant ones or vice versa. Class imbalance can lead to biased models that perform poorly on the minority class. Addressing class imbalance through techniques like data augmentation, class weighting, or resampling strategies is crucial for improving model performance.
3. **Overfitting**: Deep neural networks like VGGNet are prone to overfitting, especially when trained on small or noisy datasets. Overfitting occurs when the model learns to memorize the training data rather than generalize to unseen examples, leading to poor performance on new data. Regularization techniques such as dropout, weight decay, or early stopping may be necessary to mitigate overfitting.

2.2.3 Implementation :

Implementing skin cancer detection using VGGNet involves several steps, including data preparation, model construction, training, evaluation, and deployment. Below is a high-level overview of how you can implement this:

1. **Data Preparation:**
    - Collect a dataset of skin lesion images labeled with their corresponding diagnoses (benign or malignant). You can use publicly available datasets like ISIC or Dermofit.

2. **Training:**
    - Split your dataset into training, validation, and test sets. The training set is used to update the model's weights, the validation set is used to tune hyperparameters, and the test set is used to evaluate the final model.
    - Train the model using a suitable optimization algorithm (e.g., stochastic gradient descent) and a loss function appropriate for binary classification (e.g., binary cross-entropy).
    - Monitor the model's performance on the validation set and adjust hyperparameters (e.g., learning rate, batch size) as needed to prevent overfitting.
3. **Evaluation:**
    - Evaluate the trained model on the test set to assess its performance. Compute metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC) to measure the model's effectiveness in skin cancer detection.
    - Visualize the model's predictions and examine misclassified examples to gain insights into areas for improvement.
4. **Deployment:**
    - Deploy the trained model for real-world use. This could involve integrating it into a web or mobile application where users can upload images of skin lesions for analysis.
    - Ensure that the deployed model is accompanied by appropriate documentation and disclaimers regarding its limitations and the importance of consulting a medical professional for diagnosis and treatment.

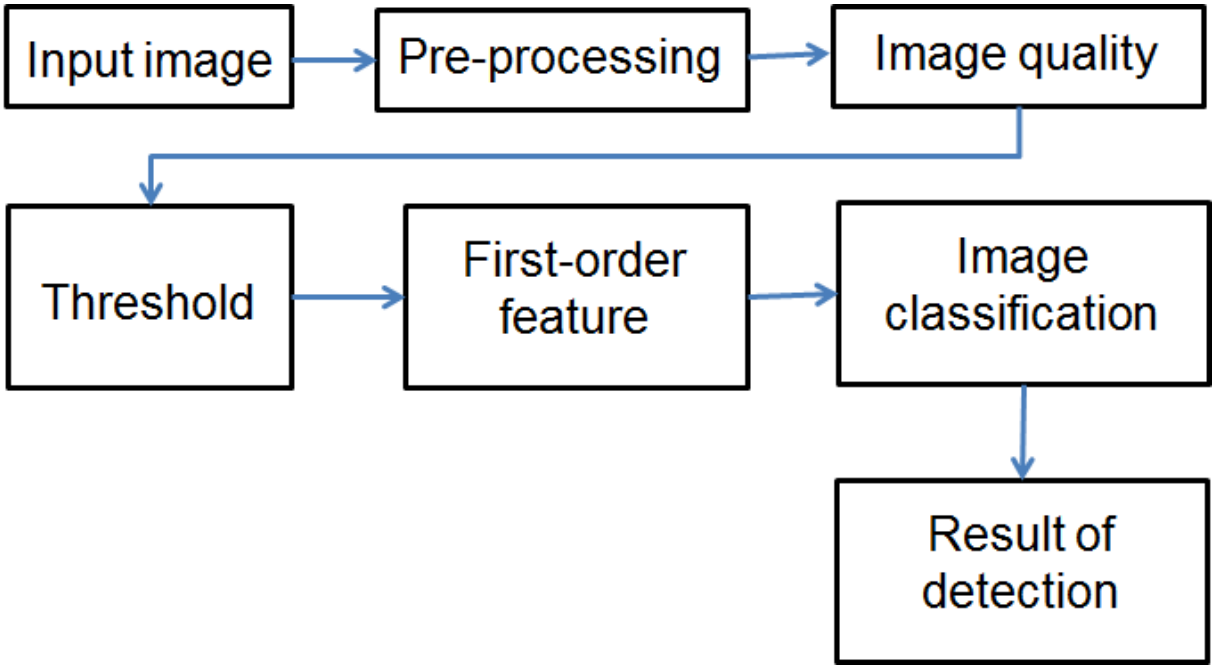


Fig* 2.2.3.1 *: vggnet implementation

2.3 Skin cancer classification using transfer learning

2.3.1 Introduction

Skin cancer is one of the most prevalent types of cancer globally, with its incidence steadily rising over the past few decades. Early detection and accurate classification of skin lesions are critical for effective treatment and prognosis. While dermatologists possess remarkable expertise in diagnosing skin cancer, the growing demand for healthcare services necessitates the development of automated systems to assist in diagnosis[4]

In recent years, advancements in machine learning and computer vision techniques have paved the way for the development of automated skin cancer classification systems. Among these techniques, Support Vector Machines (SVM) and transfer learning have emerged as powerful tools for building robust and accurate classification models[4].

Transfer learning, on the other hand, leverages knowledge gained from training a model on one task and applies it to a different but related task. In the context of skin cancer classification, transfer learning involves using pre-trained convolutional neural network (CNN) models, which have been trained on large-scale image datasets like ImageNet, and fine-tuning them for the specific task of classifying skin lesions

In this paper, we propose a skin cancer classification framework that combines the strengths of SVM and transfer learning. We first utilize transfer learning to extract meaningful features from skin lesion images using a pre-trained CNN. These features are then fed into an SVM classifier to make predictions regarding the malignancy of the lesions.

2.3.2 Merits and Demerits

Merits :

1. **Utilization of Pre-trained Models**: Transfer learning allows leveraging pre-trained CNN models that have been trained on large datasets like ImageNet. This approach harnesses the generalization capabilities of the pre-trained model, which has learned rich hierarchical representations of visual features.
2. **Feature Extraction**: Transfer learning enables the extraction of high-level features from skin lesion images without the need for manual feature engineering. This saves time and effort while potentially capturing complex patterns present in the data more effectively than handcrafted features.
3. **Improved Performance**: By combining the discriminative power of SVMs with the rich feature representations learned by CNNs, the classification model can achieve high accuracy in distinguishing between different types of skin lesions. This is particularly beneficial in medical diagnosis, where accurate classification is crucial for patient outcomes.
4. **Reduced Overfitting**: Pre-trained CNN models have been trained on large and diverse datasets, which helps in learning generic features that generalize well to new tasks. Transfer learning from these models can mitigate the risk of overfitting, especially when dealing with limited training data, by leveraging the pre-trained model's ability to capture relevant image features.
5. **Interpretability (to some extent)**: While deep learning models are often criticized for their lack of interpretability, SVMs offer some level of interpretability, as the decision boundary is defined by a subset of the training data points (support vectors). This can provide insights into which features contribute most to the classification decision, aiding in understanding the model's behavior

Demerits :

1. **Limited Flexibility**: SVMs have limited flexibility compared to deep learning models like CNNs. SVMs rely on manually engineered features, which may not capture the complex patterns present in skin lesion images as effectively as deep learning approaches. This can result in suboptimal performance, especially when dealing with highly heterogeneous or noisy data.
2. **Feature Engineering Overhead**: SVMs require careful feature engineering to extract relevant information from the input data. While transfer learning helps in extracting features from pre-trained CNN models, additional feature engineering may still be necessary to ensure optimal performance of the SVM classifier. This process can be time-consuming and may require domain expertise.
3. **Limited Generalization**: Transfer learning relies on the assumption that features learned from the source task (e.g., ImageNet classification) are transferable to the target task (skin cancer classification). However, features learned from generic image classification tasks may not always generalize well to the specific characteristics of skin lesion images. This can lead to overfitting or poor generalization performance, especially when the target task has significantly different data distributions or characteristics compared to the source task.

Challenges:

Addressing these challenges requires careful consideration of dataset selection, model architecture, hyperparameter tuning, and evaluation metrics. Additionally, advancements in transfer learning techniques and model interpretability may help mitigate some of these challenges in the future.

2.3.3 Implementation

1. **Data Collection and Preprocessing**:
    - Gather a dataset of skin lesion images with corresponding labels indicating the type of skin cancer (e.g., melanoma, basal cell carcinoma, etc.).
    - Preprocess the images, including resizing them to a uniform size, normalizing pixel values, and potentially augmenting the data to increase variability and robustness.
2. **Transfer Learning**:
    - Choose a pre-trained CNN model (e.g., VGG, ResNet, Inception) pre-trained on a large dataset like ImageNet.
    - Remove the fully connected layers at the top of the pre-trained model, as they are specific to the original classification task.
    - Add new fully connected layers suitable for skin cancer classification, with the output layer having nodes corresponding to the number of classes in your dataset.
    - Optionally, freeze the weights of the pre-trained layers to prevent them from being updated during training, although fine-tuning these layers with a lower learning rate may improve performance.
3. **Feature Extraction**:
    - Use the pre-trained CNN to extract features from the skin lesion images. Pass each image through the CNN and extract the output of one of the intermediate layers as features.
    - Flatten or average the feature maps to obtain a feature vector for each image.
4. **Training SVM Classifier**:
    - Train an SVM classifier using the extracted features as input and the corresponding labels as targets.
    - Tune the hyperparameters of the SVM classifier (e.g., kernel type, regularization parameter) using cross-validation to optimize performance.

Integrate the implemented method into existing medical image analysis workflows or develop a standalone software tool for clinical use. Ensure compatibility with standard image formats and provide user-friendly interfaces for data input, parameter adjustment, and result visualization. Throughout the implementation process, maintain documentation and code organization to facilitate reproducibility, collaboration, and future enhancements. Continuous testing and validation against diverse datasets are essential to ensure method robustness and reliability across different clinical scenarios.

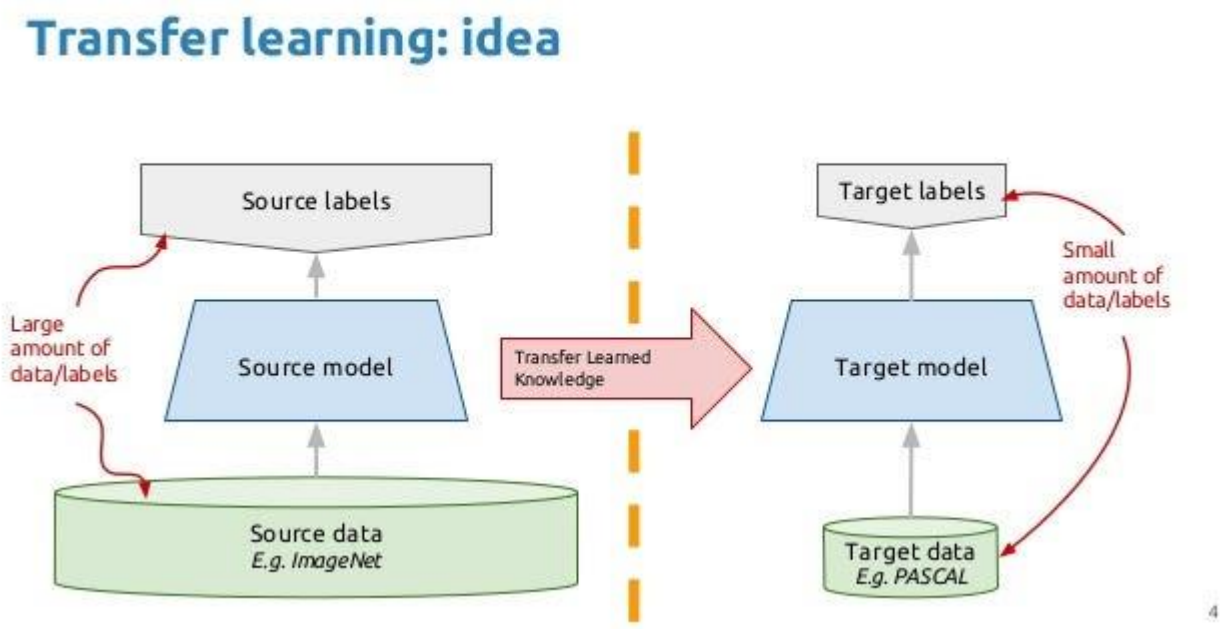


Fig* 2.3.3.1 : *transfer learning implementation

CHAPTER 3
PROPOSED SYSTEM

CHAPTER 3

PROPOSED SYSTEM

3.1 Objective of the Proposed System

Building a malignancy identification system using CNN is a widely embraced learning implementations within healthcare In the preexisting frameworks three diseases it can manage but in the proposed system it can handle approxiametly 7 diseases like basel cell,dermatofibroma,actinickeratoses,,beingkeratosis,.melonoma.melanocyticnevi.To implement this Gather a large dataset of skin diseases images of ham10000 Each image should be Resize and normalize the images to ensure consistency in the source feed Augment the dataset by applying transformations like rotation, flipping, and zooming to bolster the model's adaptability.. A typical CNN architecture consists of Filter layers

3.2 Algorithms Used for Proposed Model

3.2.1 CNN Algorithm

A Convolutional Neural Network (CNN) is a type of deep learning model specifically designed for processing structured grid-like data, such as images. CNNs have become the cornerstone of many computer vision tasks, including image classification, object detection, segmentation, and more. Here's an overview of the CNN algorithm. By iteratively adjusting the architecture, hyperparameters, and training data, CNNs can learn hierarchical representations of visual features and achieve state-of-the-art performance on various computer vision tasks.

1.Input Layer:

- CNNs take images as input, where each image is represented as a grid of pixel values. The dimensions of the input images are typically specified (e.g., height, width, number of channels

2 .Convolutional Layers:

- The convolutional layers are the core building blocks of CNNs. They consist of multiple learnable filters (also called kernels or convolutional kernels) that slide over the input image to perform convolution operations.
- Each filter extracts features from the input image by computing element-wise multiplications between the filter weights and the corresponding pixel values in the receptive field of the input image.
- The output of each convolutional operation is referred to as a feature map, which captures different aspects of the input image (e.g., edges, textures, patterns).

3.Activation Function:

- After each convolutional operation, an activation function is applied element-wise to introduce non-linearity into the network. Common choices include Rectified Linear Unit (ReLU), sigmoid, or hyperbolic tangent (tanh) functions.

4. Pooling Layers:

- Pooling layers are used to reduce the spatial dimensions (width and height) of the feature maps while retaining the most important information.
- Max pooling is the most commonly used pooling operation, where the maximum value within each pooling region is retained, effectively downsampling the feature maps.

5.Fully Connected Layers:

- The fully connected layers are typically placed at the end of the CNN architecture and serve as the classifier or regressor.
- These layers connect every neuron in one layer to every neuron in the next layer, forming a densely connected network.
- The output of the fully connected layers is passed through an activation function (e.g., softmax for classification tasks) to produce the final output probabilities or values.

6.Training:

- CNNs are trained using backpropagation, where gradients of the loss function with respect to the network parameters are computed and used to update the parameters iteratively.
- Training is typically per formed on batches of input data to improve computational efficiency and convergence speed.

7.Evaluation:

- After training, the performance of the CNN model is evaluated on a separate validation or test set using evaluation metrics specific to the task (e.g., accuracy, precision, recall, F1-score).

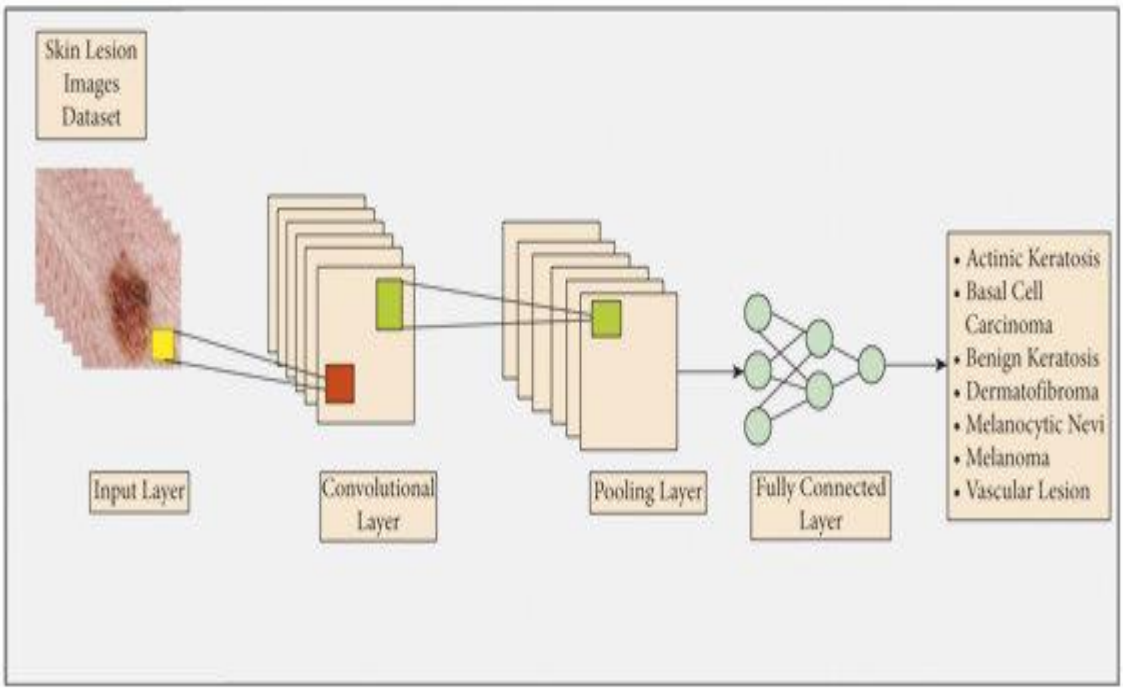


Fig 3.2.1.1 : Proposed CNN Architecture

3.3 Designing

3.3.1 UML Diagram

Use case Diagram :

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML

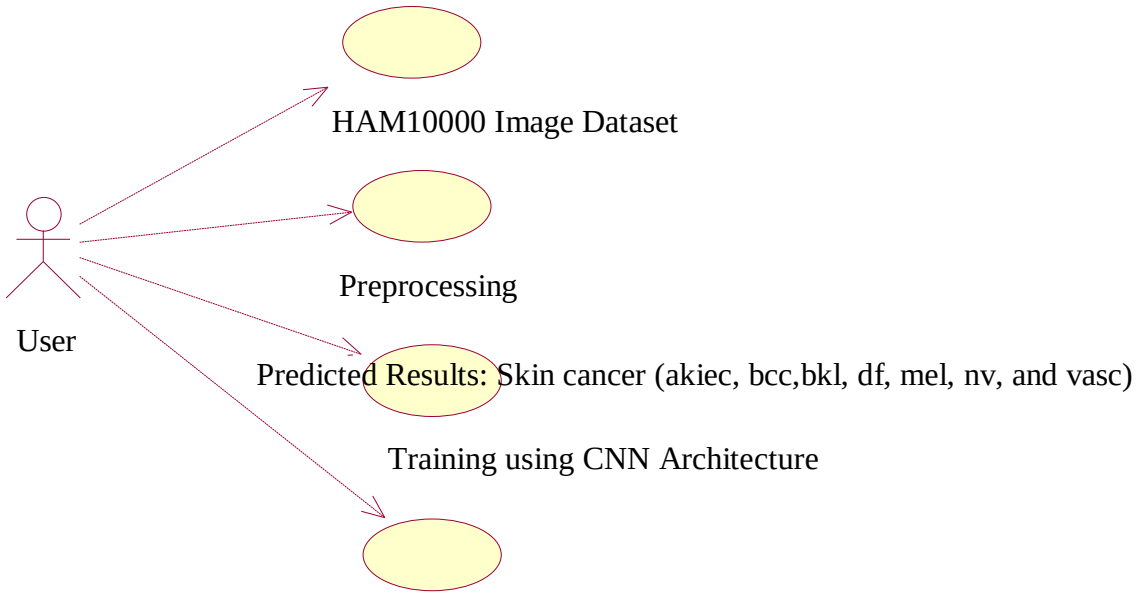


Fig* 3.3.1.1 *: Use Case Diagram

Sequence Diagram :

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

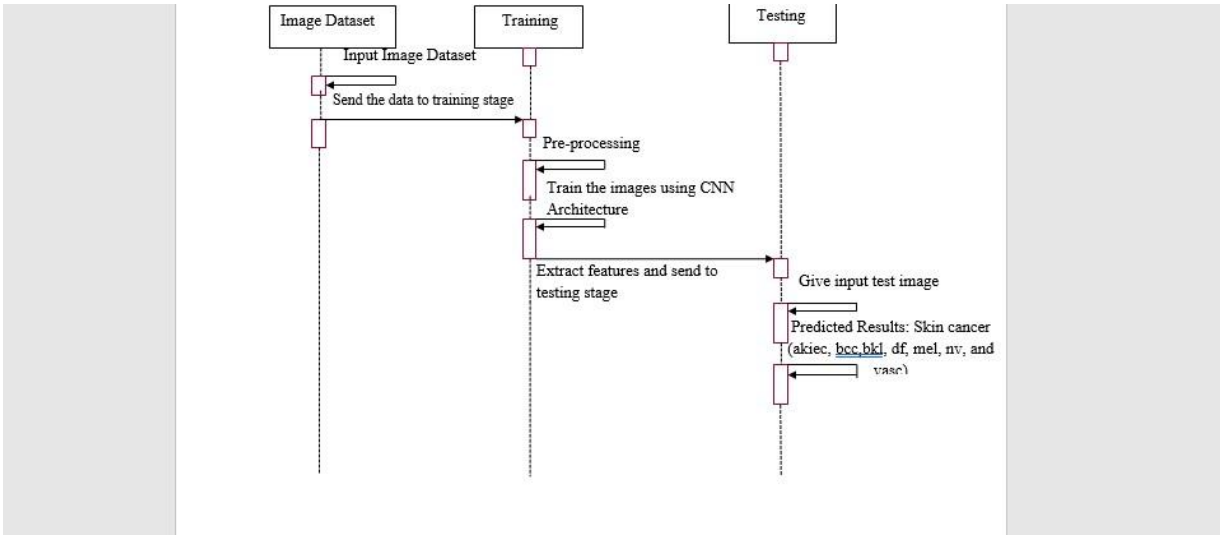


Fig* 3.3.1.2 *: Sequence Diagram

Activity Diagram :

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

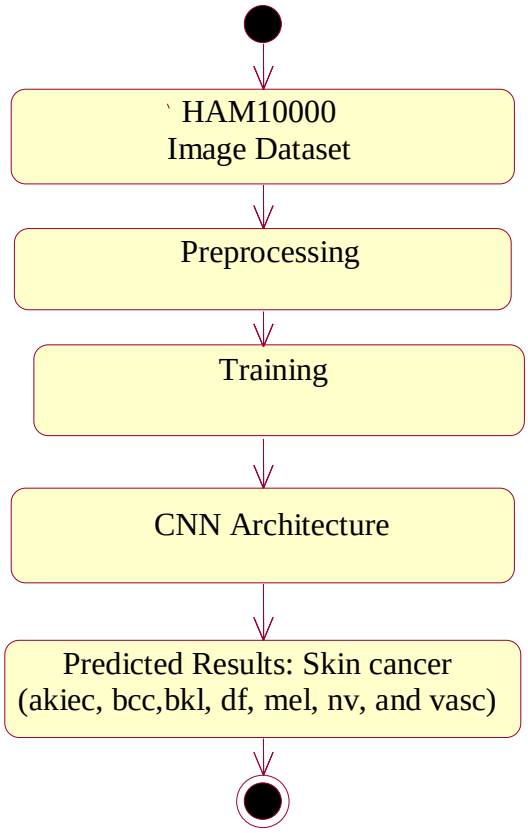


Fig* 3.3.1.2 *: Activity Diagram

3.4 Stepwise Implementation and Code

Dataset:

In the first module of Skin Cancer Prediction Using Deep Learning Techniques, we developed the system to get the input dataset. Data collection process is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get; the better our model will perform. There are several techniques to collect the data, like web scraping, manual interventions. Our dataset is placed in the project and it's located in the model folder. The dataset is referred from the popular standard dataset repository kaggle where all the researchers refer it. The dataset consists of 10,015 images. The following is the URL for the dataset referred from kaggle.

Importing the necessary libraries:

We will be using Python language for this. First we will import the necessary libraries such as keras for building the main model, sklearn for splitting the training and test data, PIL for converting the images into array of numbers and other libraries such as pandas, numpy, matplotlib and tensorflow.

Retrieving the images:

In this module we will retrieve the images from the dataset and convert them into a format that can be used for training and testing the model. This involves reading the images, resizing them, and normalizing the pixel values. We will retrieve the images and their labels.

Then resize the images to (28, 28) as all images should have same size for recognition. Then convert the images into numpy array.

This module involves loading and retrieving the preprocessed images and their corresponding class labels from the dataset. The images will be utilized as input to the deep learning model for training and evaluation.

Splitting the dataset:

In this module, the image dataset will be divided into training and testing sets. Split the dataset into Train and Test. 80% train data and 20% test data. This will be done to train the model on a subset of the data, validate the model's performance, and test the model on unseen data to evaluate its accuracy. Split the dataset into train and test. 80% train data and 20% test data.

To train and evaluate the model effectively, the dataset needs to be split into training, validation, and test sets. This module divides the dataset into these subsets, ensuring that the model is trained on one portion, validated on another, and finally tested on unseen data to assess its generalization performance.

Building the model:

The concept of convolutional neural networks are very successful in image recognition. The key part to understand, which distinguishes CNN from traditional neural networks, is the convolution operation. Having an image at the input, CNN scans it many times to look for certain features. This scanning (convolution) can be set with 2 main parameters: stride and padding type. As we see on below picture, process of the first convolution gives us a set of new frames, shown here in the second column (layer). Each frame contains an information about one feature and its presence in scanned image. Resulting frame will have larger values in places where a feature is strongly visible and lower values where there are no or little such features. Afterwards, the process is repeated for each of obtained frames for a chosen number of times. In this project I chose a classic CNN model which contains only two convolution layers.

The latter layer we are convolving, the more high-level features are being searched. It works similarly to human perception. To give an example, below is a very descriptive picture with features which are searched on different CNN layers. As you can see, the application of this model is face recognition. You may ask how the model knows which features to seek. If you construct the CNN from the beginning, searched features are random. Then, during training process, weights between neurons are being adjusted and slowly CNN starts to find such features which enable to meet predefined goal, i.e. to recognize successfully images from the training set.

Between described layers there are also pooling (sub-sampling) operations which reduce dimensions of resulted frames. Furthermore, after each convolution we apply a non-linear function (called **ReLU**) to the resulted frame to introduce non-linearity to the model.

Eventually, there are also fully connected layers at the end of the network. The last set of frames obtained from convolution operations is flattened to get a one-dimensional vector of neurons. From this point we put a standard, fully-connected neural network. At the very end, for classification problems, there is a softmax layer. It transforms results of the model to probabilities of a correct guess of each class

CODE

```
from flask import Flask, render_template, request
import tensorflow as tf
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.metrics import AUC
import numpy as np
app = Flask(__name__)
dependencies = {
   'auc_roc': AUC
}
verbose_name = {
0: 'Actinic keratoses and intraepithelial carcinomae',
1: 'Basal cell carcinoma',
2: 'Benign keratosis-like lesions',
3: 'Dermatofibroma',
4: 'Melanocytic nevi',
5: 'Pyogenic granulomas and hemorrhage',
6: 'Melanoma',
        }
model = load_model('skin.h5')

def predict_label(img_path):
        test_image = image.load_img(img_path, target_size=(28,28))
        test_image = image.img_to_array(test_image)/255.0
        test_image = test_image.reshape(1, 28,28,3)

        predict_x=model.predict(test_image)
        classes_x=np.argmax(predict_x,axis=1)

        return verbose_name[classes_x[0]]


@app.route("/")
@app.route("/first")
def first():
        return render_template('first.html')

@app.route("/login")
def login():
        return render_template('login.html')

@app.route("/index", methods=['GET', 'POST'])
```

```
def get_output():
        if request.method == 'POST':
                img = request.files['my_image']

                img_path = "static/tests/" + img.filename
                img.save(img_path)

                predict_result = predict_label(img_path)

        return render_template("prediction.html", prediction = predict_result, img_path =
img_path)



@app.route("/Graph")
def Graph():
        return render_template('Graph.html')

@app.route("/chart")
def chart():
        return render_template('chart.html')


if __name__ =='__main__':
        app.run(debug = True)
```

```
import numpy as np
import pandas as pd
Data Loading
data = pd.read_csv("D:\skin cancer\hmnist_28_28_RGB.csv")
data.head()
meta_df = pd.read_csv("D:\skin cancer\HAM10000_metadata.csv")
meta_df.head()
meta_df.shape
(10015, 7)
data.shape
(10015, 2353)
yData = data['label']
XData = data.drop(columns = ['label'])
EDA
distribution = meta_df['dx'].value_counts()
distribution
Name: count, dtype: int64
# classes = {4: ('nv', ' melanocytic nevi'),
#           6: ('mel', 'melanoma'),
#           2 :('bkl', 'benign keratosis-like lesions'),
#           1:('bcc' , ' basal cell carcinoma'),
#           5: ('vasc', ' pyogenic granulomas and hemorrhage'),
#           0: ('akiec', 'Actinic keratoses and intraepithelial carcinomae'),
#           3: ('df', 'dermatofibroma')}
dist = data['label'].value_counts()
dist
label
```

Fixing Data Imbalance

over sample the dateset using Random Over Sampler

```
from imblearn.over_sampling import RandomOverSampler
sampler = RandomOverSampler()
XData,yData  = sampler.fit_resample(XData,yData)
XData.shape, yData.shape
((46935, 2352), (46935,))
XData = np.array(XData).reshape((-1, 28, 28, 3))
XData = XData / 255
XData.shape
(46935, 28, 28, 3)
```

Create the train and validation set

```
from sklearn.model_selection import train_test_split
Xtrain, Xtest, Ytrain, Ytest = train_test_split(XData,yData, test_size=0.2)
Xtrain.shape, Xtest.shape
((37548, 28, 28, 3), (9387, 28, 28, 3))
```

Train the model

```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
import tensorflow as tf
img_width, img_height = 28,28
input_shape = (img_width, img_height, 3)
model = Sequential()
model.add(Conv2D(32, (2, 2), input_shape=input_shape))
model.add(Activation('swish'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
model.add(Conv2D(32, (2, 2)))
model.add(Activation('swish'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (2, 2)))
model.add(Activation('swish'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(64))
model.add(Activation('swish'))
model.add(Dropout(0.5))
model.add(Dense(7))
model.add(Activation('softmax'))
model.compile(loss='sparse_categorical_crossentropy',
optimizer='nadam',
metrics=['accuracy'])
callback = tf.keras.callbacks.ModelCheckpoint(filepath='skin.h5',
                                              monitor='val_acc', mode='max',
                                             verbose=1)
early_stopping = tf.keras.callbacks.EarlyStopping(patience=10, restore_best_weights=True)
history = model.fit(Xtrain,
                    Ytrain,
                    epochs = 100,
                    validation_data = (Xtest, Ytest),
                    callbacks=[callback, early_stopping])
```

CHAPTER 4
RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

In skin cancer detection using CNNs, The model's efficacy is typically evaluated using various metrics to assess its accuracy and effectiveness. Below are metrics and results that are often considered Precission

among all the samples predicted as positive. It is computed as the proportion of correctly identified positive predictions relative to the aggregate of true positives and false positives. Recall gauges the model's proficiency in accurately pinpointing positive samples amidst all genuine positive instances.

4.1 Output screens

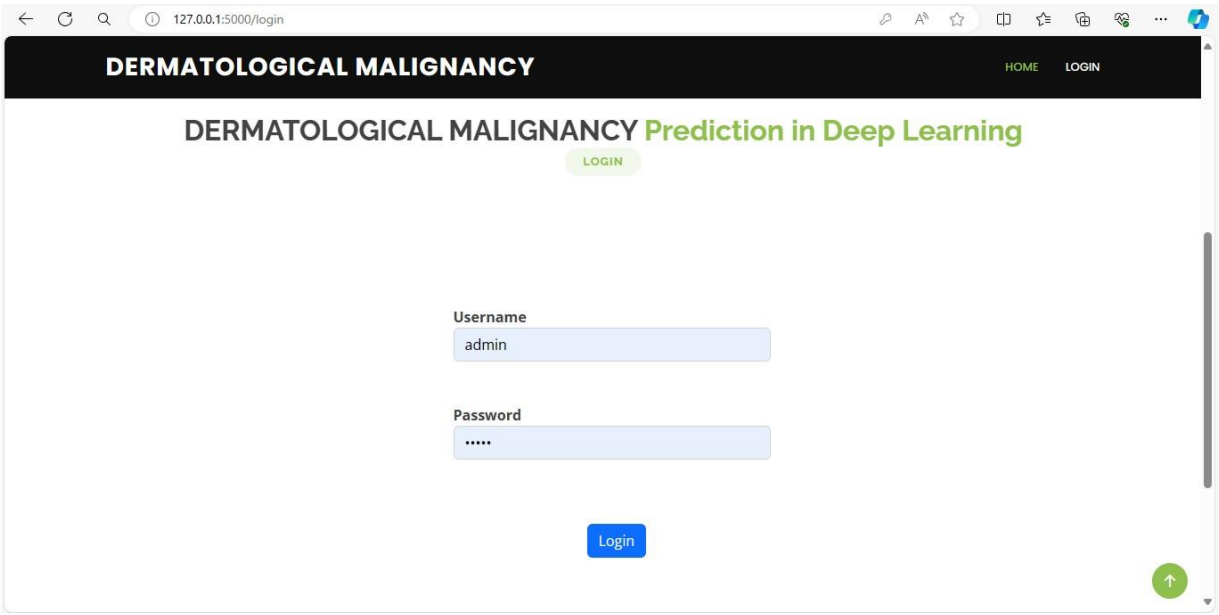


Fig* 4.1.1 *: login page

In the above 4.1.1 image it is the login page were it will make the website secure and unauthorized people cant login which make the portal very secure and use and it is very easy to use

above figure 4.1.2 figure where we have a web portal of

Choose file when we press the button preview where we can search Human Against Machine with 10000 images dataset trained dataset of skin diseases which you need to find the images

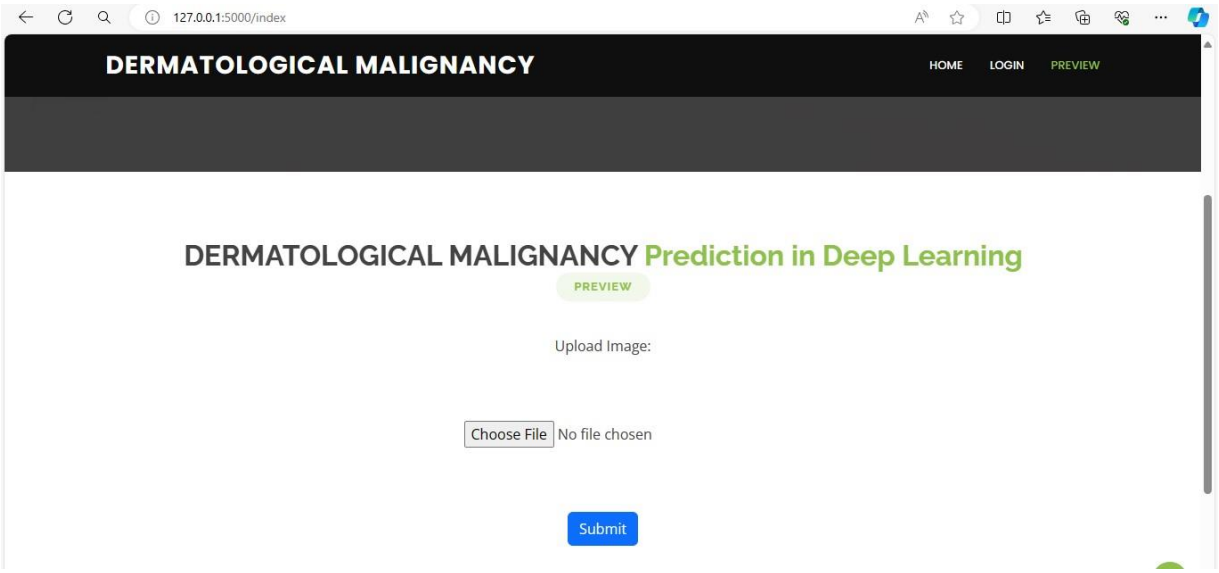


Fig* 4.1.2 *: choose file

In the above figure 4.1.3 figure where we have a web portal of

Choose file when we press the button preview where we can search the image in the data or trained dataset of skin diseases which you need to find the randiomics and upload in that folder

In the above figure 4.1.4 figure where we have a web portal of

Choose file when we press the button preview where we can search the image in or trained dataset of skin diseases which you need to find the randiomics sets and upload in that browser later it will use cnn procedure for locating which disease it is and it will give the name of which disease it is with the exactness score

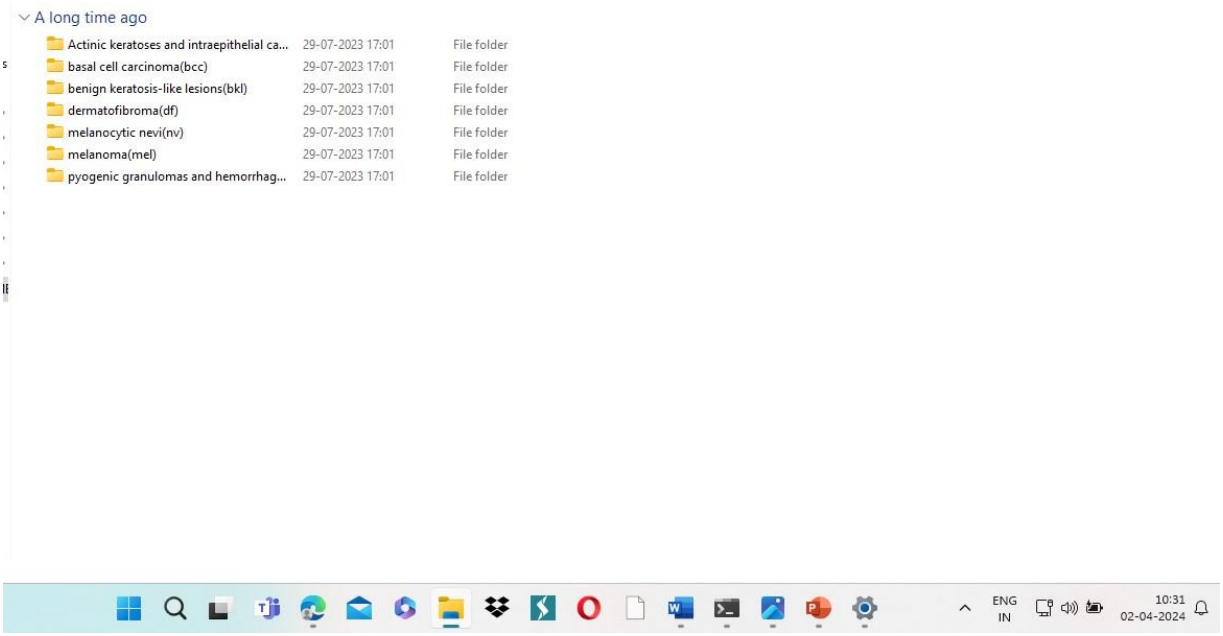


Fig* 4.1.3 *: choose image

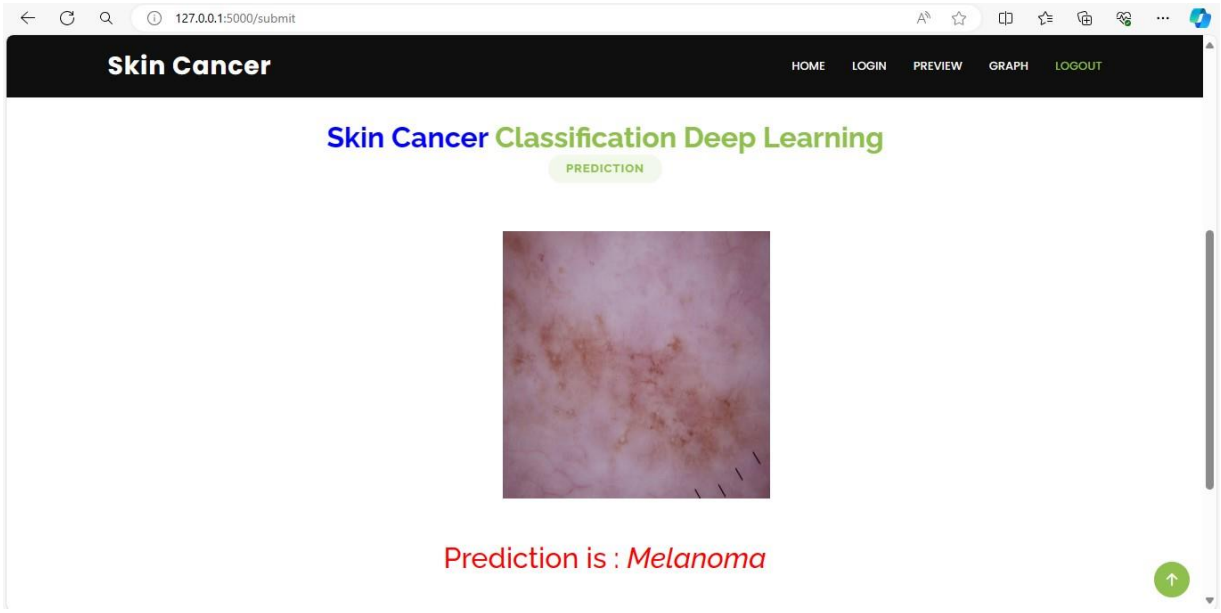


Fig* 4.1.4 *: disease detection

4.2 Performance metrics :

Performance metrics for skin cancer detection using Convolutional Neural Networks (CNNs) typically include:

1. **Accuracy**: The proportion of correctly classified images out of the total number of images.
   Accuracy=Number of correctly classified imagesTotal number of imagesAccuracy=Total number of imagesNumber of correctly classified images
2. **Precision**: The ratio of true positive predictions to the total number of positive predictions. It measures the classifier's ability to not label a negative sample as positive.

METHOD	DATA SET	PERFORMANCE
vggnet	isic	73%
Transfer learning	scd	80%
cnn	Ham10000	94%

Table* 4.2.1 *: performance metrices of vggnet,Transfer learning, cnn

CHAPTER 5
CONCLUSION

CHAPTER 5

CONCLUSION AND FEATURE WORK

CONCLUSION

In conclusion, the proposed system leverages the power of convolutional neural networks and incorporates various enhancements to improve accuracy, efficiency, and generalization capabilities. Our proposed system achieved training accuracy of 91.00% and validation accuracy of 92.00%. Through rigorous data preprocessing and dataset organization, the project effectively handles dermatoscopic images from diverse populations and various skin cancer types

FEATURE WORK

Larger and Diverse Datasets: Expanding the dataset to include a larger number of dermatoscopic images from diverse populations and skin types can enhance the model's ability to generalize across different patient demographics. Access to more comprehensive datasets would ensure better representation of rare skin cancer types and aid in building a more robust and reliable predictive system.

REFERENCES

REFERENCES

[1] S. Prasher, L. Nelson and S. Gomathi, "Resnet 50 Based Classification Model for Skin Cancer Detection Using Dermatoscopic Images," 2023 3rd Asian Conference on Innovation in Technology (ASIANCON), Ravet IN, India, 2023,pp. 1-5, doi: 10.1109/ASIANCON58793.2023.10270158.

[2] Nourabuared, “Skin Cancer Based on VGG19 and Transfer Learning”, In 3rd International Conference on Signal Processing and Information Security IEEE Conference, pp 1-4, DOI

[3].N. C. Lynn and Z. M. Kyu, , Taipei, Taiwan, 2017, pp. 117-122, DOI: 10.1109/PDCAT.2017.00028.

[4] .Harikrishna, “Skin Cancer Classification using Transfer Learning IEEE International Conference on Advent Trends in Multidisciplinary Research and Innovation (ICATMRI-2020)

[5].N. C. Lynn and Z. M. Kyu, "Segmentation and Classification of Skin Cancer Melanoma from Skin Lesion Images," 2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies, Taipei, Taiwan, 2017, pp. 117-122, DOI: 10.1109/PDCAT.2017.00028.

GITHUB LINK:
DOI LINK: https://doi.org/10.22214/ijraset.2024.59154

iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** III **Month of publication:** March 2024

DOI: https://doi.org/10.22214/ijraset.2024.59154

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

Dermatological Malignancy Detection Using CNN

Ms. Komal Parashar[1], M. V. Devendranathreddy[2], M. Deepak Reddy[3], S. Jashwitha[4]
[1]*Assistant Professor, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana*
[2,3,4] *UG Student, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana*

Abstract: ***Skin cancer is a prevalent and potentially life-threatening disease that continues to pose a significant public health concern. Starting stage detection Plays a crucial part in enhancing treatment efficacy. Furthermore, the advancement of deep learning has enabled the creation of automated frameworks for detecting maligant cancer. This proposed system utilizes a varied dataset encompassing various types of skin cancer, such as malignant melanomas and benign nevi, to educate a learning model. Utilizing CNN, pertinent features are automatically extracted from skin cancer images, facilitating precise classification. Assessment of the suggested system showcases encouraging outcomes concerning sensitivity, specificity, and overall precision. Comparative assessments against conventional techniques underscore the superior efficacy and effectiveness of the devised model in identifying potential skin cancer instances. Moreover, the system's interpretive capacity is heightened through the integration integration of attention mechanisms offering insights into the areas of focus within the skin cancer images***

Keywords: python, Ham1000 dataset, CNN, melanoma, basal cell, skin cancer, malignant,

I. INTRODUCTION

Skin cancer can be dangerous, and its severity relies on the hinge of skin cancer, its stage at the time of diagnosis, and how quickly it is treated. basal cell, carcinoma , squamous cell carcinoma , and melanoma[3] etc are top skin diseases.personal Detecting skin maligant using CNN is a common and effective approach with in the realm of randiomics. CNNs are well-suited in the objective like classification, making them suitable for identifying patterns and features in medical images. Here's a general outline of the steps you might follow to create a maligancy identity system using CNN. Improve the chances of successful treatment and reduce the risk of complications associated with advanced stages. Enhance accessibility to skin cancer detection, especially region featuring limited access to dermatologists or specialized healthcare services. Contribute to public health efforts by facilitating large-scale screenings and early interventions., which are self-executing contracts with predefined conditions, By achieving these objectives, the use of CNNs in detection aims to make a positive impact on patient outcomes, healthcare efficiency, and public health initiatives related to skin cancer awareness and prevention.

II. RELATED WORK

Skin cancer detection is an active area of research, with extensive investigations aimed at devising precise and efficient approaches for early identification . Below are some notable works and approaches in the realm of cancer detection. pioneering groundbreaking advancements in skin randiomics technologies and methodologies has provided a dataset that researchers use for benchmarking their algorithms. This dataset includes a Multifariousnessskin lesion images with corresponding diagnoses. It's essential to check for the latest research publications and developments in the field, as advancements in technology and methodologies continue to shape the landscape of maligent detection[5].

A. Skin Cancer Classification using Resnet

Skin cancer detection using ResNet involves employing a learning model, specifically ResNet (Residual Neural Network), to analyze and classify skin lesions as either benign or malignant network Evaluate the data model on the test the dataset set to assess its generalization performance. Metrics like exactitude, exactness, retrieval rate , and F1 score for the proficiency the model's It's significant name of the success of dermatological malignancy detection using ResNet reliable on quality and diversity of the dataset, appropriate hyperparameter tuning, and magnitude of the model. Continuous monitoring and updates may be necessary to adapt to new data and enhance the model's performance over time[1].

B. Skin Cancer Classification using Vggnet

Obtain a dataset of dermatological cancer images. A ubicutious dataset for this task is reducing the structure of system. and fine-tuning it on a dataset of skin cancer images Resize images to the input size required by the VGGNet model (typically 224x224 pixels). Load the pre-trained VGGNet model.

You can use a pre-trained VGG16 or VGG19 model from a learning library like Keras or PyTorch. Train the modified VGGNet on your skin cancer dataset. Use the training for instruction and the validation set to monitor the model's performance and avoid overfitting. Fine-tune the model by adjusting turning parameters like lsteprate, grouping magnitude, and the number of epochs., Utilize metrics such as exactitude, exactness, retrieval rate to evaluate the model's effectiveness in skin cancer classification.[2]

C. Skin Cancer Classification using SVM and Transfer Learning

Dermatological cancer classification using svm and transfer learning: Detecting skin cancer using Support Vector Machines (SVM) involves training a classification learning data to classify skin cancer disease as either good or bad based on certain forward representation from images. Here's a simplified guide on how you can approach keeps a copy of the database, Collect a dataset of skin lesion images labeled as benign or malignant. Ensure that your dataset is diverse, representative, Extract relevant features from the images to feed into the SVM model. Easily used for radiomics include color histograms, texture features, and shape descriptors. partitioning the dataset into training and validation sets. The training set is employed for model training, while the validation set is utilized to assess its performance on unseen data., Evaluate the performance of. Adjust Once satisfied with the model's performance, deploy it to a system where it should be employed for cancer detection on Remember that this is a simplified guide, and Additional aspects to take into account are parameter adjustmen, cross-validation, and fine tuning the feature representation provides that can be explored to improve the model's performance.[4]

III. METHODS AND EXPERIMENTAL DETAILS

Building a malignancy identification system using CNN is a widely embraced learning implementations within healthcare In the preexisting frameworks three diseases it can manage but in the proposed system it can handle approxiamety 7 diseases like basel cell,dermatofibroma,actinickeratoses,,beingkeratosis,.melonoma.melanocyticnevi.To implement this Gather a large dataset of skin diseases images of ham10000 Each image should be Resize and normalize the images to ensure consistency in the source feed Augment the dataset by applying transformations like rotation, flipping, and zooming to bolster the model's adaptability.. A typical CNN architecture consists of Filter layers, Down sampling layers, and Dense layers ,Convolutional layers use filters to detect patterns and features in the input images. Pooling layers reduce the spatial dimensions, preserving the essential information. connected layers make decisions based on the features. Split the dataset into training, validation, and test sets privacy Train the CNN using the training set, adjusting the weights hinged upon the error (loss) calculated during each iteration.

Assess the todays efficacy on the evaluation dataset set using Evaluate the model's efficacy on the test set utilizing metrics like exactitude, exactness, retrieval rate, and measure-score Continuously monitor the proficency in the real-world setting. By following these steps, a CNN capable of discerning e patterns indicative of skin cancer from images, contributing to early detection and improved patient outcomes.

A. CNN Algorithm

CNNs, or convolutional neural networks, are a subset of A CNN's convolutional phase is its basic construction phase. With the aim of extract Extracting distinct characteristics from the input data at a local level, it applies a group of filters, also known as kernels. These filters create a responce map by swiping upon the randicoms and multiplying each element by the local region.

1) *Convolutional Layer:* The input volume's spatial dimensions are down sampled through the utilization for pooling layers. By taking the maximum value from a collection of values in a region, a typical approach called "max pooling" reduces the spatial size while preserving the most crucial information. Pooling Enhances the network further resilient and reduces processing.
2) *Pooling Layer:* The input volume's spatial dimensions are down sampled through the application of pooling layers. By taking the maximum value from a group of values in a region, a typical approach called "max pooling" reduces the spatial size while preserving the most crucial information. Pooling Augments the network's capabilities further resilient and reduces processing.
3) *Activation Function:* A mathematical operation that is applied to a neural network neuron's (or node's) output is known as an activation function. It provides the network non-linearities, which enable a. The An activation function uses its input, or group of inputs, to determine a neuron's output.
4) *Training:* With the intent of a (CNN) to generate accurate predictions on new, unseen data, the model's parameters (weights and biases) are optimized using a labeled dataset.
5) *Output Layer:* Reliant on the task at hand, the output layer's activation function (softmax is frequently utilized for multi-class

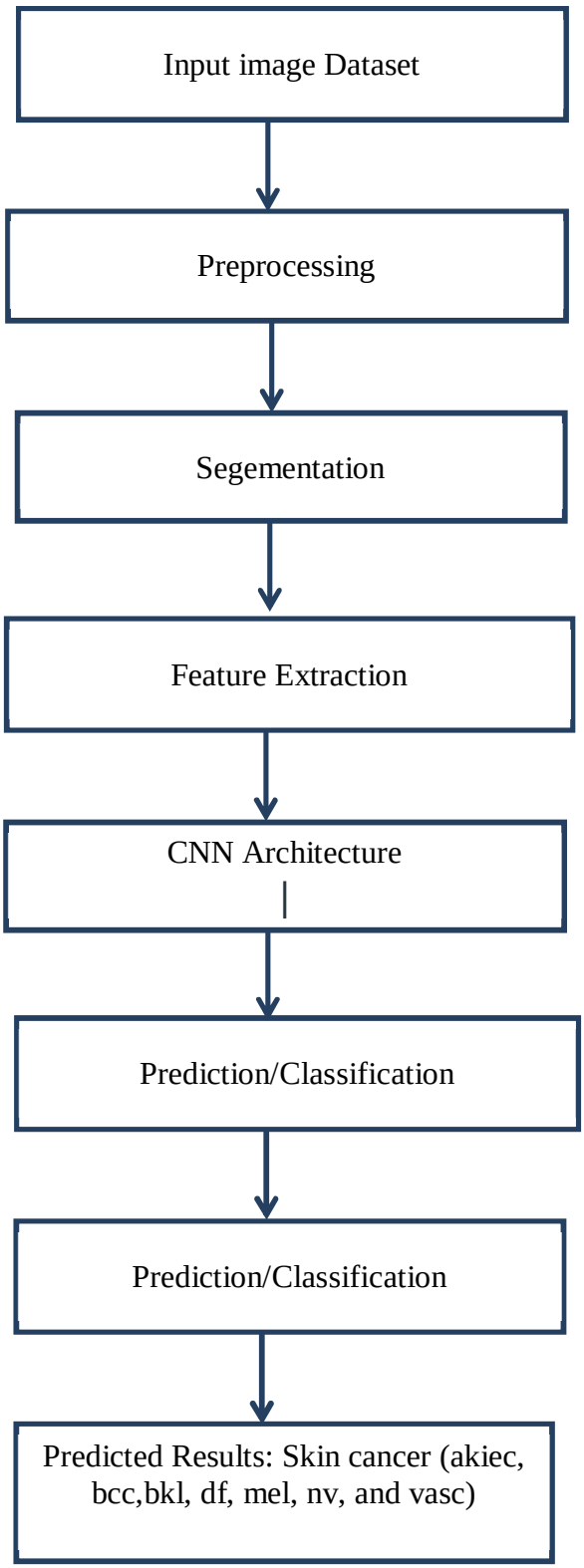


Fig 1.1-user interface architecture

A. Dataset

The HAM10000 dataset contains images of various skin lesions, and it includes multiple classes representing different skin diseases and conditions. The dataset includes the following disease classes:

1) mel: Melanoma
2) nv: Melanocytic nevus (common mole)
3) bkl: Benign keratosis-like lesions (seborrheic keratosis, solar lentigines, and lichen-planus-like keratosis)
4) bcc: Basal cell carcinoma
5) akiec: Actinic keratosis and intraepithelial carcinoma / Bowen's disease
6) vasc: Vascular lesions (angiomas, angiokeratomas, and pyogenic granulomas)
7) df: Dermatofibroma

Each class represents a specific skin disease or category. Researchers and practitioners use this dataset to develop and Assess the performance of learning models for the automated classification and diagnosis of these skin lesions. It's noteworthy that... that accurate diagnosis of skin diseases often requires a trained dermatologist, and Machine learning models are crafted to assist in The diagnostic process instead of supplanting professional medical judgment.

IV. RESULTS AND DISCUSSION

In skin cancer detection using CNNs, The model's efficacy is typically evaluated using various metrics to assess its accuracy and effectiveness. Below are metrics and results that are often considered Precission among all the samples predicted as positive. It is computed as the proportion of correctly identified positive predictions relative to the aggregate of true positives and false positives. Recall gauges the model's proficiency in accurately pinpointing positive samples amidst all genuine positive instances.

A. Login

Creating a login page in a dermatological malignancy detection system typically involves integrating user authentication and access control to ensure secure access to the application. Here's a simplified overview of how you might implement a login page for a cancer detection system

B. Preview

In the preview we should select image from the dataset and upload in the portal and we can get the output of which disease is it and we have the graph in which it shows how much accuracy it is there

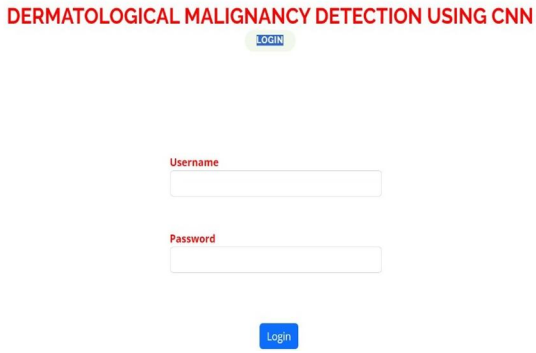


Fig 2.1-login page

Fig-2.2-image search

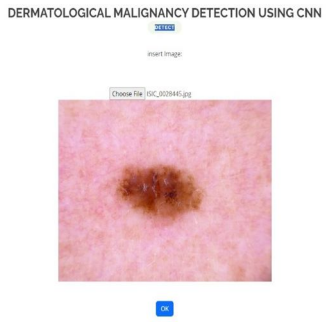


Fig.2.3-image upload

In the above 2.1 image it is the login page were it will make the website secure and unauthorized people cant login which make the portal very secure and use and it is very easy to use above figure 2.2 figure where we have a web portal of Choose file when we press the button preview where we can search Human Against Machine with 10000 images dataset trained dataset of skin diseases which you need to find the images

In the above figure 2.3 figure where we have a web portal of Choose file when we press the button preview where we can search the image in the data or trained dataset of skin diseases which you need to find the randiomics and upload in that folder

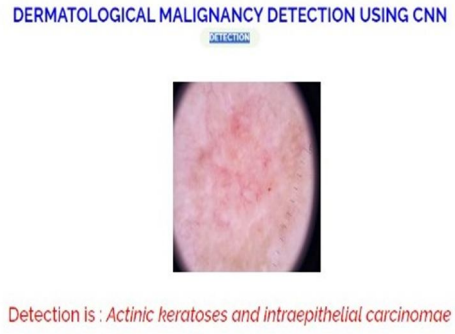


Fig-2.4 image disease detection

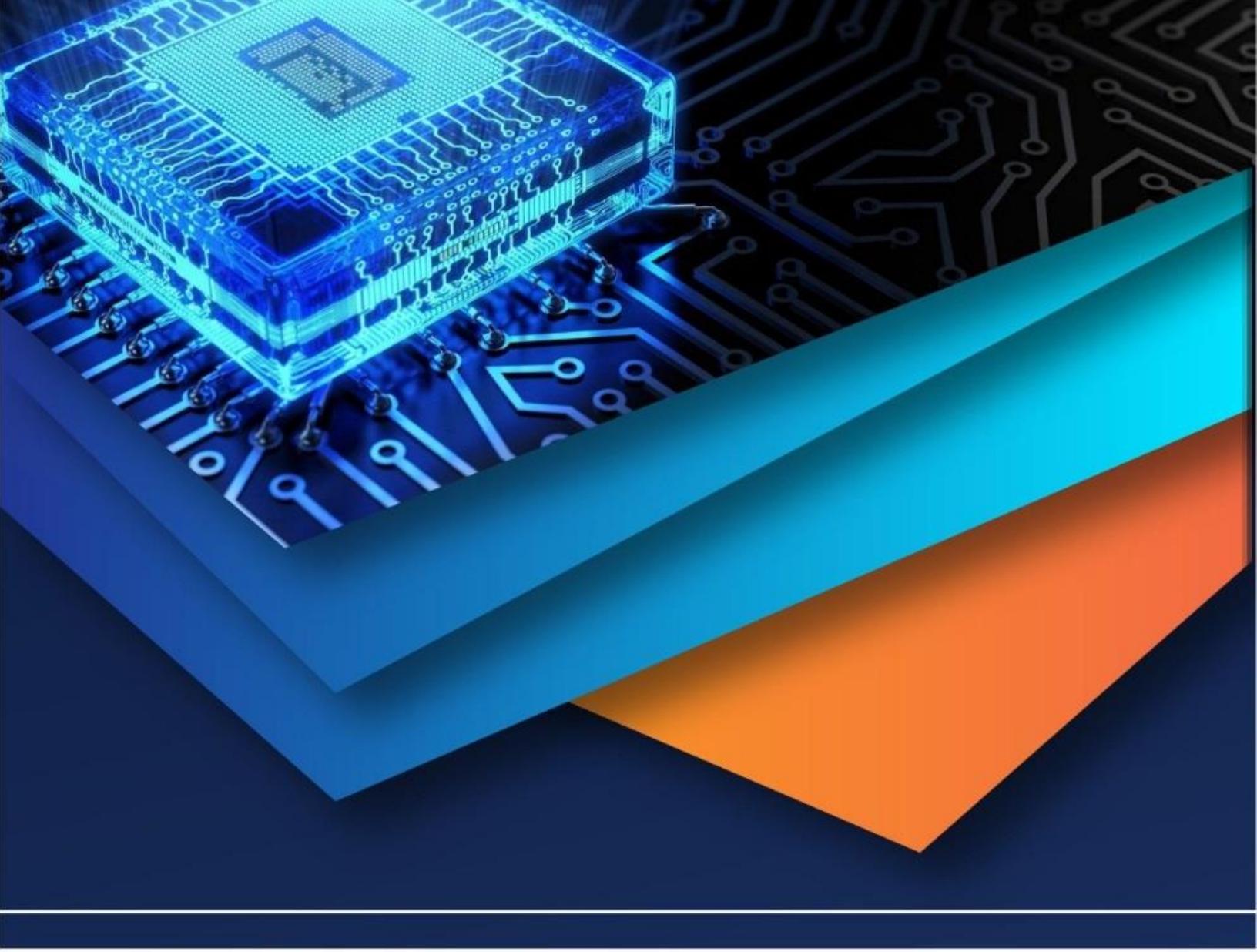
In the above figure 2.4 figure where we have a web portal of Choose file when we press the button preview where we can search the image in or trained dataset of skin diseases which you need to find the randiomics sets and upload in that browser later it will use cnn procedure for locating which disease it is and it will give the name of which disease it is with the exactness score

V. CONCLUSION

In conclusion, the application of CNN for dermatological malignancy detection represents a significant advancement in the field of skin and medical imaging. The use of learning techniques, particularly CNNs, Has demonstrated extraordinary efficacy in precise identification and classifying dermatological lesions, including potentially cancerous ones. This technology brings several key advantages

REFERENCES

[1] S. Prasher, L. Nelson and S. Gomathi, "Resnet 50 Based Classification Model for Skin Cancer Detection Using Dermatoscopic Images," 2023 3rd Asian Conference on Innovation in Technology (ASIANCON), Ravet IN, India, 2023,pp. 1-5, doi: 10.1109/ASIANCON58793.2023.10270158.
[2] Nourabuared, “Skin Cancer Based on VGG19 and Transfer Learning”, In 3rd International Conference on Signal Processing and Information Security IEEE Conference, pp 1-4, DOI
[3] N. C. Lynn and Z. M. Kyu, , Taipei, Taiwan, 2017, pp. 117-122, DOI: 10.1109/PDCAT.2017.00028.
[4] Harikrishna, “Skin Cancer Classification using Transfer Learning IEEE International Conference on Advent Trends in Multidisciplinary Research and Innovation (ICATMRI-2020)
[5] N. C. Lynn and Z. M. Kyu, "Segmentation and Classification of Skin Cancer Melanoma from Skin Lesion Images," 2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies, Taipei, Taiwan, 2017, pp. 117-122, DOI: 10.1109/PDCAT.2017.00028.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59154, entitled

Dermatological Malignancy Detection Using CNN

by

Ms. Komal Parashar

after review is found suitable and has been published in Volume 12, Issue III, March 2024

in

International Journal for Research in Applied Science & Engineering Technology

(International Peer Reviewed and Refereed Journal)

Good luck for your future endeavors

Editor in Chief, iJRASET

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59154, entitled

Dermatological Malignancy Detection Using CNN

by

M. V. Devendranathreddy

after review is found suitable and has been published in Volume 12, Issue III, March 2024

in

International Journal for Research in Applied Science & Engineering Technology

(International Peer Reviewed and Refereed Journal)

Good luck for your future endeavors

Editor in Chief, iJRASET

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59154, entitled

Dermatological Malignancy Detection Using CNN

by

M. Deepak Reddy

after review is found suitable and has been published in Volume 12, Issue III, March 2024

in

International Journal for Research in Applied Science & Engineering Technology

(International Peer Reviewed and Refereed Journal)

Good luck for your future endeavors

Editor in Chief, iJRASET

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL

SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59154, entitled

Dermatological Malignancy Detection Using CNN

by

S. Jashwitha

after review is found suitable and has been published in
Volume 12, Issue III, March 2024
in

International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET