

Epidermal Neoplasm Detection Using Convolutional Neural Networks

A

Project Report Submitted

in partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

By

Elike Stella

20761A05E9

Yandrapati Harshitha Sampath

20761A05J2

Darapureddy Dhanush Kumar

20761A05E8

Under the esteemed guidance of

Dr. Ch. Venkata Narayana

Professor



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING
(AUTONOMOUS)**

Accredited by NAAC with 'A' Grade & NBA (Under Tier - I)

An ISO 21001:2018, 14001:2015, 50001:2018 Certified Institution

Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada

L. B. Reddy Nagar, Mylavaram, NTR Dt., A.P.-521 230

2020 - 2024

**LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING
(AUTONOMOUS)**

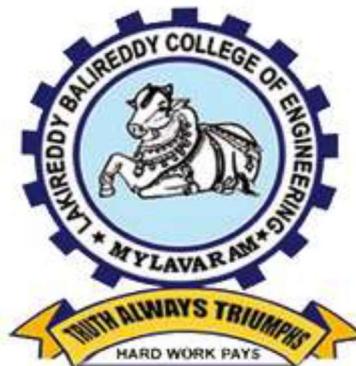
Accredited by NAAC with ‘A’ Grade & NBA (Under Tier – I)

An ISO 21001:2018, 14001:2015, 50001:2018 Certified Institution

Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada

L. B. Reddy Nagar, Mylavaram, NTR Dt., A.P.-521 230

**Department Of
COMPUTER SCIENCE & ENGINEERING**



CERTIFICATE

This is to certify that the project entitled “**Epidermal Neoplasm Detection Using Convolutional Neural Networks**” is being submitted by

Elike Stella

20761A05E9

Yandrapati Harshitha Sampath

20761A05J2

Darapureddy Dhanush Kumar

20761A05E8

in partial fulfillment of the requirements for the award of degree of **B.Tech in Computer Science & Engineering** from **Jawaharlal Nehru Technological University Kakinada** is a record of bonafide work carried out by them at **Lakireddy Bali Reddy College of Engineering**.

The results embodied in this Project report have not been submitted to any other University or Institute for the award of any degree or diploma.

PROJECT GUIDE

Dr. Ch. Venkata Narayana

HEAD OF THE DEPARTMENT

Dr. D. Veeraiah

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

we take great pleasure in expressing our deep sense of gratitude to our project guide **Dr. Ch. Venkata Narayana**, Professor, for his valuable guidance during the course of our project work.

we take great pleasure in expressing our deep sense of gratitude to our project coordinator **Dr. M. Sitha Ram**, Associate Professor, for his valuable guidance during the course of our project.

we would like to thank **Dr. D. Veeraiah**, Professor & Head of the Department of Computer Science & Engineering for his encouragement.

we would like to express our heartfelt thanks to **Dr. K. Appa Rao**, Principal, Lakireddy Bali Reddy College of Engineering for providing all the facilities for our project.

Our utmost thanks to all the faculty members and Non-Teaching Staff of the Department of Computer Science & Engineering for their support throughout our project work.

Our Family Members and Friends receive our deepest gratitude and love for their support throughout our academic year.

Elike Stella

20761A05E9

Yandrapati Harshitha Sampath

20761A05J2

Darapureddy Dhanush Kumar

20761A05E8

DECLARATION

We are here by declaring that the project entitled "**Epidermal Neoplasm Detection Using Convolutional Neural Networks**" work done by us. We certify that the work contained in the report is original and has been done by us under the guidance of our supervisor. The work has not been submitted to any other institute in preparing for any degree or diploma. We have followed the guidelines provided by the institute in preparing the report. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the institute. Whenever we have used materials (data, theoretical analysis, figures and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references.

Signature(s) of the Students(s)

Elike Stella (20761A05E9)

Yandrapati Harshitha Sampath (20761A05J2)

Darapureddy Dhanush Kumar (20761A05E8)

ABSTRACT

Skin cancer is the dangerous medical condition which is observed by the abnormal growth of skin cells. It is majorly caused due to ultraviolet radiations. Detecting it in the early stage may reduce its severity. This project delves into a comprehensive analysis of skin cancer prediction, employing three distinct deep learning models: ResNet50, Inception v3 and VGG16. Utilizing a diverse dataset comprising dermoscopic images, the study rigorously evaluates and compares the performance of these models across various metrics such as accuracy, loss, precision and recall. Also we have created an user interface to test the skin cancer images. After a great deal of testing, ResNet50 is the best option for detecting skin-cancer since it can identify a wide range of skin tumors with greater accuracy. The project incorporates advanced data augmentation techniques to bolster model resilience and generalization, ensuring robust performance in real-world scenarios. This highlights its potential significance in clinical applications, highlighting how deep machine learning is essential to improving medicinal image processing and improving the results for patients.

LIST OF CONTENTS

CONTENTS	PAGE NO
1. INTRODUCTION	
1.1 Overview of the Project	1
1.2 Novelty of the project	2
1.3 Feasibility Study	3
1.4 Scope	5
2. LITERATURE SURVEY	
2.1 Existing System & Drawbacks	6
2.2 Proposed System & Advantages	7
2.3 Deep Learning	8
3. SYSTEM ANALYSIS	
3.1 Overview of System Analysis	12
3.2 Software used in the project	14
3.3 Modules	16
3.4 System Requirements	17
4. SYSTEM DESIGN	
4.1 Overview of System Design	18
4.2 UML Diagrams	19
5. CODING & IMPLEMENTATION	
5.1 Coding	24
5.2 Implementation	35
6. SYSTEM TESTING	
6.1 Overview of Testing	42
6.2 Types of Testing Methods	42
7. RESULTS	44
8. CONCLUSION	50
9. REFERENCES	51

LIST OF FIGURES	PAGE NO
3.1: Project SDLC	12
4.1: Proposed Model	18
4.2.1: Use case Diagram	20
4.2.2: Sequence Diagram	21
4.2.2: Sequence Diagram	23
5.2.1.1: ResNet50 Architecture	36
5.2.1.2: Inception v3 Architecture	37
5.2.1.3: VGG16 Architecture	38
5.2.2.1: Dataset Description	40
5.2.2.2: Dataset Images	40-41
7.1: ResNet50 Accuracy & Loss	44
7.2: Inception v3 Accuracy & Loss	45
7.3: VGG16 Accuracy & Loss	45
7.4: Home Page	46
7.5: Melanocytic Nevi Prediction	46
7.6: Pyogenic granulomas Prediction	47
7.7: Benign Prediction	47
7.8: Dermatofibroma Prediction	48
7.9: Melanoma Prediction	48
7.10: Basal Cell Carcinoma Prediction	49
7.11: Actinic Keratosis Prediction	49

1. INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The surging global incidence of skin cancer, a pervasive and potentially fatal affliction, has intensified the demand for innovative diagnostic strategies. A transformative shift, fueled by a mix of neural networks, deep artificial intelligence, and computational imaging, now stands as a revolutionary force, pledging heightened accuracy and advanced prognostic capabilities [1]. Groundbreaking initiatives have prepared the way for the identification of skin cancer that is melanoma, leveraging a synthesis of image processing and machine learning methodologies [1]. The study of lesions of the skin categorization using ensembles of deep convolutional neural networks opened up fresh possibilities for progress and marked a significant turning point in the field of artificial intelligence [2].

An important next step in the identification of skin tumours is to compare algorithms based on deep- learning and machine learning [3]. An innovative approach integrating individual advantage and group decision ensembles to amplify classification accuracy marked a significant stride towards preventive diagnostic measures [4]. Innovative project employing efficiency networks to classify skin tumours into many classes was a major advancement in the field of preventative diagnostics [5]. Methods for diagnosis gained an interactive component with the development of an attack- based process that combined picture and information [6]. The significance of feature variety in automated skin tumour diagnosis was highlighted by project conducted on skin cancer identification using integrated making choices by deeper drivers [7]. Integration of convolutional neural networks with texture features highlighted the importance of feature diversity in automating skin cancer detection [8].

Innovative methodologies emerged with the introduction of a novel regularizer in Skin tumor identification using convolution neural network models [12]. A revisit to ensemble learning underscored its persistent efficacy in skin lesion classification, contributing to the discourse on ensemble learning in dermatological diagnostics [13]. Beyond computational methodologies, a study on secondary melanoma using ^{1}H NMR metabolomics delved into the

metabolic aspects of melanoma, adding a novel dimension to the understanding of skin cancer. An important turning point in the merging of dermatologists and technology was the investigation of applications for computers in the detection and prediction of cancer of the skin. Many projects have been made in this skin cancer detection. Mostly they have predicted melanoma since it is very common type of skin cancer. They have used many machine learning and deeplearning models like SVM, Mobilenet, inception, sequential models, VGG19, ResNet and so on. Other types of skin cancers were also predicted such as benign, melanoma, basal cell carcinoma etc.

In our project we have taken tree models such as ResNet50, Inception v3 and VGG16. These 3 models are compared to know which model is giving the best accuracy for skin cancer detection. We trainedthese models on HAM10000 dataset which contains 10015 dermatoscopic images. The dataset is preprocessed in a wide range before giving it to the model. Each model is trained with the dataset one by one and we have taken the observations from the accuracy and loss given by the models. From the observations drawn we have concluded that ResNet50 gives the highest accuracy compared to other. We have given the results of each model with a confusion matrices and accuracy, loss plots. We have created a user interface which takes skin image as input and predicts the class.

1.2 NOVELTY OF THE PROJECT

The project's novelty is underscored by a distinctive amalgamation of state-of-the-art methodologies, presenting a cutting-edge method for the identification and categorization of skin tumours. At its core, the project distinguishes itself through the integration of advanced image processing, feature extraction, and machine learning techniques, collectively surpassing existing methods in accuracy and reliability.

One pivotal aspect of the project's uniqueness lies in its meticulous image preprocessing techniques. Leveraging sophisticated procedures such as noise reduction, resizing the image, the project ensures an unparalleled enhancement of dermatoscopic image's quality. This meticulous preprocessing sets the stage for subsequent stages, substantially elevating the precision of feature extraction.

A key innovation is manifested in the project's approach to feature extraction. While embracing established features for characterizing skin lesions, the project introduces novel features meticulously designed to capture subtle intricacies within lesions. This refined feature extraction methodology aims to discern nuanced patterns indicative of different skin cancer types, markedly improving the model's classification accuracy.

The crux of the project's distinctiveness lies in the application of ensemble learning, particularly the incorporation of diverse convolutional neural networks (CNNs). Uniquely, each CNN within the ensemble undergoes pretraining on distinct datasets, harnessing transfer learning to capitalize on comprehensive knowledge from broader image datasets.

An innovative addition to the methodology is the introduction of an attention-based mechanism, strategically combining image data with pertinent metadata. This attention mechanism introduces a nuanced layer to the classification process, considering not only visual features but also supplementary information linked to skin lesions. This holistic approach augments diagnostic accuracy by providing a more comprehensive analysis of skin cancer cases.

Moreover, the project pioneers a combined decision-making strategy rooted in deep learning models. By aggregating predictions from multiple models, the project maximizes the strengths of each, resulting in a more nuanced and reliable classification outcome. This distinctive fusion of predictions promises a sophisticated approach to decision support in skin cancer diagnosis.

In essence, the project's novelty lies in its unique synthesis of advanced image processing, novel feature extraction, ensemble learning, attention-based fusion, and combined decision-making. This thorough and diverse method represents a major advancement in the field of identifying skin tumours by providing an extremely efficient and unique instrument for early identification and categorization.

1.3 FEASIBILITY STUDY

In this we study various feasibilities where existing and software equipment were sufficient for completing the project. The economic Feasibility determines whether doing the

project is economically beneficial. The outcome of first phase was that the request and various studies are approved and it was decided that the project taken up will serve the end user. On developing and implementation this software saves a lot of amount and sharing of valuable time.

- Economical Feasibility
- Technical Feasibility
- Social Feasibility

1.3.1 Economical Feasibility

The study is carried out to check the economic impact that the system will have on the organization. The amount of fund pour in project and development of system is limited. The expenditure is justified.

1.3.2 Technical Feasibility

This is carried out to check the technical feasibility that is, the technical requirements of the system. Any system developed must not have a high demand on available technical resources. The developed System must have modest requirements and are required for implementing this system. Our project has modest technical requirements.

1.3.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not be threatened by the system. His/her level of confidence must be increased so that he/she is able to make some constructive criticism which is welcomed.

1.4 SCOPE

The primary use of a skin cancer detection project is to assist healthcare professionals in accurately diagnosing skin lesions, particularly in identifying potential cases of melanoma and other types of malignant skin cancer. By leveraging deep learning algorithms trained on large datasets of annotated skin lesion images, these projects aim to provide automated or computer-aided diagnosis tools.

Early detection of skin cancer can significantly improve patient outcomes by enabling timely intervention and treatment. Automated detection systems can assist in identifying suspicious lesions at an early stage, even before they become clinically apparent. Deep learning models trained on extensive datasets can complement the diagnostic capabilities of healthcare professionals by providing additional insights and reducing diagnostic errors. Skin cancer detection systems can be deployed in various healthcare settings, including clinics, hospitals, and even remote areas with limited access to dermatologists. This can improve access to timely and accurate diagnosis, particularly in underserved communities. Automated detection systems can analyze images of skin lesions much faster than human experts, potentially reducing the time required for diagnosis and treatment planning.

Rather than replacing dermatologists, these systems can serve as decision support tools, aiding dermatologists in their diagnostic process and potentially reducing their workload. Skin cancer detection projects can also serve as educational tools for healthcare professionals, providing access to annotated datasets and training materials for improving diagnostic skills.

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM AND DRAWBACKS

Skin tumors, especially a form of is a major worldwide medical problem that requires sophisticated detection techniques. Significant progress in the identification of skin tumors has been sparked by current developments in computational biology and artificial intelligence. The groundwork for further project was laid by Vijayalakshmi [1], who led investigations into identifying melanoma using computational imaging and artificial intelligence. Dermal lesion categorization underwent an evolution with the introduction of deep learning, namely convolutional artificial neural networks (CNNs).

Harangi [2] introduced ensemble methods with deep CNNs for skin lesion classification, enhancing diagnostic accuracy. Gong et al. [4] explored the classification of dermoscopy images using CNNs, emphasizing the effectiveness of ensemble strategies.

EfficientNets gained prominence in multiclass skin cancer classification, as demonstrated by Ali et al. [5]. Pacheco and Krohling [6] proposed, combined for improved classification. Imran et al. [7] highlighted the significance of combined decision-making by deep learners in skin cancer detection. Alizadeh and Mahloojifar [8] integrated CNNs with texture features for automatic skin cancer detection, showcasing the synergy of different methodologies. Han et al. [9] focused on keratinocytic skin cancer detection, leveraging region-based CNNs for enhanced accuracy. Tschandl et al. [10] achieved expert-level diagnosis using combined CNNs, underscoring the potential for leveraging deep learning in clinical settings. A systematic review by Haggenmüller et al. [11] emphasized the growing body of literature on skin cancer classification via CNNs. Albahar [12] introduced a novel regularizer in CNNs for improved skin lesion classification. In some papers they investigate how they can create an ensemble of deep convolutional neural networks to improve further their individual accuracies in the task of classifying dermoscopy images into the three classes melanoma, nevus, and seborrheic keratosis when they have no opportunity to train them on adequate number of annotated images. To achieve high classification accuracy, they fuse the outputs of the classification layers of four different deep neural network architectures [13].

In addition to algorithmic advancements, several studies explored the integration of clinical knowledge into the diagnostic process. The ABCDE rule, proposed by Nachbar et al. [14], provided a valuable framework for dermatoscopic analysis. Faziloglu et al. [15] contributed to melanoma discrimination using color histogram analysis, showcasing the interdisciplinary nature of skin cancer project.

The latest findings, as revealed by The studies by Esteva et al. [16], showed that deep brain networks could classify skin cancer at the physician level. underscoring the transformative potential of artificial intelligence in clinical practice. Shanthi et al. [17] furthered this progress by proposing an automatic diagnosis system based on AI. Veronika Cheplygina [18] and team used different models like CNN and SVM to predict the skin cancer with an accuracy of 85%. Many other machine learning models are used with skin lesion images to predict the class of skin cancer[19]. The studies corroborates that very deep CNNs with effective training mechanisms can be employed to solve complicated medical image analysis tasks, even with limited training data [20].

In summary, the integration of deep neural networks, artificial intelligence, and computer vision has ushered in an entirely novel phase in skin tumour diagnosis. The cited works collectively contribute to a robust foundation for future project, emphasizing the interdisciplinary collaboration between medical professionals and technologists.

The existing systems mainly classified melanoma, basal cell carcinoma, benign, squamous cell carcinoma types of cancers. They have used different algorithms like basic CNN, Sequential models EfficientNet, MobileNet, VGG16, Inception v3, ResNet etc .

Major Drawbacks :

- Accuracy
- Lack of exploring different models
- Permitted to only few classes

2.2 PROPOSED SYSTEM AND ADVANTAGES

In our project we have taken three models such as ResNet50, Inception v3 and VGG16.These3 models are compared to know which model is giving the best accuracy for skin

cancer detection. We trained these models on HAM10000 dataset which contains 10015 dermatoscopic images. The dataset is preprocessed in a wide range before giving it to the model. Each model is trained with the dataset one by one and we have taken the observations from the accuracy and loss given by the models. From the observations drawn we have concluded that ResNet50 gives the highest accuracy compared to other 2. We have given the results of each model with a confusion matrices and accuracy, loss plots. We have created an user interface which takes skin image as input and predicts the class.

Three Models Used:

- ResNet50, Inception v3, VGG16 models are compared.
- 7 classes of skin cancer are predicted.
- User interface is created.
- Accuracy improved to 98.9% for ResNet50 model.

2.3 Deep Learning

Deep learning is a subset of machine learning that involves training artificial neural networks to recognize patterns and make predictions based on data. It is inspired by the structure and function of the human brain, which consists of interconnected neurons that process information in parallel. In deep learning, large amounts of data are fed into a neural network, which consists of multiple layers of interconnected nodes. Each layer extracts features from the data and passes them on to the next layer, until final output is produced. The network learns by adjusting the weights and biases of the connections between nodes in response to feedback on its predictions.

One of the key advantages of deep learning is its ability to automatically learn hierarchical representations of data. This means that the network can learn to recognize complex patterns in the data, even if they are not explicitly defined by the programmer. For example, a deep learning algorithm can learn to recognize a face by automatically identifying the edges and contours of facial features at lower layers and combining them into more complex features such as eyes, nose, and mouth at higher layers. Deep learning has found a wide range of applications in fields such as computer vision, natural language processing,

speech recognition, and robotics.

In computer vision, deep learning algorithms have achieved state-of-the-art performance in tasks such as image classification, object detection, and segmentation. In natural language processing, deep learning has been used to build language models that can generate coherent text, translate between languages, and perform question-answering tasks. In speech recognition, deep learning has been used to improve the accuracy of automatic speech recognition systems. In robotics, deep learning has been used to enable robots to learn from their environments and perform tasks such as object manipulation and navigation. Despite its many successes, deep learning also has some limitations.

One of the main challenges is the need for large amounts of labeled data to train the networks. This can be a bottleneck in many applications, particularly in domains where labeled data is scarce or expensive to obtain. Another challenge is the interpretability of deep learning models, which can be difficult to understand and debug due to their high complexity. Nevertheless, deep learning continues to be a rapidly growing field of research, with many exciting developments on the horizon. One of the key challenges in deep learning is the risk of overfitting, which occurs when a model becomes too complex and starts to memorize the training data rather than learning generalizable patterns.

2.3.1 CNN

A Convolutional Neural Network (CNN) is a type of deep neural network commonly used in image and video processing applications. CNNs are particularly well-suited to these tasks because they can automatically learn hierarchical representations of the data, starting with simple features such as edges and textures, and building up to more complex representations such as objects and scenes. A CNN consists of several layers, each of which performs a specific type of computation on the input data. The first layer is typically a convolutional layer, which applies a set of filters to the input image to extract features such as edges, corners, and textures. The output of the convolutional layer is then passed through a pooling layer, which reduces the

spatial dimensions of the feature map by performing operations such as max pooling or average pooling.

This helps to reduce the number of parameters in the model and makes it less susceptible to overfitting. After several convolutional and pooling layers, the output is typically flattened and passed through one or more fully connected layers, which perform a classification or regression task based on the extracted features. The final layer of the network is often a softmax layer, which produces a probability distribution over the possible classes or outputs. One of the key advantages of CNNs is their ability to learn spatially invariant features, meaning that they can recognize objects regardless of their position, rotation, or scale in the input image. This is achieved by sharing the weights of the filters across the spatial dimensions of the input, which allows the network to learn features that are applicable to different parts of the image. CNNs have been used for a wide range of image and video processing tasks, such as image classification, object detection, segmentation, and style transfer. They have achieved state-of-the-art performance on benchmark datasets such as ImageNet and have been deployed in real-world applications such as self-driving cars, medical imaging, and surveillance systems.

Despite their many successes, CNNs also have some limitations. One of the main challenges is the need for large amounts of labeled data to train the network, which can be a bottleneck in many applications. Another challenge is the interpretability of the learned features, which can be difficult to understand and visualize due to the high dimensionality of the feature maps. Nevertheless, CNNs continue to be a widely used and important tool in the field of deep learning. The layers of the CNN are:

- **Convolutional Layer:** A convolutional layer is one of the fundamental building blocks of a Convolutional Neural Network (CNN). It is designed to extract features from input data, such as images, by applying a set of learnable filters to the input data. With this layer we can extract the important data present in the input image.
- **Dropout Layer:** Dropout is a regularization technique commonly used in deep learning

models, including Convolutional Neural Networks (CNNs), to prevent overfitting and improve generalization performance. The dropout layer is a type of layer that implements this technique.

- Pooling Layer: A pooling layer is a type of layer commonly used in Convolutional Neural Networks (CNNs) to reduce the spatial dimensions of the feature map generated by the convolutional layer. Pooling layers help to reduce the number of parameters in the network, making it less susceptible to overfitting, while also providing some degree of translation invariance. There are three types in pooling they are: Max Pooling, Average Pooling and Min Pooling.
- Batch Normalization Layer: The batch normalization layer operates on a mini batch of input data, typically of size 32, 64, or 128. During training, it normalizes the input activations of the previous layer by subtracting the batch mean and dividing by the batch standard deviation. The resulting activations are then scaled and shifted using learnable parameters (gamma and beta) to provide the network with the flexibility to adjust the normalized activations as needed.
- Flatten Layer: The flatten layer simply reshapes the output tensor of the previous layer into a single dimension. For example, if the output tensor has dimensions $32 \times 32 \times 64$, the flatten layer would reshape it into a vector of length $64 \times 32 \times 32 = 65,536$. This flattened vector can then be fed into a fully connected layer for classification or regression.
- Fully Connected Layer: A fully connected layer, also known as a dense layer, is a type of layer commonly used in Convolutional Neural Networks (CNNs) that helps to map the features learned by the convolution layers to the output classes.

3. SYSTEM ANALYSIS

3.1 OVERVIEW SYSTEM ANALYSIS

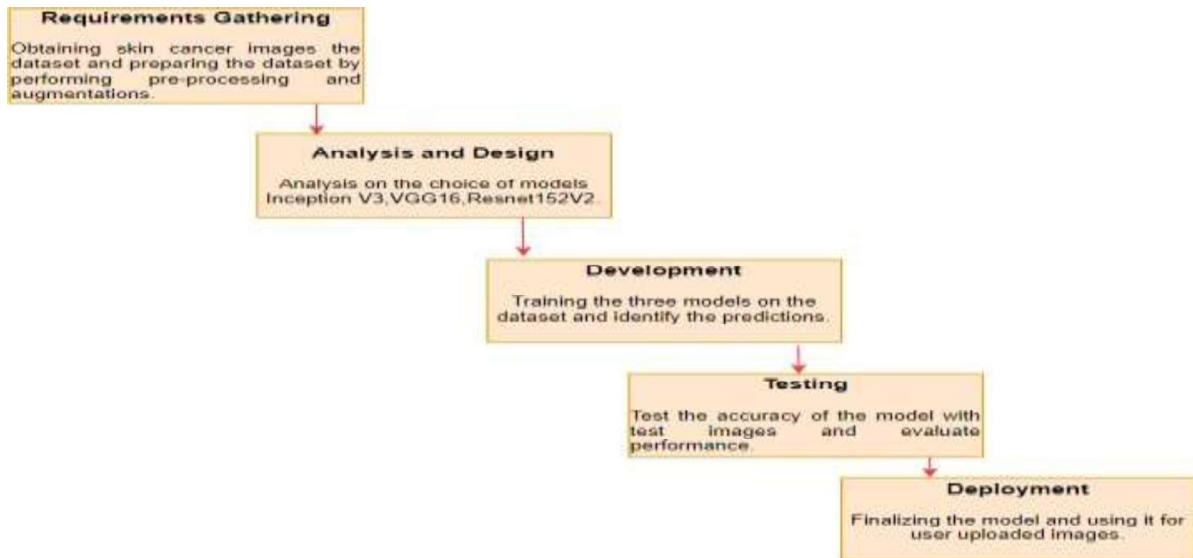


Fig.3.1: Project SDLC

- Project Requisites Accumulating and Analysis
- Application System Design
- Practical Implementation
- Manual Testing of my Application
- Application Deployment of System
- Maintenance of the Project

3.1.1 Requisites Accumulating and Analysis

It's the first and foremost stage of the any project as our is a an academic leave for requisites amassing we followed of IEEE Journals and Amassed so many IEEE Related papers and final culled a Paper designated by setting and substance importance input and for analysis stage we took referees from the paper and did literature survey of some papers and amassed all the Requisites of the project in this stage.

3.1.2 System Design

During the Design Phase, the system is designed to satisfy the requirements identified in the previous phases. The requirements identified in the Requirements Analysis Phase are transformed into a System Design Document that accurately describes the design of the system and that can be used as an input to system development in the next phase.

3.1.3 Implementation

The Implementation is Phase where we endeavor to give the practical output of the work done in designing stage and most of Coding in Business logic lay comes into action in this stage its main and crucial part of the project.

3.1.4 Testing

Unit Testing:

It is done by the developer itself in every stage of the project and fine-tuning the bug and module predicated additionally done by the developer only here we are going to solve all the runtime errors.

Manual Testing:

As our Project is academic leave we can do any automatic testing so we follow manual testing by endeavor and error methods.

3.1.5 Deployment of System

Once the project is totally tested, we will come to deployment of client system in genuinely worldas its academic leave we did deployment in our college lab only with all need Software's with having Windows OS.

3.1.6 Maintenance

The Maintenance of our Project is one time process only.

3.2 SOFTWARE USED IN THE PROJECT

What things you need to install the software :

1. Python 3.8.18

Python 3.8.18 is a stable and reliable version of Python, suitable for various software development projects. It's recommended to keep your Python installations up-to-date with the latest maintenance releases to benefit from bug fixes, performance improvements, and security patches.

2. Jupyter Notebook

It is an interactive computing environments for developing and experimenting with code and data visualization.

3. Libraries and frameworks

Pickle: A module used for serializing and deserializing Python objects. It's often used for saving trained machine learning models to disk.

PIL (Python Imaging Library): A library for opening, manipulating, and saving many different image file formats. In this context, it's likely used for working with image data.

Matplotlib: A plotting library for creating visualizations and graphs in Python. It's commonly used for visualizing data and results.

Pandas: A powerful library for data manipulation and analysis. It's commonly used for

handling structured data, such as CSV files or Excel spreadsheets.

NumPy: A fundamental package for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

Seaborn: A data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive statistical graphics.

Keras: A high-level neural networks API, running on top of TensorFlow or other deep learning frameworks. It's used for building, training, and deploying deep learning models with ease.

TensorFlow (via TensorFlow.Keras): An open-source machine learning framework developed by Google. In this context, TensorFlow is used as the backend for Keras, providing support for training and running deep learning models efficiently on CPUs and GPUs.

Scikit-learn (sklearn): A machine learning library for classical machine learning algorithms. It provides simple and efficient tools for data mining and data analysis, including classification, regression, clustering, and more.

4. Pretrained Models

ResNet50: A pre-trained convolutional neural network model commonly used for image classification tasks. It's part of the TensorFlow/Keras applications module and can be easily loaded for transfer learning or fine-tuning on custom datasets.

Inception-V3: is a powerful CNN architecture that combines innovative design principles with efficient computational techniques, making it well-suited for a wide range of image processing tasks.

VGG16: VGG16 is a classic CNN architecture known for its simple yet effective design and strong performance on a wide range of computer vision tasks.

3.3 MODULE

ResNet50.py :

This is the file in which the ResNet50 model is built and implemented. ResNet50, a pre-trained convolutional neural network (CNN) architecture, is loaded from Keras applications. The file consists of a workflow for training a deep learning model for skin lesion classification, including data loading, preprocessing, model construction, training, evaluation, and individual image prediction.

Inception v3.py :

This is the file in which the Inception v3 model is built and implemented. Inception v3, a pre-trained convolutional neural network (CNN) architecture, is loaded from Keras applications. The file consists of a workflow for training a deep learning model for skin lesion classification, including data loading, preprocessing, model construction, training, evaluation, and individual image prediction.

VGG16.py :

This is the file in which the VGG16 model is built and implemented. VGG16, a pre-trained convolutional neural network (CNN) architecture, is loaded from Keras applications. The file consists of a workflow for training a deep learning model for skin lesion classification, including data loading, preprocessing, model construction, training, evaluation, and individual image prediction.

App.py :

This is the main file for the project which is used to connect the ResNet50 model with the home.html file. Flask application provides a simple web interface where users can upload images of skin lesions, and the application uses a pre-trained deep learning model to predict the type of skin cancer based on the uploaded image. It then displays the prediction results along

with relevant information about the predicted skin cancer type.

Home.html :

This is the html file which provides interface to take input from the user.

Results.html :

This is the html file which will provide the result of the predicted class.

3.4 SYSTEM REQUIREMENTS

3.4.1 Software : Python

3.4.2 Version : 3.8

3.4.3 Operating System : windows

4. SYSTEM DESIGN

4.1 OVERVIEW OF SYSTEM DESIGN

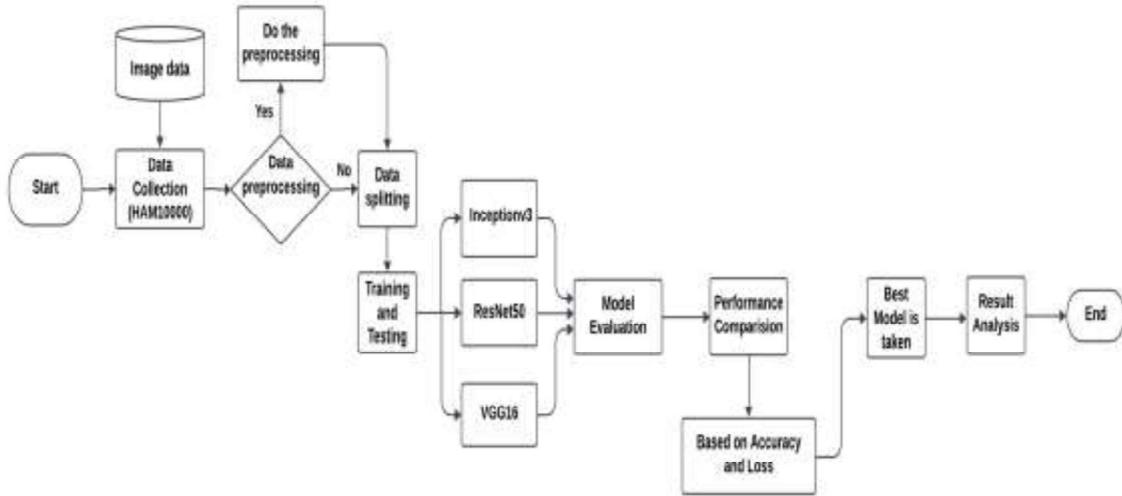


Fig.4.1: Proposed Model

Algorithm

Input:

X: Input image data

W: Convolution filter/kernel
b: Biass: Pooling stride

wi: Weights for the dense layer
bi: Bias for the dense layer

Step1: Convolution Operation: $Y[i,j] = \sum m \sum n (X[I+m, j+n]) \times (W[m, n]) + b$

Perform convolution operation on the input image X using the kernel W and bias b to obtain the feature maps Y.

Step 2: Activation of ReLU: $f(x) = \max(0, x)$

Apply the ReLU activation function element-wise to the feature maps Y to introduce non-linearity.

Step 3: Pooling Operation (Max Pooling): $[i,j] = \max_{m,n} X[i \times s + m, j \times s + n]$

Perform max pooling on the feature maps Y with a pooling window size of (m, n) and a stride of s to obtain the pooled feature maps.

Step 4: Flatten the pooled feature maps to prepare for input into the dense layer.

Step 5: Dense Layer Output: $Y=f(\sum iwi \times xi + b)$

Compute the output of the dense layer by taking the dot product of the flattened feature maps with the weights w_i , adding the bias b_i , and applying the ReLU activation function.

Step 6:

The output of the dense layer represents the probability of skin cancer presence.

4.2 UML DIAGRAMS

In the field of software engineering, Unified Modeling Language (UML) is the preferred model for modeling. UML is a standard purpose-oriented modeling language which includes a clear text used to produce a vague model of program, called the UML model. The model contains a "Semantic backplane" - texts such as written use cases that run model objects and drawings.

Importance of UML in Modeling:

Model language is a language with its own words and pronouns that focus on the psychological and physical expression of the system. UML modeling language is thus a common language for software applications. UML is not a graphical programming language, but the models it generates can be directly linked to various programming languages. This in turn means that it is possible to create a map from a model in UML to Java, C ++ or Visual Basic, or even related data tables or a persistent database-focused store. This map allows advanced engineering: coding from UML model to programming language. Going back is also possible to re-create a model from usage back to UML. The UML is a programming language which is used for development of object-based software. To arrange a program code effectively, programmers often be creating "objects" which are structured within systems. The UML, which has been modeled on the Object Management Group (OMG), is designed for this purpose.

A conceptual model of UML:

The three major elements of UML are

- The UML's basic building blocks
- The rules that dictate how those building blocks may be put together
- Some common mechanism that apply throughout the UML

4.2.1 Use case Diagram

The use case diagram shows the set of various kinds of use cases with the characters and the relationships. We will use a case study diagram to illustrate the standalone visual of the system. Use case diagram is very crucial in planning and matching the behavioral behavior of a program.

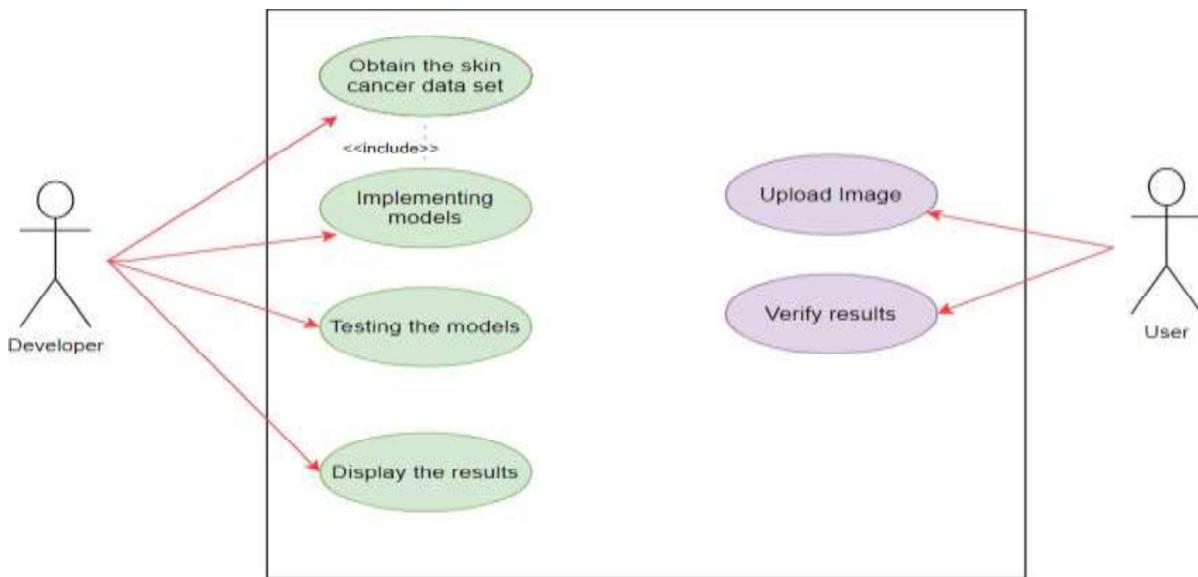


Fig.4.2.1: Use case Diagram

4.2.2 Sequence Diagram

A sequence diagram is a type of interaction diagram in UML (Unified Modeling Language) that

Epidermal Neoplasm Detection Using CNN

shows how objects interact in a particular sequence within a system. It illustrates the flow of messages, actions, and events between different objects or components over time.

Objects: Represent entities or instances involved in the interaction. These could be classes, components, or instances of classes.

Lifelines: Vertical dashed lines that represent the lifespan of each object participating in the interaction. They extend from the top (creation) to the bottom (destruction) of the diagram.

Messages: Horizontal arrows that depict communication between objects. Messages can be synchronous

Activation Bars: Rectangular boxes along a lifeline that represent the period during which an object is performing an operation or processing a message. They indicate the time an object is actively executing a method or processing a message.

Return Messages: Messages that represent the response from the receiver to the sender. They indicate the result of a method invocation or processing of a message.

Control Flow: The order in which messages are exchanged between objects, usually depicted by the sequence of arrows and their arrangement on the diagram.

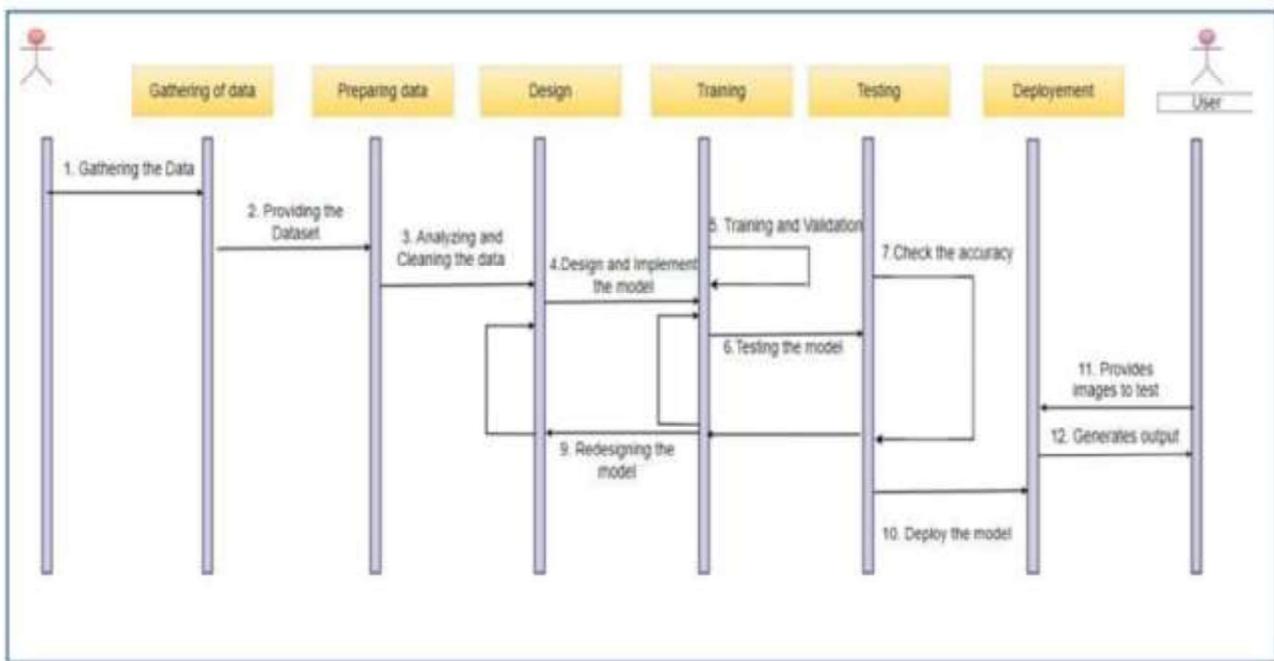


Fig.4.2.2: Sequence Diagram

4.2.3 Activity Diagram

The aim of an activity diagram is to give an idea of the flow and what happens within the application case or between several categories. An activity diagram simply describes the internal tasks performed and the changes caused by the completion of certain tasks. At the incomprehensible level it describes the sequence of tasks. This focuses on the events that occur in one thing as it responds to messages, an activity diagram can be used to perform the whole process. The sketch of the work usually contains:

Activity states and action states

To call the operation of an object that sends a signal for an object or to create or destroy an object, these useful figures are called provinces because they are the regions of the system, representing the action.

The Activity states can then be divided and their activity that is being represented by other activity diagrams can be failed or stopped and take some duration to complete.

Transitions

When an action or state function ceases the flow of control it immediately moves on to the next action or state of action. Explains this flow using changes to show the way from another action or status situation to the next action or work situation. You represent this as an easy-to- direct line.

Object Flow

Items may be involved in the control movement associated with the job drawing. The use of a dependent relationship is called the flow of an object because it indicates the participation of the object in the control movement.

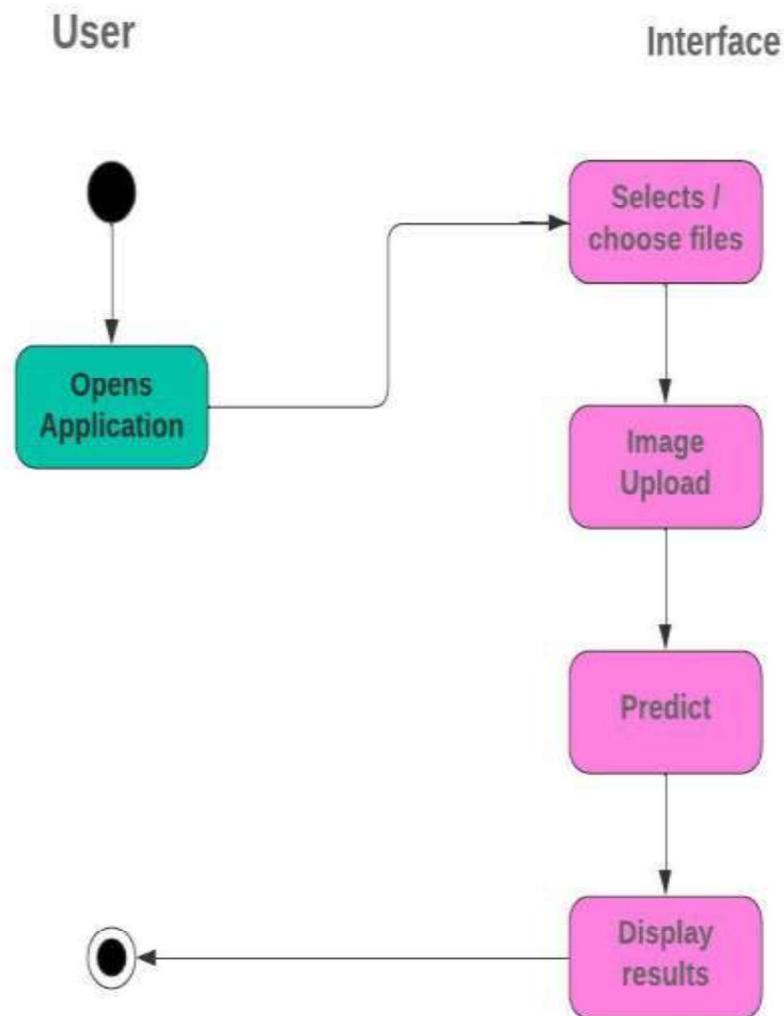


Fig.4.2.3: Activity Diagram

5. CODE & IMPLEMENTATION

5.1 CODING

5.1.1 Importing libraries:

```
import pickle  
  
from PIL import Image import matplotlib.pyplot as pltimport pandas as pd  
  
import numpy as np import seaborn as sns  
  
from keras.models import load_model,Model  
  
from keras.layers import MaxPool2D, Dropout, Dense, GlobalAveragePooling2D  
  
from keras.optimizers import Adam, SGD  
  
from tensorflow.keras.utils import to_categorical  
  
from keras.callbacks import ReduceLROnPlateau, ModelCheckpoint, EarlyStopping  
  
from keras import applications  
  
from keras.preprocessing.image import ImageDataGenerator  
  
import sklearn.metrics as metrics  
  
from sklearn.metrics import confusion_matrix  
  
from sklearn.model_selection import train_test_split  
  
from keras import layers  
  
#import pickle5 as pickle  
  
from sklearn.metrics import classification_report  
  
from tensorflow.keras.applications import ResNet50 # Use tf.keras instead of keras  
  
from tensorflow import keras
```

5.1.2 Loading Dataset:

```
skin_df=pd.read_csv("D:/major project/HAM10000_metadata.csv")  
  
skin_df.head()skin_df.info()
```

5.1.3 Dataset Visualization:

```

sns.countplot(x = "dx", data = skin_df)plt.show()

lesion_dict = {

'nv': 'Melanocytic nevi','mel': 'Melanoma',

'bkl': 'Benign keratosis-like lesions','bcc': 'Basal cell carcinoma',

'akiec': 'Actinic keratoses','vasc': 'Vascular lesions', 'df': 'Dermatofibroma'

}

skin_dff['cell_type'] = skin_dff['dx'].map(lesion_dict.get) skin_df['dx_idx'] = pd.Categorical(skin_dff['cell_type']).codes fig, ax1 = plt.subplots(1, 1, figsize = (10, 5))

sns.countplot(y='dx_idx',data=skin_df, hue="cell_type",ax=ax1)plt.show()

print('Total number of images =',len(skin_df))

print('Melanocytic nevus is unbalanced because it is much more common than other labels.')

show =skin_df['cell_type'].value_counts()

print(show)

skin_df=pickle.load( open ( "D:/major project/skin_dff.pkl" , "rb" ))

plt.imshow(skin_df["image"][0])print(skin_df['image'][0].shape)

Y = to_categorical(skin_df["dx_idx"], num_classes = 7)print("Shape one-hot encoding: ", Y.shape)

X_train, X_test, Y_train, Y_test = train_test_split(skin_df,Y,test_size=0.20)X_train = np.asarray(X_train['image'].tolist())

X_test = np.asarray(X_test['image'].tolist())print(X_train.shape)

print(Y_train.shape)print(X_test.shape) print(Y_test.shape)

X_train,X_val, Y_train,Y_val = train_test_split(X_train,Y_train,test_size=0.10,

```

```
random_state=42)X_train = X_train.reshape(X_train.shape[0],128,128,3)

X_test = X_test.reshape(X_test.shape[0],128,128,3)X_val =

X_val.reshape(X_val.shape[0],128,128,3)

print ("number of final training samples = " + str(X_train.shape[0]))print ("number of test

samples1 = " + str(X_test.shape[0]))

print ("X_train shape:" + str(X_train.shape))print ("Y_train shape:" + str(Y_train.shape))print

("X_test shape: " + str(X_test.shape)) print ("Y_test shape: " + str(Y_test.shape)) print("X_val

shape" + str(X_val.shape))

print("Y_val shape" + str(Y_val.shape))
```

5.1.4 ResNet 50

5.1.4.1 Model Building :

```
img_height,img_width = 128,128

num_classes = 7

base_model = ResNet50(weights='imagenet', include_top=False, input_shape=

(img_height,img_width,3))

for layer in base_model.layers:

    layer.trainable = True

x = base_model.output

x = keras.layers.GlobalAveragePooling2D()(x)

x = keras.layers.Dropout(0.5)(x)

predictions = keras.layers.Dense(num_classes, activation= 'softmax')(x)

model = keras.models.Model(inputs = base_model.input, outputs = predictions)

adam = Adam(learning_rate=0.0001)

model.compile(optimizer= adam, loss='categorical_crossentropy', metrics=['acc'])
```

```
model.summary()

train_datagen = ImageDataGenerator(rotation_range=60, width_shift_range=0.2,
                                    height_shift_range=0.2,
                                    shear_range=0.2, zoom_range=0.2, fill_mode='nearest')

train_datagen.fit(X_train)

val_datagen = ImageDataGenerator()

val_datagen.fit(X_val)

checkpoint = ModelCheckpoint('D:/major project/checkpoint_rsnt.h5', monitor='val_acc',
                            verbose=1, save_best_only=True, mode='max')

reduce_lr = ReduceLROnPlateau(monitor='val_acc', factor=0.5, patience=3,
                             verbose=1, mode='max', min_lr=0.000001)

callbacks_list = [checkpoint, reduce_lr]

epochs=20

batch_size=32

from tensorflow.keras.models import load_model
```

```
loaded_model = load_model("C:/Users/HP/Downloads/check_rsnt .h5")
```

5.1.4.2 Fitting the model:

```
history = loaded_model.fit(train_datagen.flow(X_train,Y_train, batch_size=batch_size),
                           epochs = epochs, validation_data = val_datagen.flow(X_val, Y_val),
                           verbose = 1, steps_per_epoch=(X_train.shape[0] // batch_size),
                           validation_steps=(X_val.shape[0] // batch_size),
                           callbacks=[callbacks_list])

loaded_model.save("D:/major project/ResNet10.h5")
```

5.1.5 Inception v3

5.1.5.1 Model Building :

```
from keras.applications.inception_v3 import InceptionV3, preprocess_input, preprocess_top
pretrained_model = InceptionV3(input_shape=(128, 128, 3), include_top=False, weights="imagenet")

for layer in pretrained_model.layers:
    print(layer.name)

    layer.trainable = False
    print(len(pretrained_model.layers))

last_layer = pretrained_model.get_layer('mixed10')
print('last layer output shape:', last_layer.output_shape)
last_output = last_layer.output

x = layers.GlobalMaxPooling2D()(last_output)
x = layers.Dense(512, activation='relu')(x)
x = layers.Dropout(0.5)(x)
x = layers.Dense(7, activation='softmax')(x)
model = Model(pretrained_model.input, x)

optimizer = Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.999, epsilon=1e-8, amsgrad=True)
model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])
model.summary()

train_datagen = ImageDataGenerator(rotation_range=60, width_shift_range=0.2, height_shift_range=0.2, shear_range=0.2, zoom_range=0.2, fill_mode='nearest')
train_datagen.fit(X_train)
val_datagen = ImageDataGenerator()
val_datagen.fit(X_val)

for layer in pretrained_model.layers:
    layer.trainable = True
    optimizer = Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.999, epsilon=1e-8, amsgrad=True)
    model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['acc'])
    #early_stopping = EarlyStopping(monitor='val_loss', patience=10, mode='min', verbose=1)

checkpoint = ModelCheckpoint('D:/major project/inception/checkpoint_v3.h5', monitor='val_acc', verbose=1, save_best_only=True, mode='max')
```

```
learning_rate_reduction = ReduceLROnPlateau(monitor='val_acc', patience=3, verbose=1,
    factor=0.5,min_lr=0.000001, cooldown=2)
```

```
model.summary()
epochs=20 batch_size=32
```

```
from tensorflow.keras.models import load_model
```

```
loade_model = load_model("C:/Users/HP/Downloads/checkpoint_v3.h5")
```

5.1.5.2 Fitting the Model :

```
history = loade_model.fit(train_datagen.flow(X_train,Y_train, batch_size=batch_size), epochs
    = epochs, validation_data = val_datagen.flow(X_val, Y_val),
verbose = 1, steps_per_epoch=(X_train.shape[0] // batch_size),
validation_steps=(X_val.shape[0] // batch_size),
callbacks=[learning_rate_reduction, checkpoint])
```

```
loade_model.save("D:/major project/inception/Inception v3.h5")
```

5.1.6 VGG16

5.1.6.1 Building the Model:

```
from keras.applications.VGG16 import VGG16, preprocess_input
pre_trained_model = VGG16(input_shape=(128, 128, 3), include_top=False, weights="imagenet")
for layer in pre_trained_model.layers:
    print(layer.name)
    layer.trainable = True
print(len(pre_trained_model.layers))
last_layer = pre_trained_model.get_layer('block5_pool')
print('last layer output shape:', last_layer.output_shape)
last_output = last_layer.output
```

Epidermal Neoplasm Detection Using CNN

```
x = layers.GlobalMaxPooling2D()(last_output)

x = layers.Dense(512, activation='relu')(x)

x = layers.Dropout(0.5)(x)# 0,5'lik bir bırakma oranı ekleyin

x = layers.Dense(7, activation='softmax')(x)

model = Model(pre_trained_model.input, x)

optimizer = Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.999, epsilon=1e-8, amsgrad=True)

model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])

model.summary()

train_datagen = ImageDataGenerator(rotation_range=60, width_shift_range=0.2,
height_shift_range=0.2,shear_range=0.2,zoom_range=0.2, fill_mode='nearest')

train_datagen.fit(X_train)

val_datagen = ImageDataGenerator()

val_datagen.fit(X_val)

for layer in model.layers[:15]:

    layer.trainable = False

for layer in model.layers[15:]:

    layer.trainable = True

optimizer = Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.999, epsilon=1e-8, amsgrad=False)

model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['acc'])

model.summary()

checkpoint = ModelCheckpoint('C:/Users/dhanu/Desktop/Project/Models/VGG16/Epoch20/checkpoint_VGG
16.h5', monitor='val_acc', verbose=1,save_best_only=True, mode='max')

learning_rate_reduction = ReduceLROnPlateau(monitor='val_acc', patience=3, verbose=1,
factor=0.5,min_lr=0.000001, cooldown=3)

epochs=20
```

```
batch_size=32

from tensorflow.keras.models import load_model
loaded_model = 

    load_model("C:/Users/dhanu/Desktop/Project/Models/VGG16/Epoch20/checkpoint_VGG16.h5"
    )
)
```

5.1.6.2 Fitting the Model :

```
history = loaded_model.fit(train_datagen.flow(X_train,Y_train, batch_size=batch_size),
                           epochs = epochs, validation_data = val_datagen.flow(X_val, Y_val),
                           verbose = 1, steps_per_epoch=(X_train.shape[0] // batch_size),
                           validation_steps=(X_val.shape[0] // batch_size),
                           callbacks=[learning_rate_reduction,checkpoint])

#model.save("/content/drive/MyDrive/Project/Models/VGG16/Epoch1/VGG16_1.h5")

loaded_model.save('C:/Users/dhanu/Desktop/Project/Models/VGG16/Epoch20/VGG16.h5')
```

5.1.7 App.py

```
from flask import Flask, request, render_template, flash, redirect, url_for
from PIL import Image
import numpy as np
from tensorflow.keras.models import load_model
import base64
import io
import imghdr
app = Flask(__name__)
app.secret_key = b'_5#y2L"F4Q8z\n\xec]/'

model = load_model("C:/Users/dhanu/Desktop/Project/ResNet10.h5")
```

```
classes = {  
    0: "actinic keratoses and intraepithelial carcinomae(Cancer)",  
    1: "basal cell carcinoma(Cancer)",  
    2: "benign keratosis-like lesions(Non-Cancerous)",  
    3: "dermatofibroma(Non-Cancerous)",  
    4: "melanocytic nevi(Non-Cancerous)",  
    5: "pyogenic granulomas and hemorrhage(Can lead to cancer)",  
    6: "melanoma(Cancer)",  
}
```

```
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg', 'gif'}
```

```
def allowed_file(filename):  
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS  
  
@app.route("/", methods=["GET", "POST"])  
  
def runhome():  
    return render_template("home.html")  
  
@app.route("/showresult", methods=["GET", "POST"])  
  
def show():  
    if request.method == "GET":  
        flash('Please upload an image.', 'warning')  
        return redirect(url_for("runhome"))  
    if 'pic' not in request.files or request.files['pic'].filename == "":
```

Epidermal Neoplasm Detection Using CNN

```
flash('No file selected.', 'warning')

    return redirect(request.url)
pic = request.files["pic"]

if not allowed_file(pic.filename):

    flash('INVALID FILE EXTENSION. Please upload an image file with extensions: .png, .jpg, .jpeg, .gif', 'warning')

    return redirect(request.url)

img_bytes = pic.read()

img_type = imghdr.what(None, img_bytes)

if img_type not in ['jpeg', 'png', 'gif']:

    flash('Invalid image type. Please upload a valid image file (jpeg, png, gif)', 'warning')

    return redirect(request.url)

inputimg = Image.open(io.BytesIO(img_bytes))

if inputimg.mode != 'RGB':

    inputimg = inputimg.convert('RGB')

inputimg = inputimg.resize((128, 128))

img = np.array(inputimg).reshape(-1, 128, 128, 3)

result = model.predict(img)

result = result.tolist()

max_prob = max(result[0])

class_ind = result[0].index(max_prob)

result = classes[class_ind]

image_data = base64.b64encode(img_bytes).decode("utf-8")

if class_ind == 0:

    info = "Actinic keratosis also known as solar keratosis or senile keratosis are names given to
```

Epidermal Neoplasm Detection Using CNN

intraepithelial keratinocyte dysplasia. As such they are a pre-malignant lesion or in situ squamous cell carcinomas and thus a malignant lesion."

elif class_ind == 1:

info = "Basal cell carcinoma is a type of skin cancer. Basal cell carcinoma begins in the basal cells — a type of cell within the skin that produces new skin cells as old ones die off. Basal cell carcinoma often appears as a slightly transparent bump on the skin, though it can take other forms. Basal cell carcinoma occurs most often on areas of the skin that are exposed to the sun, such as your head and neck"

elif class_ind == 2:

info = "Benign lichenoid keratosis (BLK) usually presents as a solitary lesion that occurs predominantly on the trunk and upper extremities in middle-aged women. The pathogenesis of BLK is unclear; however, it has been suggested that BLK may be associated with the inflammatory stage of regressing solar lentigo (SL1)"

elif class_ind == 3:

info = "Dermatofibromas are small, noncancerous (benign) skin growths that can develop anywhere on the body but most often appear on the lower legs, upper arms or upper back. These nodules are common in adults but are rare in children. They can be pink, gray, red or brown in color and may change color over the years. They are firm and often feel like a stone under the skin. "

elif class_ind == 4:

info = "A melanocytic nevus (also known as nevocytic nevus, nevus-cell nevus and commonly as a mole) is a type of melanocytic tumor that contains nevus cells. Some sources equate the term mole with 'melanocytic nevus', but there are also sources that equate the term mole with any nevus form."

elif class_ind == 5:

info = "Pyogenic granulomas are skin growths that are small, round, and usually bloody red in color. They tend to bleed because they contain a large number of blood vessels. They're also known as lobular capillary hemangioma or granuloma telangiectaticum."

elif class_ind == 6:

info = "Melanoma, the most serious type of skin cancer, develops in the cells (melanocytes) that produce melanin — the pigment that gives your skin its color. Melanoma can also form in your eyes and, rarely, inside your body, such as in your nose or throat. The exact cause of all melanomas isn't clear, but exposure to ultraviolet (UV) radiation from sunlight or tanning lamps and beds increases your risk of developing melanoma."

```
else :  
  
    info = "Please Upload the Skin Cancer Image, The Image Uploaded is not a Skin Cancer  
    Image"  
  
    return render_template("reults.html", result=result, info=info, inputimg=image_data)  
  
if __name__ == "__main__":  
  
    app.run(host="0.0.0.0", port=5000, debug=True)
```

5.2 IMPLEMENTATION

The proposed methodology is implemented using python programming language. The data set used for training the model is obtained from Kaggle. The dataset contains 10015 images. Several libraries are imported, including pickle, PIL, matplotlib, pandas, numpy, seaborn, keras, and others. These are necessary for various tasks such as data manipulation, visualization, and building and training neural networks. The code loads a dataset from a CSV file named "HAM10000_metadata.csv" containing information about skin lesions. Pandas is used to read the CSV file into a DataFrame. Seaborn and Matplotlib are used for visualization. sns.countplot is utilized to visualize the distribution of different classes of skin lesions.

A dictionary lesion_dict is created to map abbreviated class names to full class names. The DataFrame is modified to include a new column for the full class names (cell_type) and numerical codes for classes (dx_idx). The dataset is loaded using pickle, presumably containing preprocessed images and their corresponding labels.

The dataset is split into training and testing sets using train_test_split function from scikit-learn. Labels are one-hot encoded using to_categorical from Keras.

1. Data Collection: Data collection is a crucial step in various fields, including project, business, healthcare, and many others. The process involves gathering information.
2. Dataset Loading: Dataset loading is a crucial step ,we have to choose a dataset then download the dataset after we import required libraries then load the dataset.
3. Cleaning datset: Cleaning a dataset is a crucial step in the data preprocessing pipeline. It involves identifying and handling missing data, dealing with outliers, addressing inconsistent or erroneous entries, and ensuring that the data is in a format suitable for

analysis or model training.

4. Processing and Resizing Images: When working with image data, processing and resizing are common tasks, especially in the context of tasks like image classification or object detection.
5. Training and Test Split: This process helps you train your model on one subset, tune hyperparameters on another, and evaluate its performance on a third.
6. Feature Normalization: Normalization ensures that all features have similar scales, preventing some features from dominating others during the training process
7. Data Augmentation: Data augmentation is a technique used to artificially increase the size of a dataset by applying various transformations to the existing data.

5.2.1 Model Selection (ResNet50, Inception v3, VGG16):

RESNET 50 :

ResNet50, a pre-trained convolutional neural network (CNN) architecture, is loaded from Keras applications. The top layer (fully connected layer) is removed, and the remaining layers are marked as trainable. Additional layers are added on top of ResNet50 for fine-tuning, including GlobalAveragePooling2D, Dropout, and Dense layers with softmax activation for multi-class classification. The model is compiled with Adam optimizer and categorical cross-entropy loss.

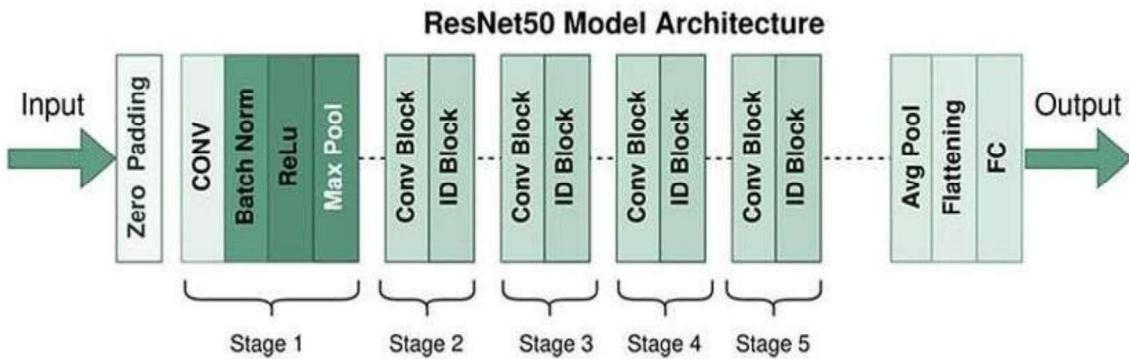


Fig.5.2.1.1: ResNet50 architecture

Inception v3 :

Inception v3 is a convolutional neural network (CNN) architecture developed by Google as part of the Inception family. It is designed for image classification and recognition tasks, and it has gained popularity due to its exceptional performance and efficiency. Inception v3 incorporates techniques such as batch normalization and factorized convolutions to accelerate training and improve generalization performance. Additionally, it utilizes aggressive regularization methods like dropout to prevent overfitting during training.

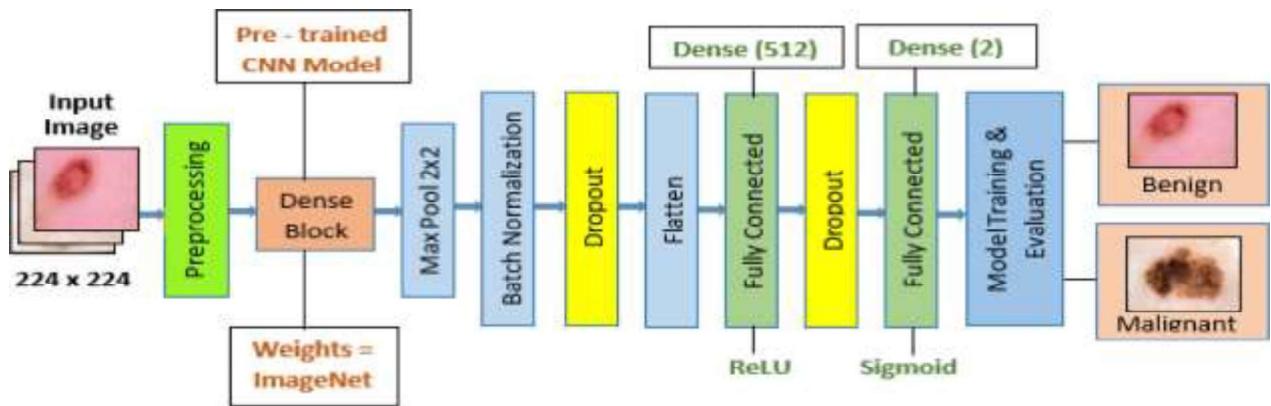


Fig.5.2.1.2: Inception v3 architecture

VGG16 :

VGG16 is a convolutional neural network (CNN) architecture proposed by the Visual Geometry Group (VGG) at the University of Oxford. It is renowned for its simplicity and effectiveness in image classification tasks. The "16" in VGG16 refers to the total number of weight layers, including convolutional and fully connected layers. One of the distinctive characteristics of VGG16 is its uniform architecture, where the convolutional layers consist of 3×3 filters with a stride of 1, and max-pooling layers are applied after every two convolutional blocks to reduce spatial dimensions. This simplicity allows for easy understanding and implementation of the model.

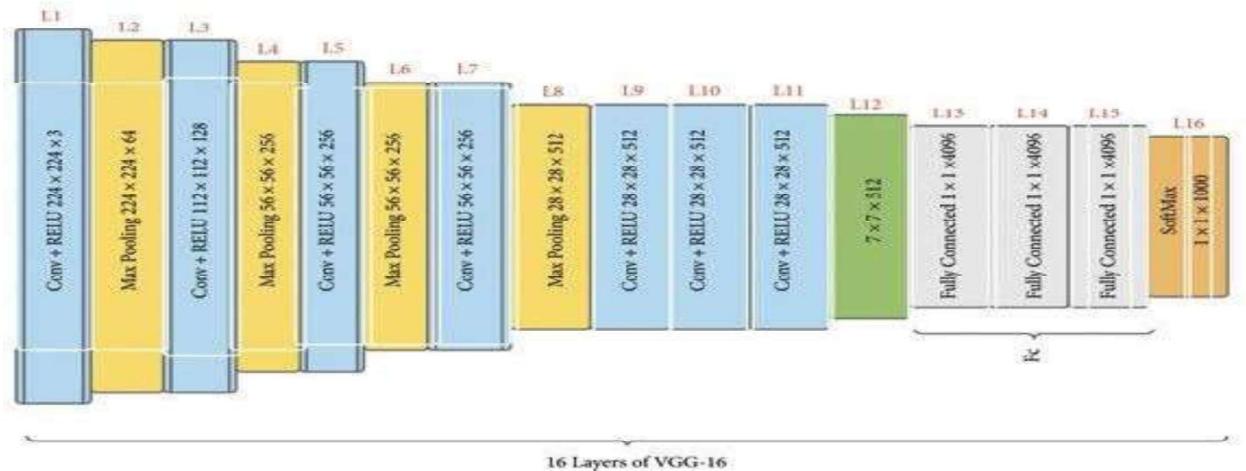


Fig.5.2.1.3: VGG16 architecture

1. Build Model: Building a machine learning model involves defining its architecture, compiling it, and training it on a dataset.
2. Compile Model: Compiling a model involves specifying additional settings that are necessary for training. This includes the choice of an optimizer, a loss function, and metrics for evaluation.
3. Fit the model: Fitting the model involves training it on your dataset.
4. Evaluate model: Evaluating a trained model involves assessing its performance on a separate dataset that it hasn't seen during training, often referred to as the test set. Considering the accuracies and loss of three models we have compared the performance of these models. And ResNet50 got the highest accuracy and lowest loss among three models.

5.2.2 Dataset preparation :

The HAM10000 dataset, a poignant acronym for "Human Against Machine with 10000 training images," emerges as an inspired initiative to invigorate the intersection of biology, medicine, and machine learning. Distinct from conventional digit classification datasets, this repository sparks enthusiasm among students, particularly in the realms of biology and medicine, by presenting a more captivating challenge.

Where traditional datasets falter in size and diversity, HAM10000 rises to the occasion. Curated with meticulous attention, 10,015 dermatoscopic pictures from different demographics and collection techniques are stored in it. This deliberate diversity enhances its utility as a robust training set, encompassing a spectrum of challenges encountered in real-world clinical scenarios.

HAM10000 delineates itself by offering a nuanced exploration of diagnostic realms related to pigmented skin lesions. With diagnostic categories spanning The collection of data includes vascular tumors, dermatofibroma, melanoma, carcinoma of basal cells, benign keratosis-like tumours, and melanoma. mirrors the complexity inherent in dermatology diagnostics.

Beyond the image pixels, each entry in the dataset is a repository of rich contextual information. gender, age, diagnostic(dx), diagnostic type (dx_type), lesion_id, image_id, and localization contribute to a multifaceted understanding of each case, elevating the dataset's potential for academic machine learning pursuits.

Setting a benchmark for rigor, HAM10000 surpasses the 50% mark in histopathological confirmation (histo) as the ground truth.

One of the standout features is the introduction of lesion tracking. This trailblazing concept allows the correlation of multiple images with a specific lesion, providing a dynamic dimension to the dataset. Projectors can trace the evolutionary trajectory or variations within a particular skin lesion.

In a departure from convention, HAM10000 keeps its test set veiled, adding an element of suspense and intrigue. Projectors are encouraged to engage with the official test set on the dedicated evaluation server to uphold fairness in model evaluations.

In essence, the HAM10000 dataset etches its uniqueness through a confluence of diversity, meticulous context, and pioneering features. Beyond a mere repository of images, it stands as a catalyst for innovation and exploration in the realm of skin cancer diagnosis through

Epidermal Neoplasm Detection Using CNN

machine learning.

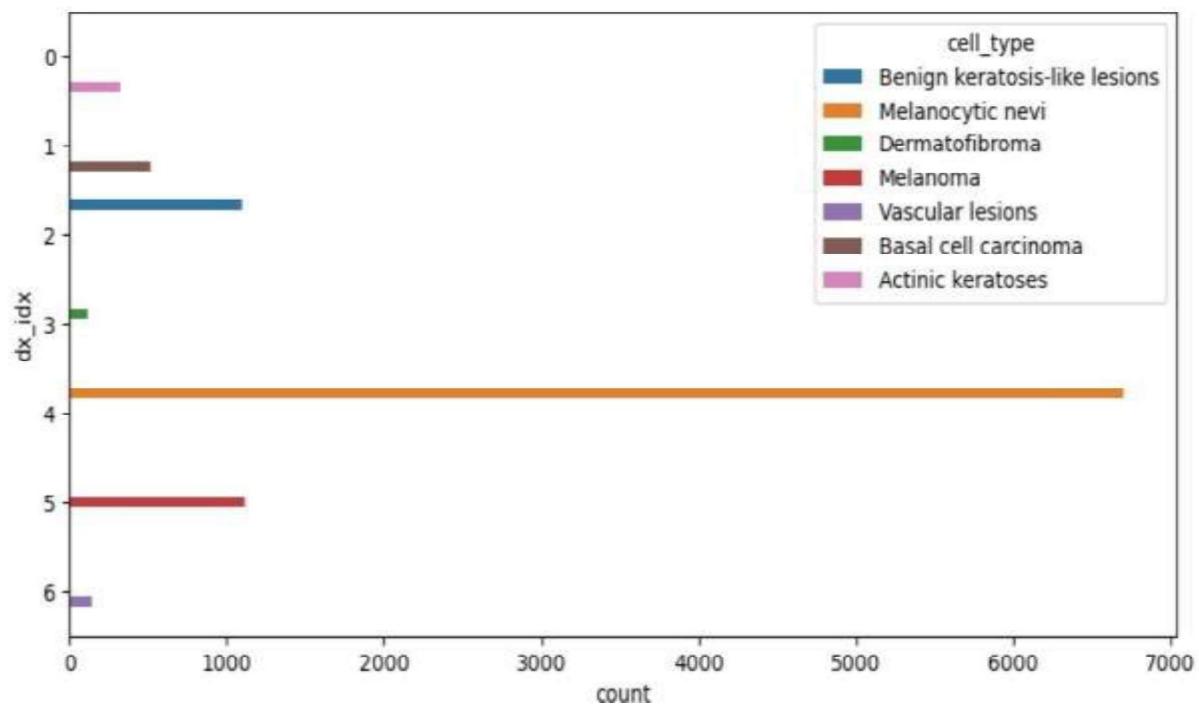


Fig.5.2.2.1: Dataset Description



Epidermal Neoplasm Detection Using CNN



Fig.5.2.2.2: Dataset Images

6. SYSTEM TESTING

6.1 OVERVIEW OF TESTING

Software testing is a process that will be used to evaluate software quality. Software testing is a powerful technology test designed to provide participants with information about the quality of a tested product or service, depending on the context in which it is intended to operate. This information may include, but is not limited to, the application process or application for tracking errors. Quality is not absolute, it is the value of one person. It is for this reason that tests may not be fully detected by the accuracy of any software, and criticism of tests or comparisons that compare the status and behavior of the product is kept to a minimum. It is very important to ensure that software testing should be separated from a separate discipline, Software Quality Assurance (S. Q. A.), which covers all areas of business processes, not just tests.

6.2 TYPES OF TESTING METHODS

Software testing methods are basically classified into BLACK BOX TESTING and WHITE-BOX TESTING. These two methods can be used to explain the point that a test engineer takes when developing the test cases.

6.2.1 Black Box Testing

The black box test works with software as a black box without understanding internal behavior. It aims to test performance. Therefore, the tester enters the data and sees only the output of the test object. This standard of testing often requires that the full test cases provided to the inspector can simply confirm that of the given input, the amount of output (or character), is the same as the expected value stated in the test case. Methods of checking black boxes include: stock sorting, boundary analysis, all double checking, fuzz testing, model-based testing, matrix tracking etc.

6.2.2 White Box Testing

The white box test, however, is when the tester achieves internal data structure, code

and algorithms. White box check methods include creating tests to satisfy other coding methods. For example, a test designer can create a test to ensure that all statements in the system are made at least once. Some examples of white box tests are dating methods and error injection changes. The white box test includes all specific tests.

Testing can be done on the following levels

6.2.3 Unit Testing:

Unit test tests for minimum component or module. Each unit (basic component) of the software has been tested to ensure that the detailed structure of the block is performed correctly. In an object-focused environment, this is usually done in the classroom, and testing of small units involves builders and destroyers.

6.2.4 Integration Testing

Integration testing reveals defects in interface and interactions between integrated components (modules). Largely continuous groups of tested software components related to building materials are compiled and tested till the software works as a system.

6.2.5 System Testing

System testing tests a fully integrated system to ensure it meets its requirements. System integration testing ensures that the system is integrated into any external or third-party programs defined in system requirements.

7. RESULTS

The model outputs reveal distinct performance characteristics among ResNet50, Inception v3, and VGG16 in skin cancer classification. ResNet50 demonstrates exceptional predictive capabilities, achieving a test success percent of 98.9% and a validation success percent of 99.3%, with minimal losses. Inception v3, while providing commendable results, exhibits a lower test and validation success rate of 94.81% and 93.77%, respectively, with comparatively higher losses. VGG16, although delivering acceptable outcomes, falls behind ResNet50 and Inception v3, with a test success rate of 93.86% and a validation success rate of 93.27%. The detailed confusion matrices further illustrate the models' classification performance, emphasizing ResNet50's superiority in accurately identifying various skin cancer types. These findings underscore the significance of selecting an appropriate deep-learning architecture, with ResNet50 emerging as a robust choice for skin cancer diagnosis in this context. Here are the screenshots of output:

ResNet50:

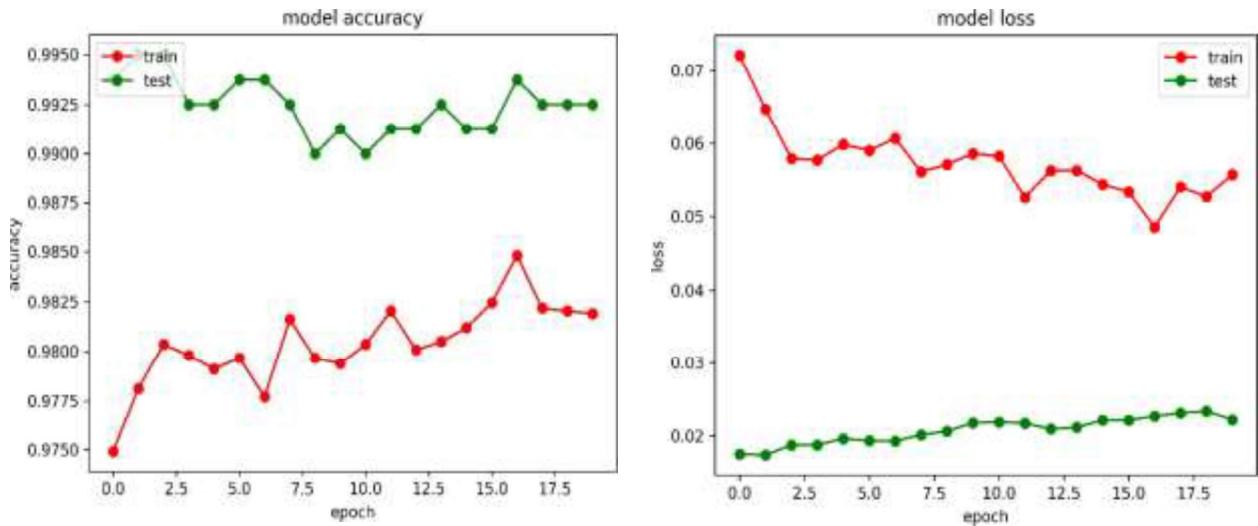


Fig.7.1: ResNet50 Accuracy & Loss

Inception v3:

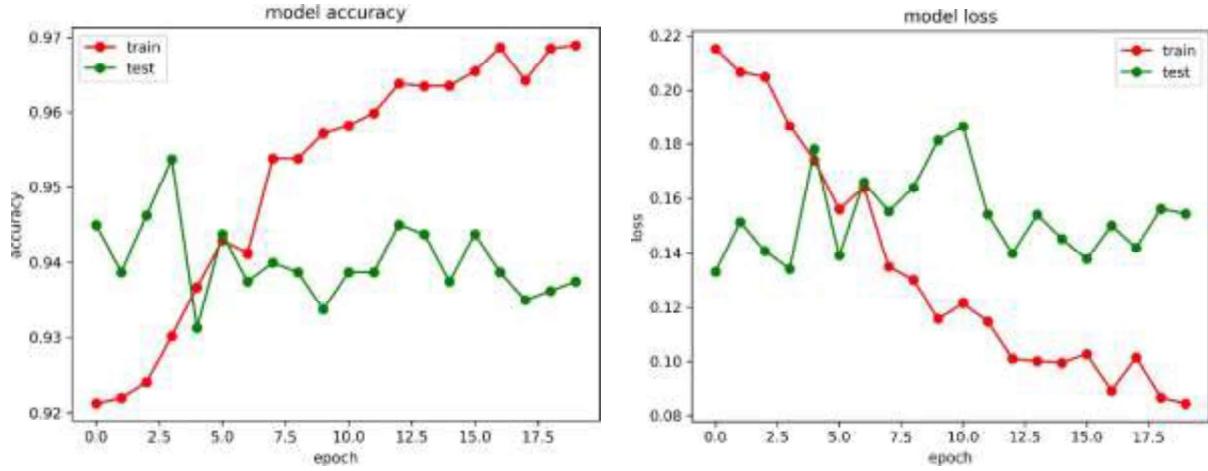


Fig.7.2: Inception v3 Accuracy & Loss

VGG16 :

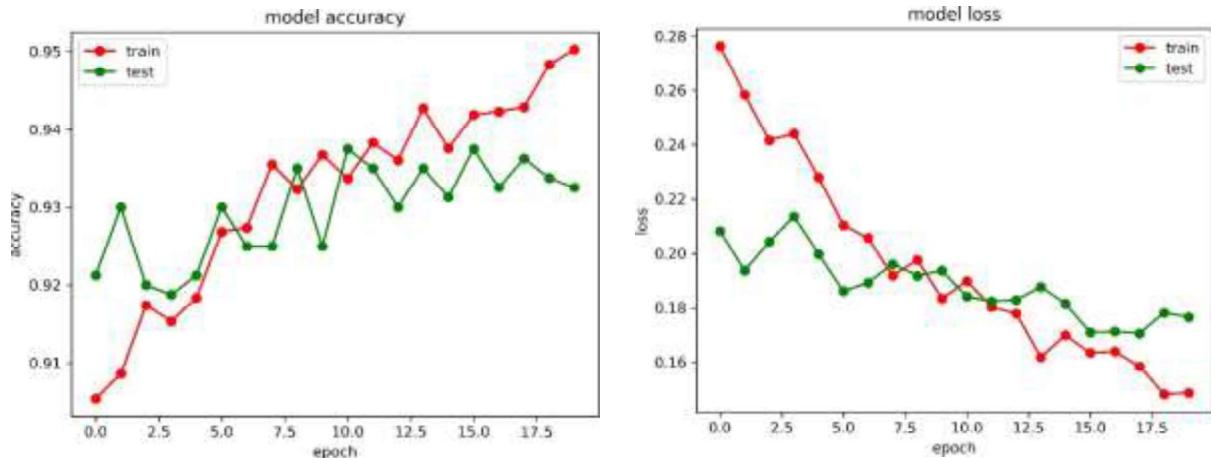


Fig.7.3: VGG16 Accuracy & Loss

User Interface:

A user interface is created where it takes the skin image as the input and gives the predicted class as the output. The following pictures shows the user interface:

Epidermal Neoplasm Detection Using CNN

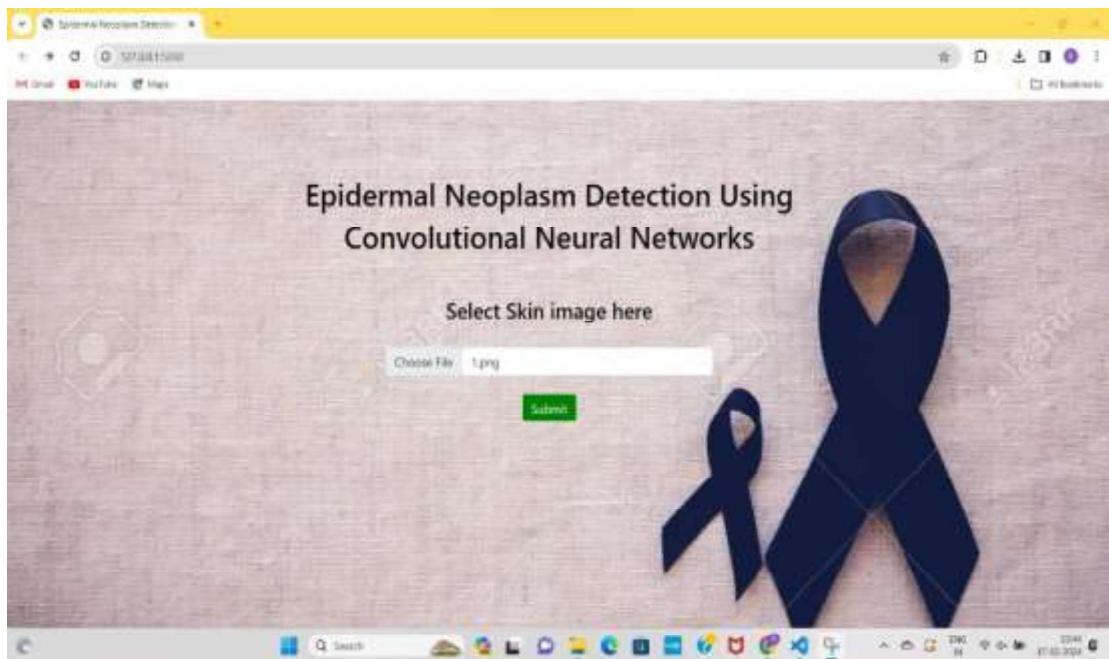


Fig.7.4: Home Page

Outputs of 7 classes:

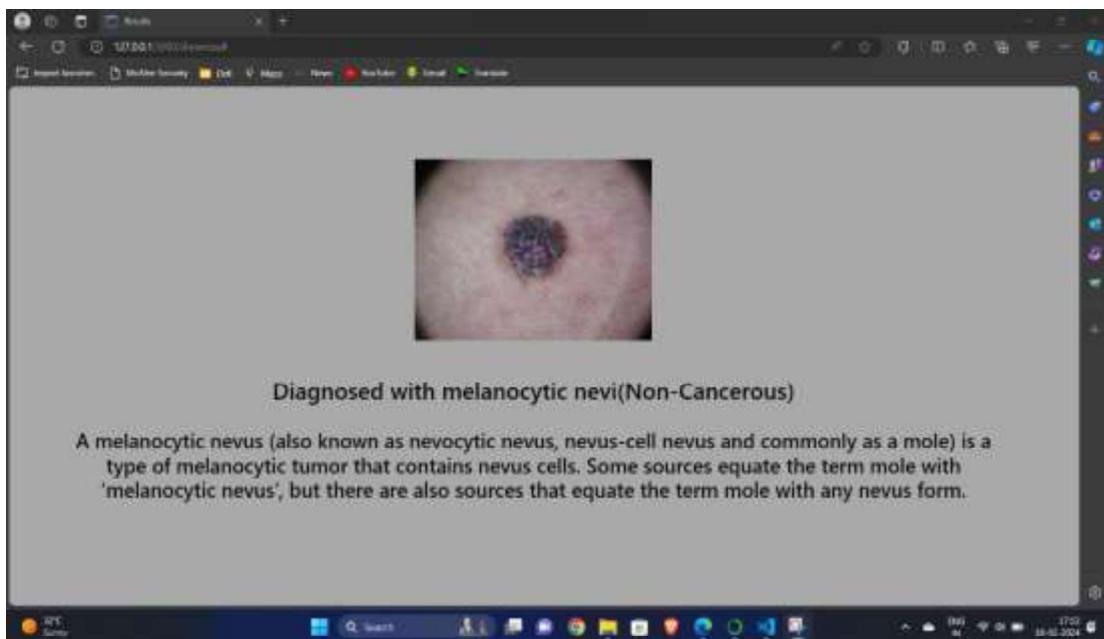


Fig.7.5: Melanocytic Nevi Prediction

Epidermal Neoplasm Detection Using CNN

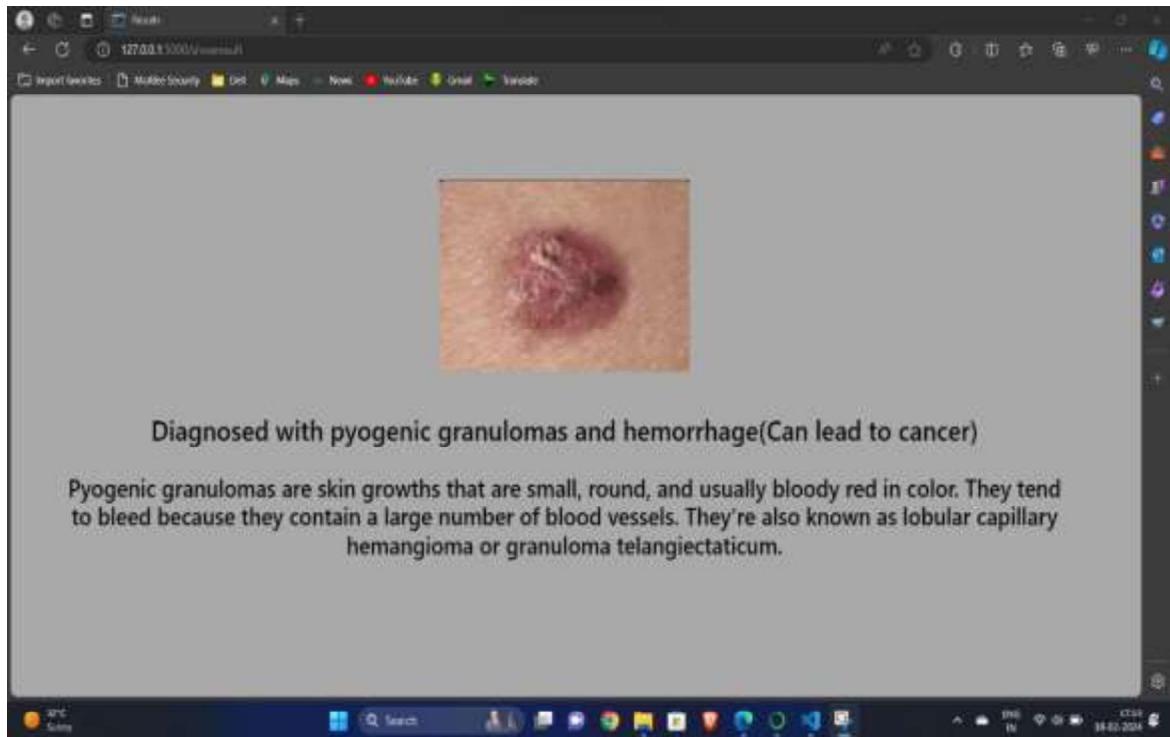


Fig.7.6: Pyogenic Granulomas Prediction

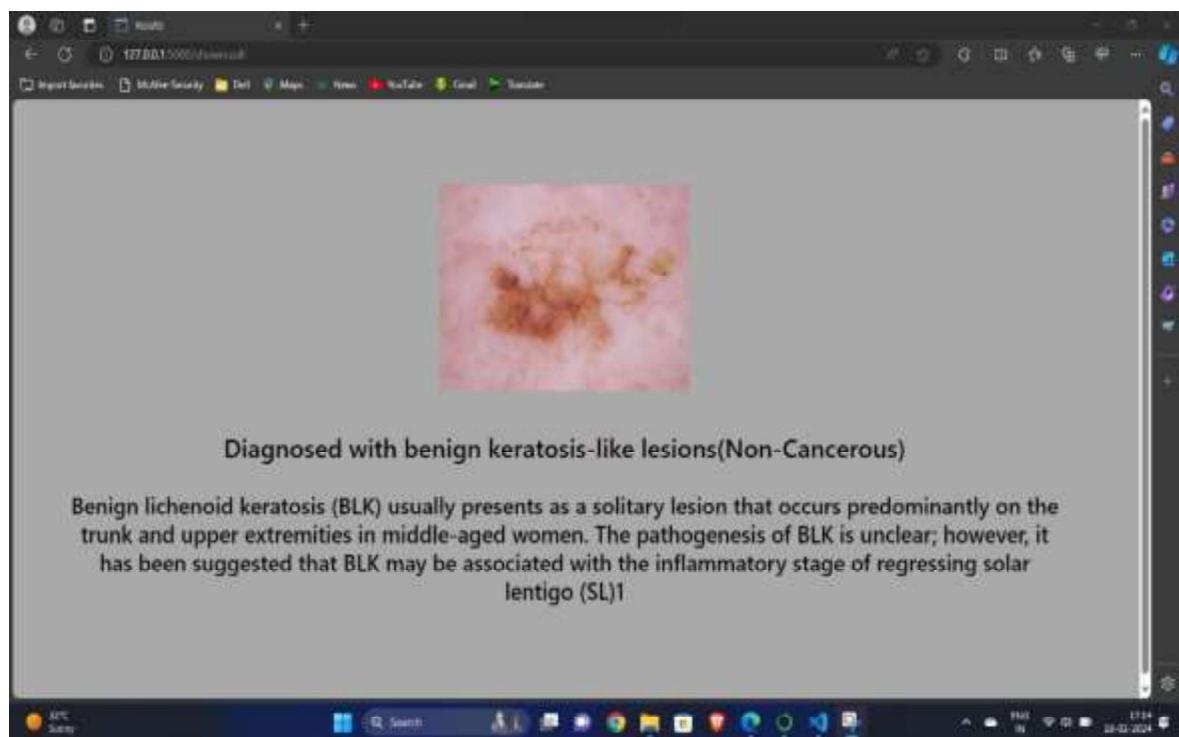


Fig.7.7: Benign Prediction

Epidermal Neoplasm Detection Using CNN

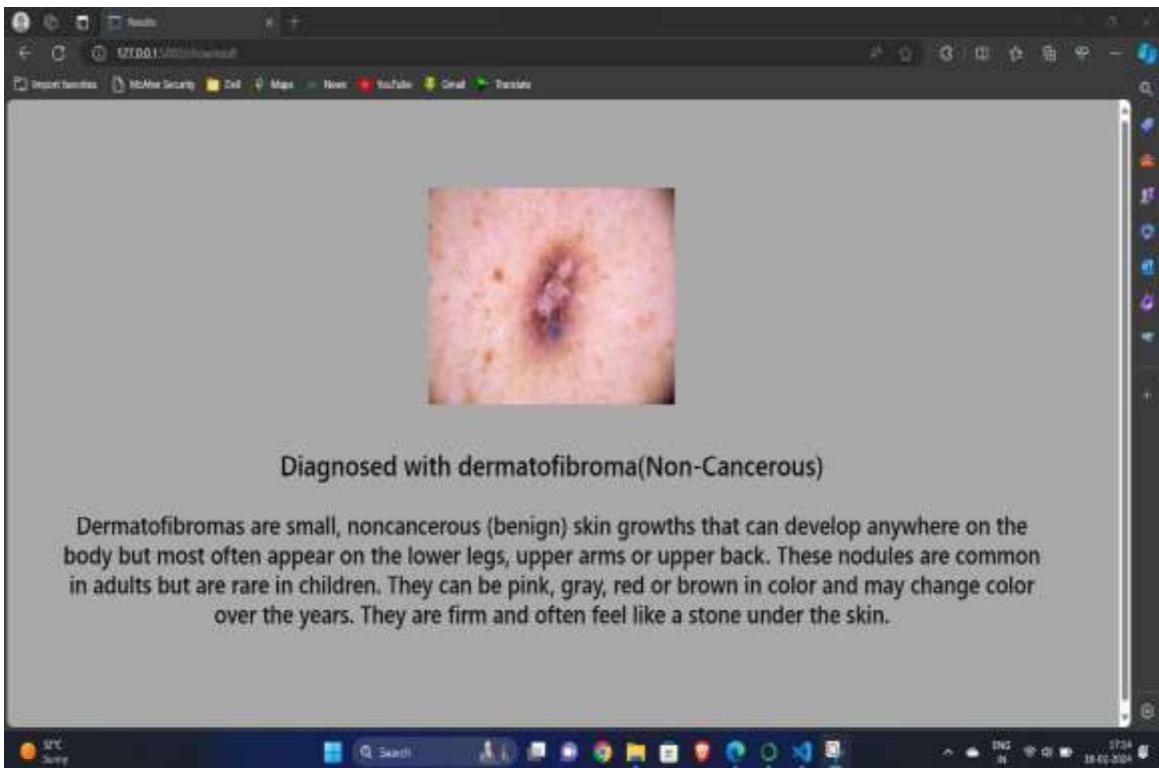


Fig.7.8: Dermatofibroma Prediction

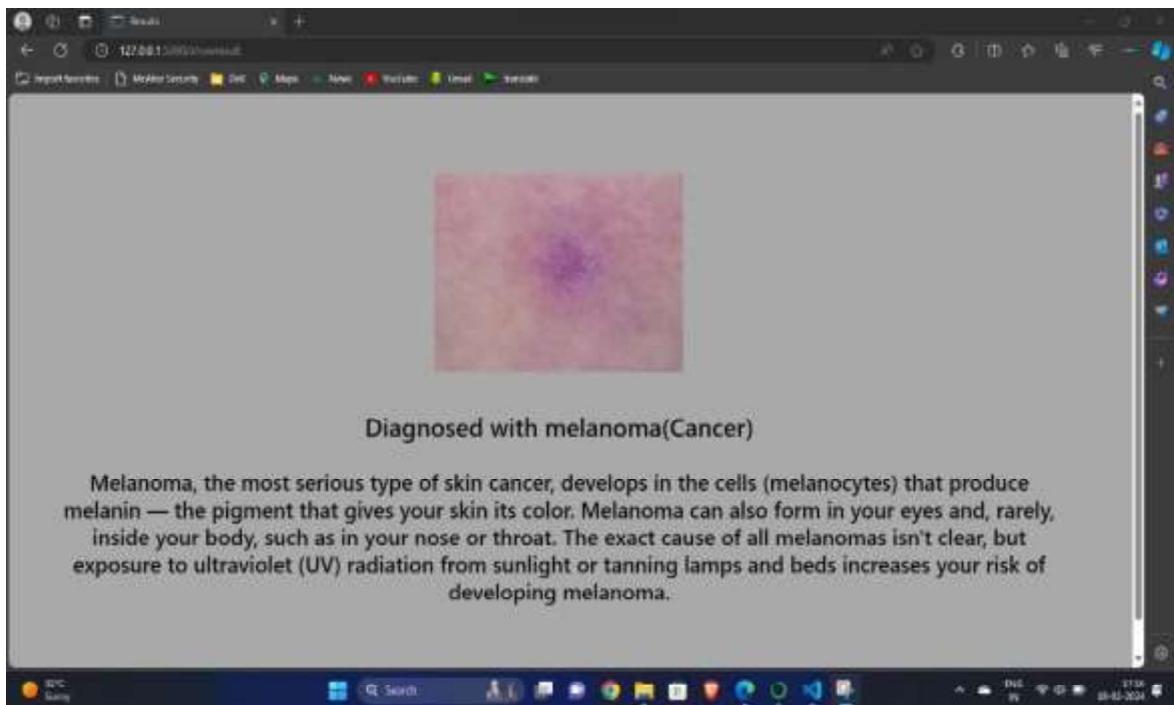


Fig.7.9: Melanoma Prediction

Epidermal Neoplasm Detection Using CNN

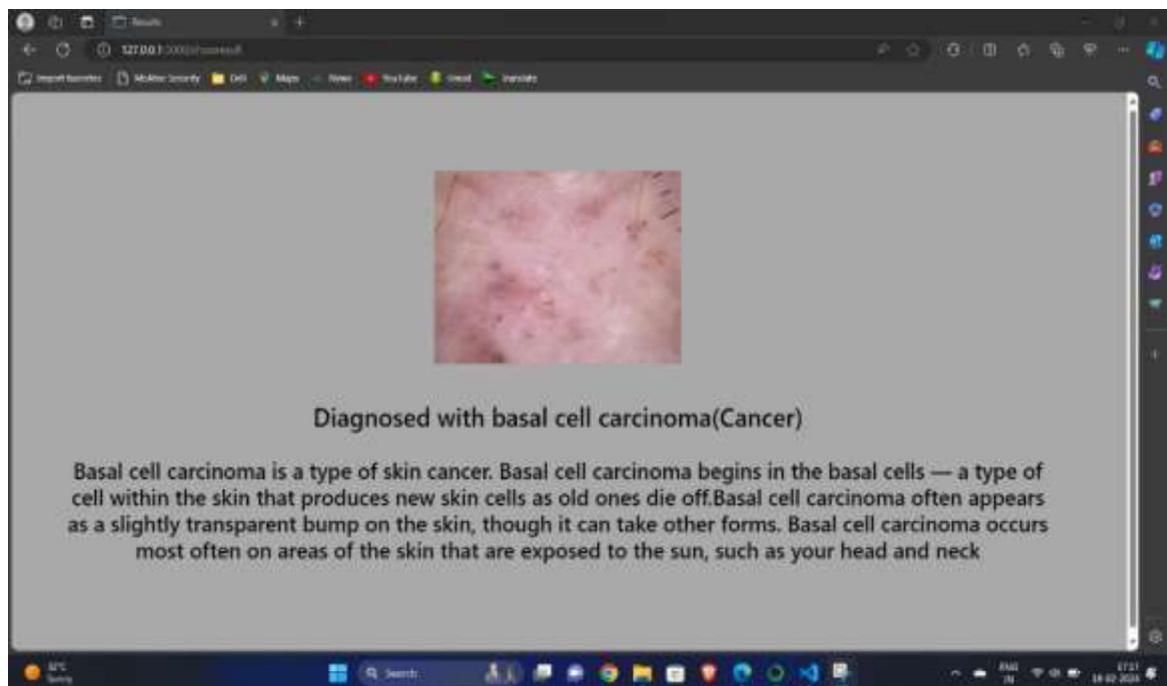


Fig.7.10: Basal Cell Carcinoma Prediction

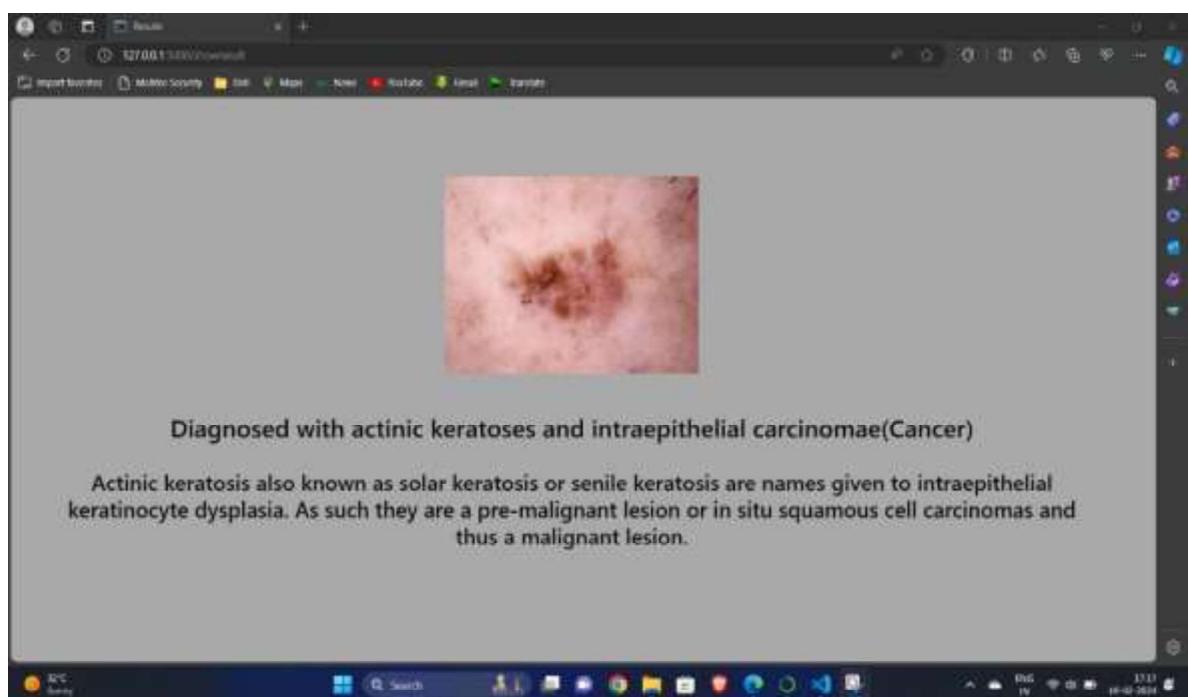


Fig.7.11: Actinic Keratosis Prediction

8. CONCLUSION

The comparative analysis of ResNet50, Inception v3, and VGG16 highlights ResNet50 as the most effective method used in this investigation to classify skin tumours. It outperforms the other models in terms of accuracy and loss metrics. However, considerations such as computational efficiency and model complexity should be weighed when choosing the most suitable model for a specific use case.

In closing, the study successfully illustrates the possibility of models based on deep learning for the categorization of skin tumors, with ResNet50 exhibiting superior performance in accuracy and minimal loss. While ResNet50 emerges as a robust choice, acknowledging the distinct strengths of Inception v3 and VGG16 emphasizes the importance of a tailored model selection based on specific needs. The results underline the nature of model comparisons and the significance of understanding individual architectural characteristics.

9. REFERENCES

- [1] Vijayalakshmi, M. M. "Melanoma skin cancer detection using image processing and machine learning." International Journal of Trend in Scientific Project and Development (IJTSRD) 3.4 (2019): 780-784.
- [2] Harangi, Balazs. "Skin lesion classification with ensembles of deep convolutional neural networks." Journal of biomedical informatics 86 (2018): 25-32.
- [3] Thakur, Nikita, and Arunima Jaiswal. "Comparative Analysis of Machine Learning and Deep Learning Algorithms for Skin Cancer Detection." Machine Intelligence and Data Science Applications: Proceedings of MIDAS 2021. Singapore: Springer Nature Singapore, 2022. 409-418.
- [4] Gong, An, Xinjie Yao, and Wei Lin. "Classification for dermoscopy images using convolutional neural networks based on the ensemble of individual advantage and group decision." IEEE Access 8 (2020): 155337-155351.
- [5] Ali, Karar, et al. "Multiclass skin cancer classification using EfficientNets—a first step towards preventing skin cancer." Neuroscience Informatics 2.4 (2022): 100034.
- [6] Pacheco, Andre GC, and Renato A. Krohling. "An attention-based mechanism to combine images and metadata in deep learning models applied to skin cancer classification." IEEE journal of biomedical and health informatics 25.9 (2021): 3554-3563.
- [7] Imran, Azhar, et al. "Skin cancer detection using combined decision of deep learners." IEEE Access 10 (2022): 118198-118212.
- [8] Alizadeh, Seyed Mohammad, and Ali Mahloojifar. "Automatic skin cancer detection in dermoscopy images by combining convolutional neural networks and texture features." International Journal of Imaging Systems and Technology 31.2 (2021): 695-707.
- [9] Han, Seung Seog, et al. "Keratinocytic skin cancer detection on the face using region-based convolutional neural network." JAMA dermatology 156.1 (2020): 29-37.
- [10] Tschandl, Philipp, et al. "Expert-level diagnosis of non-pigmented skin cancer by combined convolutional neural networks." JAMA dermatology 155.1 (2019): 58-65.
- [11] Haggenmüller, Sarah, et al. "Skin cancer classification via convolutional neural networks:

systematic review of studies involving human experts." European Journal of Cancer 156 (2021): 202- 216.

[12] Albahar, Marwan Ali, "Skin lesion classification using convolutional neural network with novel regularizer." IEEE Access 7 (2019): 38306-38313.

[13] Harangi, Balazs. "Skin lesion classification with ensembles of deep convolutional neural networks." Journal of Biomedical Informatics 86 (2018): 25-32.

[14] F. Nachbar, W. Stolz, T. Merkle, A.B. Cognetta, T. Vogt, M. Landthaler, G. Plewig, The ABCD rule of dermatoscopy: high prospective value in the diagnosis of doubtful melanocytic skin lesions, J. Am. Acad. Dermatol. 30 (4) (1994) 551–559.s.s.

[15] Y. Faziloglu, R.J. Stanley, R.H. Moss, W.V. Stoecker, R.P. McLean, Colour histogram analysis for melanoma discrimination in clinical images, Skin Res. Technol. 9 (2003) 147–155.

[16] A. Esteva, B. Kuprel, R.A. Novoa, J. Ko, S.M. Swetter, H.M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, Nature 542 (7639) (2017) 115–118.

[17] Shanthi, T., R. S. Sabeenian, and R. Anand. "Automatic diagnosis of skin diseases using convolution neural network." Microprocessors and Microsystems 76 (2020): 103074.

[18] Veronika Cheplygina, marleen de bruijne, josien p. W. Pluim, "not-so-supervised: a survey of semisupervised, multi-instance, and transfer Learning in medical image analysis" arxiv:1804.06353v2 [cs.cv] 14 sep 2018.

[19] salome kazeminia, christoph baur, arjan kuijper, bram van Ginneken, nassir navab, shadi albarqouni, anirban mukhopadhyay "gans for medical image analysis ", arxiv:1809.06222v2 [cs.cv] 21 dec 2018.

[20] L.Yu, H. Chen, Q. Dou, J. Qin, P.A. Heng, Automated melanoma recognition in dermoscopy images via very deep residual networks, IEEE Trans. Med. Imaging 36 (4) (2017) 994-1004.

EPIDERMAL NEOPLASM DETECTION USING CONVOLUTIONAL NEURAL NETWORKS

1st Dr.Ch.Venkata Narayana
Department of Computer Science and Engineering
Lakireddy Bali Reddy College of Engineering (Autonomous)
Mylavaram, India
cvnreddy.chejarla@gmail.com

4th Darapureddy Dhanush Kumar
Department of Computer Science and Engineering
Lakireddy Bali Reddy College of Engineering (Autonomous)
Mylavaram, India
dhanushdarapureddy9358@gmail.com

2nd Elike.Stella
Department of Computer Science and Engineering
Lakireddy Bali Reddy College of Engineering (Autonomous)
Mylavaram, India
elikestellaa24@gmail.com

3rd Yandrapathi Harshitha Sampath
Department of Computer Science and Engineering
Lakireddy Bali Reddy College of Engineering (Autonomous)
Mylavaram, India
harshisam2003@gmail.com

Abstract— Skin cancer is the dangerous medical condition which is observed by the abnormal growth of skin cells. It is majorly caused due to ultraviolet radiations. Detecting it in the early stage may reduce its severity. This research delves into a comprehensive analysis of skin cancer prediction, employing three distinct deep learning models: ResNet50, Inceptionv3 and VGG16. Utilizing a diverse dataset comprising dermoscopic images, the study rigorously evaluates and compares the performance of these models across various metrics such as accuracy, sensitivity, and specificity. Also we have created an user interface to test the skin cancer images. After a great deal of testing, ResNet50 is the best option for detecting skin-cancer since it can identify a wide range of skin tumors with greater accuracy. The research incorporates advanced data augmentation techniques to bolster model resilience and generalization, ensuring robust performance in real-world scenarios. This highlights its potential significance in clinical applications, highlighting how deep machine learning is essential to improving medicinal image processing and improving the results for patients.

Keywords— ResNet50, Inceptionv3, VGG16, Dermoscopic Images, Comparative Analysis.

I. INTRODUCTION

The surging global incidence of skin cancer, a pervasive and potentially fatal affliction, has intensified the demand for innovative diagnostic strategies. Groundbreaking initiatives have prepared the way for the identification of skin cancer that is melanoma, leveraging a synthesis of image processing and machine learning methodologies [1]. The study of lesions of the skin categorization using ensembles of deep convolutional neural networks opened up fresh possibilities for progress and marked a significant turning point in the field of artificial intelligence [2].

An important next step in the identification of skin tumors is to compare algorithms based on deep-learning and machine-learning [3]. An innovative approach integrating individual advantage and group decision ensembles to amplify classification accuracy marked a significant stride towards preventive diagnostic measures [4]. The significance of feature variety in automated skin tumor diagnosis was highlighted by research conducted on skin cancer identification using integrated decision making driven by deeper drivers [7]. Integration of convolutional neural networks with texture features highlighted the importance of feature diversity in automating skin cancer detection [8].

A revisit to ensemble learning underscored its persistent efficacy in skin lesion classification, contributing to the discourse on ensemble learning in dermatological diagnostics [13]. An important turning point in the merging of dermatologists and technology was the investigation of applications for computers in the detection and prediction of cancer of the skin [16]. Many researches have been made in this skin cancer detection. Mostly they have predicted melanoma since it is very common type of skin cancer. They have used many machine learning and deep learning models like SVM, Mobilenet, inception, sequential models, vgg19,Resnet and so on. Other types of skin cancers were also predicted such as benign, melanoma, basal cell carcinoma etc.

In our research we have taken three models such as ResNet50, Inceptionv3 and VGG16. These 3 models are compared to know which model is giving the best accuracy for skin cancer detection. We trained these models on HAM10000 dataset which contains 10015 dermoscopic images. The dataset is preprocessed in a wide range before giving it to the model. Each model is trained with the dataset one by one and we have taken the observations from the accuracy and loss given by the models. From the observations drawn we have concluded that Resnet50 gives the highest accuracy compared to other 2. We have created a user interface which takes skin image as input and predicts the class.

In this research first section consists of brief description on skin cancer and the models we used. Later it is followed by the significance of skin cancer detection and the introduction of CNN importance in this research with a small note of previous works done by the dermatologists and researchers on this research. Next section consists of the related work done on this research. Next followed by methodology and continues with the experimental setup. In the last section we have given the results with the comparison between 3 models followed by future scope and conclusion.

II. LITERATURE SURVEY

Skin tumors, especially a form of is a major worldwide medical problem that requires sophisticated detection techniques. Significant progress in the identification of skin tumors has been sparked by current developments in computational biology and artificial intelligence. The

groundwork for further research was laid by Vijayalakshmi [1], who led investigations into identifying melanoma using computational imaging and artificial intelligence. Dermal lesion categorization underwent an evolution with the introduction of deep learning, namely convolutional artificial neural networks (CNNs).

Harangi [2] introduced ensemble methods with deep CNNs for skin lesion classification, enhancing diagnostic accuracy. Gong et al. [4] explored the classification of dermoscopy images using CNNs, emphasizing the effectiveness of ensemble strategies.

Efficient Nets gained prominence in multiclass skin cancer classification, as demonstrated by Ali et al. [5]. Pacheco and Krohling [6] proposed, combined for improved classification. Imran et al. [7] highlighted the significance of combined decision-making by deep learners in skin cancer detection.

Alizadeh and Mahloojifar [8] integrated CNNs with texture features for automatic skin cancer detection, showcasing the synergy of different methodologies. Han et al. [9] focused on keratinocytic skin cancer detection, leveraging region-based CNNs for enhanced accuracy. Tschandl et al. [10] achieved expert-level diagnosis using combined CNNs, underscoring the potential for leveraging deep learning in clinical settings.

A systematic review by Haggenmüller et al. [11] emphasized the growing body of literature on skin cancer classification via CNNs. Albahar [12] introduced a novel regularizer in CNNs for improved skin lesion classification. In addition to algorithmic advancements, several studies explored the integration of clinical knowledge into the diagnostic process. The ABCDE rule, proposed by Nachbar et al. [15], provided a valuable framework for dermoscopic analysis. Faziloglu et al. [14] contributed to melanoma discrimination using color histogram analysis, showcasing the interdisciplinary nature of skin cancer research.

The latest findings, as revealed by the studies by Esteva et al. [17], showed that deep brain networks could classify skin cancer at the physician level, underscoring the transformative potential of artificial intelligence in clinical practice. Shanthi et al. [18] furthered this progress by proposing an automatic diagnosis system based on AI.

In summary, the integration of deep neural networks, artificial intelligence, and computer vision has ushered in an entirely novel phase in skin tumor diagnosis. The cited works collectively contribute to a robust foundation for future research, emphasizing the interdisciplinary collaboration between medical professionals and technologists.

III. METHODOLOGY

3.1 COMPONENTS

The initiative to classify skin tumors uses a structure known as a neural network based on convolution (The CNN network), specifically utilizing ResNet50 as the backbone for feature extraction. The architecture can be summarized as follows:

Initially dataset is imbalanced and contains noise, hence data preprocessing is done before giving it to the model. First we have 10015 images after oversampling we got 46935 images, this will help in increasing the accuracy. Then we have replaced null values with standard deviation and removed the noise in the data.

After data is splitted into training (80%) and testing (20%). Next we have reshaped the image to 128 x 128 pixels. The input layer processes dermoscopic images resized to 128x128 pixels. These images serve as the raw input for the subsequent layers.

The ResNet50 architecture is utilized for its ability to capture complex hierarchical features. It is composed of many remaining blocks with disconnects, rectified linear unit (ReLU) activations, batch normalization, and layers of convolution in every.

Global Average Pooling(GAP) Layer a World means pooling level is implemented after the layers of convolution to offer an efficient representation of learned information and to minimize physical dimensions. whatever the dimensions of the input data, Gaps calculates the mean of every feature map to produce a fixed-size vector.

In Fully Connected layer the result of the Gaps level is linked to the final forecasting level by the use of an entirely linked layer. Understanding intricate correlations between the collected characteristics is made easier by this component.

Softmax Activation is a function where the result of the completely linked layer is subjected to a soft max activated function, producing a probability distribution across the predefined skin cancer classes. This enables the model to assign probabilities to each class and make a final prediction.

The following training strategies are used in which the model is trained using the Adam optimizer, incorporating both momentum and adaptive learning rate techniques. The measure of loss is category cross-entropy, which quantifies the difference in actual and anticipated class densities.

To enhance model generalization, data augmentation techniques are applied during training. These include random rotations, shifts, and zooms, ensuring the model is exposed to diverse variations of skin lesion images.

The effectiveness of the framework is assessed on distinct test and verification datasets. Accuracy, along with a confusion matrix, is used to assess classification performance across multiple skin cancer categories.

Transfer learning is leveraged by initializing the ResNet50 backbone with pre-trained weights on a large dataset. By modification, a model may be made to fit the unique features of the skin cancer dataset that was employed for the present study.

Once trained, the model can be deployed to predict skin cancer types in new dermoscopic images. The research provides a comprehensive approach to automated skin lesion classification, aiding in early detection and diagnosis.

TABLE I. GENERALISED PARAMETER DESCRIPTION

Parameter	Description
Input Size	Standardized size, e.g., 128x128 pixels
Convolution Layers	Extract features through convolution operations
Pooling Layers	Down sample feature maps for spatial hierarchies
Dense Layers	Fully connected layers for classification
Activation Functions	ReLU (Rectified Linear Unit) for non-linearity

Dropout	Regularization technique to prevent overfitting
---------	---

3.2 BLOCK DIAGRAM

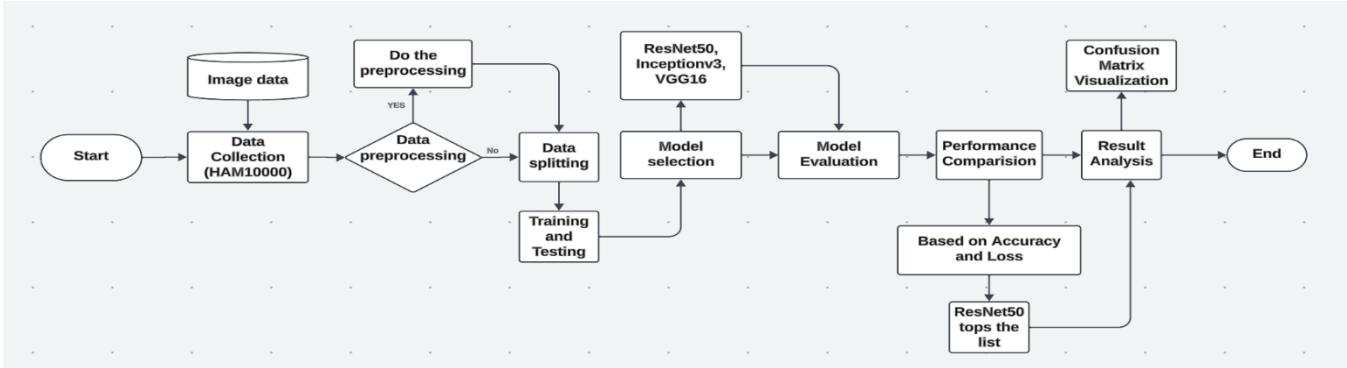


Fig 1. Overall Architecture

CNN Parameters:

TABLE II. CNN PARAMETERS

Layer	Parameter	Description	Mathematical Equation
Conv2D_1	Filters(Convolution)	Learnable filters applied to input data for feature extraction	$W_{out} = S(W_{in}-F+2P)/S+1$
	Filter Size	Size of the convolutional filter	$H_{out} = (H_{in}-F+2P)/S + 1$
MaxPooling_1	Pool Size	Factor by which to downscale	$W_{out}=(W_{in}-F)/S+1$
Flatten	-	Flatten layer output for dense layers	None
Dense_1	Neurons	Number of neurons in a dense layer	Output=Activation (Input×Weights+Biases)
	Activation Function	Activation function applied to layer output	$P(y_i) = (e^{x_i}) / (\sum_j e^{x_j})$

where,

The input tensor's width and height are represented by W_{in} and H_{in} .

Filter size is denoted by F , padding by P , and stride by S .

The values in the table are specific to each layer and are calculated based on the layer's parameters.

3.3 MODELS DESCRIPTION

ResNet50: ResNet50 is based on residual learning, introducing skip connections to address vanishing gradient issues. Its depth allows it to capture intricate features and learn complex representations. ResNet's skip connections aid in smoother gradient flow during training, making it relatively easier to train deep networks. This can result in faster convergence and reduced susceptibility to overfitting. ResNet's depth and skip connections contribute to superior performance in capturing intricate features. Because of its capacity to learn hierarchy visualizations, it frequently proves

useful for healthcare visualization applications. ResNet models, including ResNet50, are commonly used for transfer learning due to their pre-trained weights on large datasets. This can be advantageous when working with limited medical imaging datasets.

Inceptionv3: Inceptionv3, based on the Inception architecture, employs parallel filter operations of varying sizes within each module. This facilitates multi-scale feature extraction and can capture diverse patterns. Inception modules allow for efficient use of parameters, potentially reducing the risk of overfitting. However, training may require careful optimization due to the parallel structure. Inception's multi-scale feature extraction can be beneficial for skin lesion classification, capturing both fine and coarse details. However, fine-tuning may be necessary for optimal performance. Inception models may also benefit from transfer learning, but careful consideration is needed to open the past-trained segments to the particular characteristics of dermoscopic images.

VGG16: VGG16, following the VGGNet, opens a series of small convolutional filters in a consistent receptive field. It emphasizes the importance of depth in neural network architectures. VGG's straightforward architecture can make training relatively simple, but the model may require more data and computational resources to generalize well. VGG's simplicity may limit its ability to capture intricate features compared to more complex architectures. However, its straightforward design can yield competitive results, especially with sufficient data. VGG's transfer learning potential is notable, but its larger parameter count may require more data to avoid overfitting during fine-tuning.

3.4 ALGORITHM

Input:

X: Input image data

W: Convolution filter/kernel

b: Bias

s: Pooling stride

wi: Weights for the dense layer

bi: Bias for the dense layer

Step1: Convolutional Operation $Y[i,j] = \sum_m$

$\sum_n (X[I+m, j+n]) \times (W[m, n]) + b$

Perform convolution operation on the input image X using the kernel W and bias b to obtain the feature maps Y.

Step 2: Activation ReLU; $f(x)=\max(0,x)$

Apply the ReLU activation function element-wise to the feature maps Y to introduce non-linearity.

Step 3: Pooling Operation (Max Pooling): $[i,j]=\max_{m,n} X[i \times s+m, j \times s+n]$

Perform max pooling on the feature maps Y with a pooling window size of (m, n) and a stride of s to obtain the pooled feature maps.

Step 4: Flatten the pooled feature maps to prepare for input into the dense layer.

Step 5: Dense Layer Output: $Y=f(\sum_i w_i \times x_i + b)$

Compute the output of the dense layer by taking the dot product of the flattened feature maps with the weights w_i , adding the bias b_i , and applying the ReLU activation function.

Step 6: The output of the dense layer represents the probability of skin cancer presence.

IV. EXPERIMENTAL SETUP

4.1 DATASET DESCRIPTION

The HAM10000 dataset, a poignant acronym for "Human Against Machine with 10000 training images," emerges as an inspired initiative to invigorate the intersection of biology, medicine, and machine learning.

Where traditional datasets falter in size and diversity, HAM10000 rises to the occasion. Curated with meticulous attention, 10,015 dermoscopic pictures from different demographics and collection techniques are stored in it. This deliberate diversity enhances its utility as a robust training set, encompassing a spectrum of challenges encountered in real-world clinical scenarios.

The collection of data includes vascular tumors, dermatofibroma, melanoma, carcinoma of basal cells, benign keratosis-like tumors, and melanoma. mirrors the complexity inherent in dermatology diagnostics.

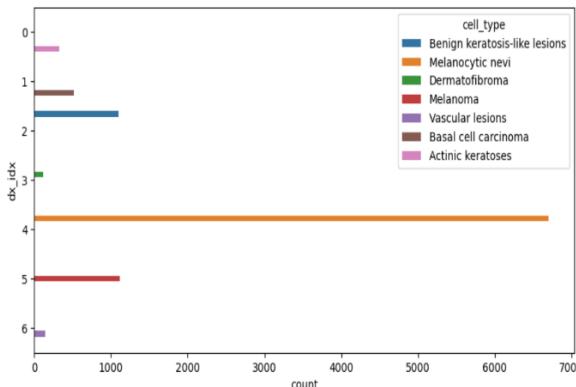


Fig 2. Dataset Description

Beyond the image pixels, each entry in the dataset is a repository of rich contextual information, gender, age, diagnostic (dx), diagnostic type (dx_type), lesion_id, image_id, and localization contribute to a multifaceted understanding of each case, elevating the dataset's potential

for academic machine learning pursuits.

Setting a benchmark for rigor, HAM10000 surpasses the 50% mark in histopathological confirmation (histo) as the ground truth.

In essence, the HAM10000 dataset etches its uniqueness through a confluence of diversity, meticulous context, and pioneering features. Beyond a mere repository of images, it stands as a catalyst for innovation and exploration in the realm of skin cancer diagnosis through machine learning.

V. RESULT AND ANALYSIS

The ResNet50, Inceptionv3 and the VGG16 model models are compared. Among these three models ResNet50 got highest accuracy of 99.05% hence we have predicted the seven classes of diseases using ResNet50 model. The predictions are as below:

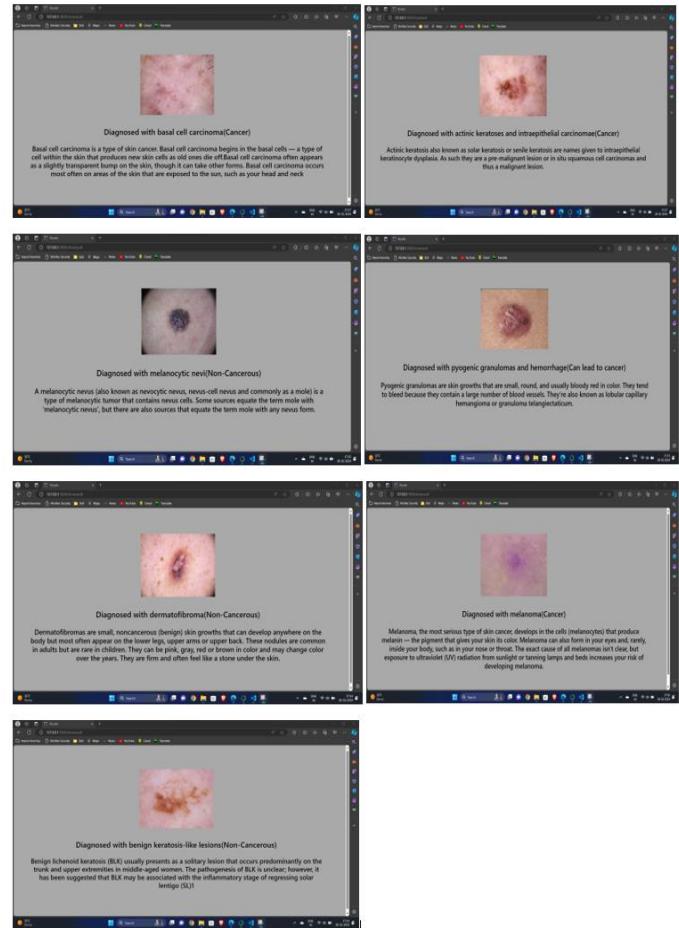


Fig 3. 7 classes prediction

5.1 RESULTS COMPARISON OF RESNET50, INCEPTIONV3, AND VGG16 MODELS:

TABLE III. COMPARISION METRICS

Model	Accuracy	Loss	Precision	Recall
ResNet50	98.9	0.02	99.16	99.15
Inceptionv3	94.8	0.15	97.27	97.25
VGG16	93.2	0.22	95.3	95.29

5.2 FORMULAE FOR COMPARISON:

$$\text{Accuracy: } (TP + TN) / (FP + FN + TP + TN).$$

$$\text{Precision: } = \frac{TP}{FP+TP}$$

$$\text{Recall (Sensitivity)} = \frac{TP}{FN+TP}$$

The proposed method's success in achieving high accuracy and balanced precision and recall values demonstrates its potential for real-world applications.

ResNet50:

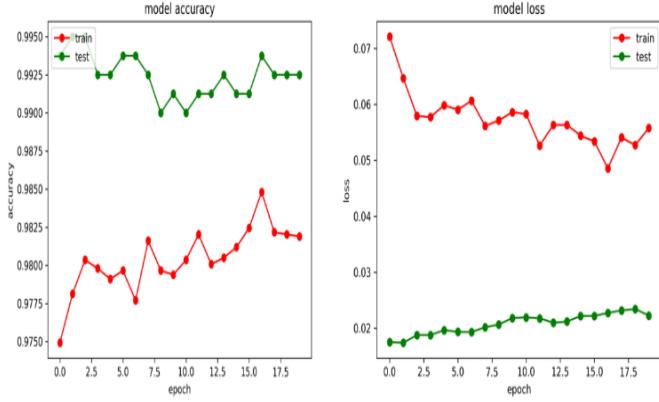


Fig 4. ResNet50 Accuracy and loss plots

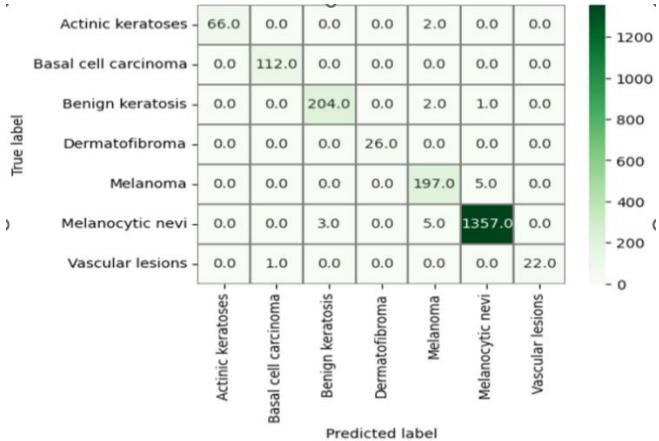


Fig 5. ResNet50 Confusion matrix

Inceptionv3:

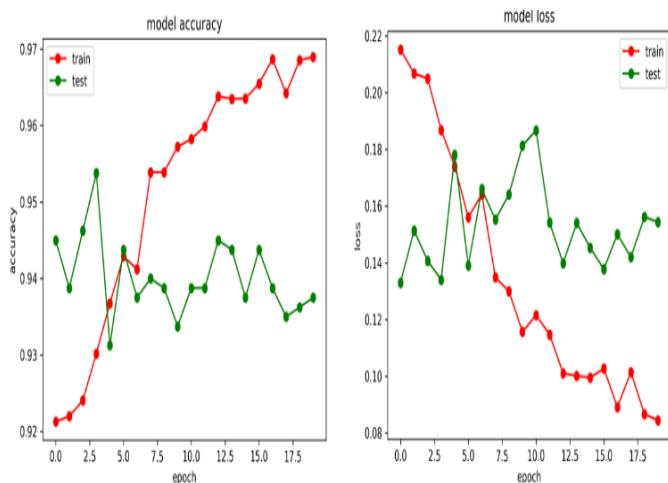


Fig 6. Inceptionv3 Accuracy and loss plots

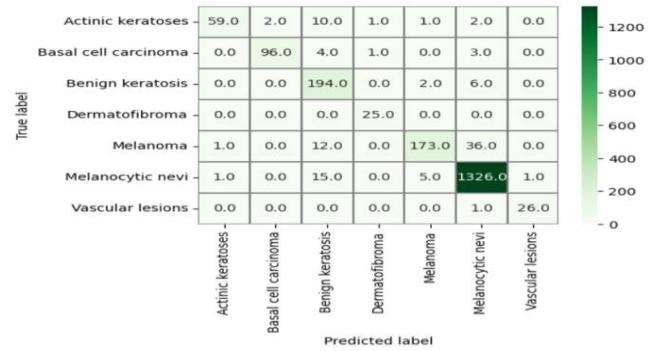


Fig 7. Inceptionv3 Confusion matrix

VGG16:

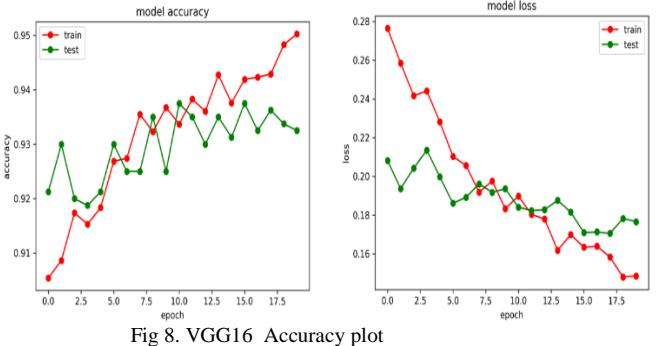


Fig 8. VGG16 Accuracy plot

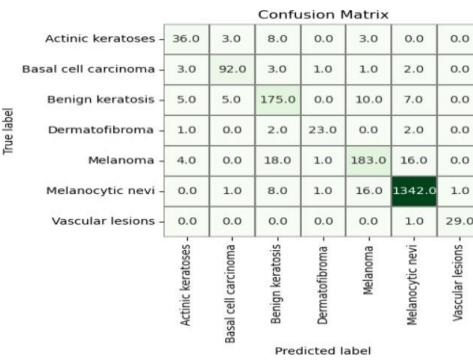


Fig 9. VGG16 Confusion matrix

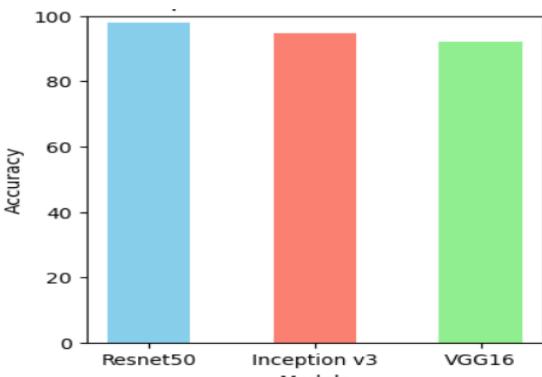


Fig 10. Comparison of 3 models accuracies

5.3 OBSERVATIONS

ResNet50 demonstrates superior predictive performance, achieving the highest success rates and lowest loss values.

Inceptionv3 performs reasonably well, with accuracy and loss values indicating effective skin cancer classification. VGG16, while providing acceptable results, lags behind ResNet50 and Inceptionv3 in terms of accuracy.

We compare the performance of the suggested technique with current algorithms in the literature to assess its efficacy. Resnet50 model introduced residual connections that enable training of much deeper networks. These architectural differences affected the model's ability to capture and represent complex patterns in the data. Resnet50 is known for its parameter efficiency due to the use of residual connections. This means it can achieve good performance with fewer parameters compared to models like VGG16, which can be advantageous especially in scenarios with limited computational resources or data availability. Resnet50 is deeper compared to InceptionV3 and VGG16. Deeper models can potentially learn more intricate features and representations, which might be beneficial for complex tasks like skin cancer detection.

VI. FUTURE SCOPE

Looking ahead, there are promising avenues to explore for refining existing models. Investigating ensemble methods that combine ResNet50, Inceptionv3, and VGG16 could yield a hybrid model harnessing the strengths of each architecture. Enhancing datasets with more diversity and volume holds the potential for improving generalization. Exploring transfer learning and adapting models to evolving datasets can contribute to increased accuracy. Collaborative efforts with healthcare professionals and integration into clinical workflows may facilitate practical applications, advancing the role of automated dermatological diagnosis in health care.

VII. CONCLUSION

In closing, the study successfully illustrates the possibility of models based on deep learning for the categorization of skin tumors, with ResNet50 exhibiting superior performance in accuracy and minimal loss. While ResNet50 emerges as a robust choice, acknowledging the distinct strengths of Inceptionv3 and VGG16 emphasizes the importance of a tailored model selection based on specific needs. The results underline the nuanced nature of model comparisons and the significance of understanding individual architectural characteristics.

REFERENCES

- [1] Vijayalakshmi, M. M. "Melanoma skin cancer detection using image processing and machine learning." International Journal of Trend in Scientific Research and Development (IJTSRD) 3.4 (2019): 780-784.
- [2] Harangi, Balazs. "Skin lesion classification with ensembles of deep convolutional neural networks." Journal of biomedical informatics 86 (2018): 25-32.
- [3] Thakur, Nikita, and Arunima Jaiswal. "Comparative Analysis of Machine Learning and Deep Learning Algorithms for Skin Cancer Detection." Machine Intelligence and Data Science Applications: Proceedings of MIDAS 2021. Singapore: Springer Nature Singapore, 2022. 409-418.
- [4] Gong, An, Xinjie Yao, and Wei Lin. "Classification for dermoscopy images using convolutional neural networks based on the ensemble of individual advantage and group decision." IEEE Access 8 (2020): 155337-155351.
- [5] Ali, Karar, et al. "Multiclass skin cancer classification using EfficientNets—a first step towards preventing skin cancer." Neuroscience Informatics 2.4 (2022): 100034.
- [6] Pacheco, Andre GC, and Renato A. Krohling. "An attention-based mechanism to combine images and metadata in deep learning models applied to skin cancer classification." IEEE journal of biomedical and health informatics 25.9 (2021): 3554-3563.
- [7] Imran, Azhar, et al. "Skin cancer detection using combined decision of deep learners." IEEE Access 10 (2022): 118198-118212.
- [8] Alizadeh, Seyed Mohammad, and Ali Mahloojifar. "Automatic skin cancer detection in dermoscopy images by combining convolutional neural networks and texture features." International Journal of Imaging Systems and Technology 31.2 (2021): 695-707.
- [9] Han, Seung Seog, et al. "Keratinocytic skin cancer detection on the face using region-based convolutional neural network." JAMA dermatology 156.1 (2020): 29-37.
- [10] Tschandl, Philipp, et al. "Expert-level diagnosis of non-pigmented skin cancer by combined convolutional neural networks." JAMA dermatology 155.1 (2019): 58-65.
- [11] Haggenmüller, Sarah, et al. "Skin cancer classification via convolutional neural networks: systematic review of studies involving human experts." European Journal of Cancer 156 (2021): 202-216.
- [12] Albahar, Marwan Ali. "Skin lesion classification using convolutional neural network with novel regularizer." IEEE Access 7 (2019): 38306-38313.
- [13] Harangi, Balazs. "Skin lesion classification with ensembles of deep convolutional neural networks." Journal of Biomedical Informatics 86 (2018): 25-32.
- [14] Y. Faziloglu, R.J. Stanley, R.H. Moss, W.V. Stoecker, R.P. McLean, Colour histogram analysis for melanoma discrimination in clinical images, Skin Res. Technol. 9 (2003) 147–155.
- [15] F. Nachbar, W. Stolz, T. Merkle, A.B. Cognetta, T. Vogt, M. Landthaler, G. Plewig, The ABCD rule of dermatoscopy: high prospective value in the diagnosis of doubtful melanocytic skin lesions, J. Am. Acad. Dermatol. 30 (4) (1994) 551–559.
- [16] R. White, D.S. Rigel, R. Friedman, Computer applications in the diagnosis and prognosis of malignant melanoma, Dermatol. Clin. 9 (4) (1992) 695–702.
- [17] A. Esteva, B. Kuprel, R.A. Novoa, J. Ko, S.M. Swetter, H.M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, Nature 542 (7639) (2017) 115–118.
- [18] Shanthi, T., R. S. Sabeenian, and R. Anand. "Automatic diagnosis of skin diseases using convolution neural network." Microprocessors and Microsystems 76 (2020): 103074.



PRIMARY SOURCES

- | | | |
|---|--|------|
| 1 | CMAK Zeelan Basha*, Sofi Tasneem, Preethi Miriyala, Syed Saleem Basha. "Enhanced Technique For Placement Monitoring using Servicenow Portal", International Journal of Innovative Technology and Exploring Engineering, 2019 | <1 % |
| 2 | doctorpenguin.com | <1 % |
| 3 | www.scpe.org | <1 % |
| 4 | "Robust Skin lesion Classification via Machine Intelligence", International Journal of Recent Technology and Engineering, 2020 | <1 % |
| 5 | Submitted to University of Durham | <1 % |
| 6 | Diponkor Bala, Md. Ibrahim Abdullah, Mohammad Alamgir Hossain, Mohammad Anwarul Islam, Md. Atiqur Rahman, Md. | <1 % |